

# Visualization of wave packets in Quantum Mechanics. Reflection and Transmission coefficients. Tunnel effect

Júlia Alsina Oriol, Marc Basquens Muñoz, and Conrad Corbella Bagot  
(Dated: May 24, 2016)

We developed an interactive program with Matlab that numerically solves the time-dependent Schrödinger equation for a gaussian wave packet under the action of different potentials and allows its visualization and propagation. Several parameters of both the wave packet and the potentials can be adjusted. Also, the reflection and transmission coefficients of the system are computed.

## I. INTRODUCTION

In Quantum Mechanics, a quantum state of a particle can be described using a wave function, that is a complex function that contains all the information about the system. In particular, its modulus squared tells us the probability distribution.

Any physical magnitude can be described as a linear hermitian operator, called observable. When we measure an observable, immediately the state of the system collapses to one of the eigenstates of the observable. The temporal evolution of a wave packet is given by the Schrödinger equation.

The graphic solution of Quantum Mechanics problems can help to its comprehension. Our objective is to design an interactive program that shows the quantum effects in reflection and transmission of wave packets under some potentials.

Usually, in a Quantum Mechanics course, this problem is solved for a single wave, but it is more realistic when wave packets are introduced. That program can be a self-learning tool to students that are in an introductory course to Quantum Mechanics.

## II. NUMERICAL METHODS

Let us consider the operator

$$\hat{F}(x, t) = -\frac{\hbar^2}{2m}\nabla^2 + V(x, t)$$

Then, the Schrödinger equation is:

$$i\hbar\frac{\partial\psi(x, t)}{\partial t} = \hat{F}(x, t)[\psi(x, t)]$$

We have developed several algorithms to numerically solve it. In order to do this, we have to discretize the wave function both in time and space. We call  $\Delta x$  and  $\Delta t$  the spatial and temporal spacing between the base points chosen.

First, the spatial part has been treated with the finite difference method in all cases. With this approximation, the laplacian yields:

$$\nabla^2\psi(x, t) \approx \frac{\psi(x + \Delta x, t) + \psi(x - \Delta x, t) - 2\psi(x, t)}{\Delta x^2}$$

For the temporal derivative, we have tested 3 different methods: forward Euler, backward Euler and fourth order Runge-Kutta.

- Forward Euler method consists in approximating the temporal derivative by:

$$\frac{\partial\psi(x, t)}{\partial t} \approx \frac{\psi(x, t + \Delta t) - \psi(x, t)}{\Delta t} \quad (1)$$

The spatial part of the equation is given by  $\hat{F}(x, t)[\psi(x, t)]$ . This method, despite being cheap in terms of computation, is relatively bad in terms of stability.

- Backward Euler method is a modification to forward Euler that grants more stability at the cost of more computations per iteration. The temporal derivative is approximated as in forward Euler (1) but now the spatial part is given by  $\hat{F}(x, t)[\psi(x, t + \Delta t)]$ , so we need to solve an equation system each step.
- Runge-Kutta of fourth order is a more complex explicit method based on quadratures that requires more computations every step, but has better convergence and stability.

### A. Stability

An important criterion to choose the numerical method that will be used is its stability. In order to have stability in all of the methods, it can be seen from mathematical analysis that it is required  $\frac{\Delta t}{\Delta x}$  to be small enough [3].

To compare the behaviour of the different options, we run a simulation of a freely propagating gaussian packet. We use  $\Delta x = 0.01$  and  $N = 2000$  base points to test the three methods. Forward Euler was unstable for values of  $\Delta t \geq 10^{-7}$ . For backward Euler, values of  $\Delta t \geq 10^{-4}$  cause the numerical solution to rapidly converge to the zero solution. Runge Kutta was stable for values of  $\Delta t \geq 10^{-3}$ .

### B. Accuracy

Another important feature of each method is its accuracy, that is, how close is the numerical solution to the

analytical one.

We use the analytical solution of a free particle in order to do the comparison:

$$\psi(x, t) = \frac{1}{\sqrt{a^2 + 4it}} e^{i\Delta k_0 \Delta((x-x_0)-k_0 t)} e^{-\frac{((x-x_0)-2k_0 t)^2}{a^2 + 4it}}$$

To check the similarity to the analytical solution, we run the three methods up to  $t = 0.25s$ . To compare the solutions we measure the difference in the position and the amplitude of the maxima of  $|\psi|^2$ . This criteria is used instead of computing the  $L^2$  norm of the difference of solutions because this last method could not give the correct result due to the ondulatory nature of the solutions. We also check the norm of the solutions.

For the test, we use a gaussian wave packet, centered at  $x_0 = -5$ , with the wave number peak at  $k_0 = 14.14$ , variance of  $\sigma = 1$  and maximum amplitude of  $A = 1$ . In the table we show the results after 0.25s of simulation: the position of the maximum of  $|\psi(x, t)|^2$  ( $x_{max}$ ), its maximum value ( $max\{|\psi|^2\}$ ) and the  $L^2$  norm of the numerical solution (Norm).

Method	$x_{max}$	$max\{ \psi ^2\}$	Norm
Analytical	2.0700	0.7071	1.2533
Forward Euler	2.0500	0.7114	1.2546
Backward Euler	1.9100	0.2646	0.4669
Runge-Kutta	2.1300	0.7045	0.6267

TABLE I. Simulation data after 0.25s vs. analytical solution.

We observe that for higher  $k_0$  the errors significantly increase. For values over  $k_0 = 60$ , Runge-Kutta is the only method that provides a solution near the analytical one, while the others give totally distorted results.

### C. Choice of method

Although the simple methods take less time to compute every step, if the minimum  $\Delta t$  required to have stability is very small, it may happen that the time used to calculate the same solution in the same simulation time is drastically greater than in more complex methods, but with greater  $\Delta t$ . In our implementations, in average:

Method	$\Delta t$	Time duration of each step (s)
Forward Euler	$10^{-7}$	$1.3 \cdot 10^{-4}$
Backward Euler	$10^{-4}$	$4.9 \cdot 10^{-4}$
4th order Runge-Kutta	$10^{-3}$	0.1068

TABLE II. Real computation time used vs. simulated time in each step. This data has been collected with an Intel Core i5-2400 CPU @ 3.10 GHz.

Hence, a simulation of 1s with these conditions would take around 1300s with Forward Euler, 4.9s with Backward Euler and 106s with Runge-Kutta. From this data, we conclude that Backward Euler is the fastest method.

However, the norm reduction that the method presents makes its use unfeasible. For this reason, we finally have chosen the 4th order Runge-Kutta method, which is balanced among the methods proposed in execution time and accuracy, being the only one that behaves reasonably for higher values of energy.

### D. Implementation

In order to use the Runge-Kutta method, we have separated the complex problem into two real problems by decoupling the real and imaginary parts. This can be done since the real and imaginary parts of the Schrödinger equation hold by linearity. Then we apply the algorithm for the decoupled model.

### E. Boundary conditions

We have implemented effective absorbing boundary conditions to prevent the wave packets to get back and create undesired effects when they reach the boundary of the box. We have achieved this by adding a dissipation term into the Schrödinger equation. There exist several ways to do this. In our case, we just added a negative term proportional to  $\psi(x, t)$  in such a way that the first derivative of the wave function becomes negative, causing the attenuation of the packet. Then, the equation that we are solving is:

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

where  $f(x)$  is the function that regulates where the damping occurs. In order to have absorbing boundary conditions, we want to attenuate  $\psi(x, t)$  near the boundaries. In our case, we have chosen a symmetric, high degree polynomial of the form:

In this way, when the wave packets get near the boundary, they drastically decay. Then, the Dirichlet boundary conditions in the code ensure that the incident wave packet disappears. To enhance visual experience of the simulation, we have cut the neighbourhood of the boundaries from the final plot.

Note that the dissipation term we have added can be interpreted as imaginary potential, since we can write:

$$V(x, t) \psi(x, t) - i f(x) \psi(x, t) = (V(x, t) - i f(x)) \psi(x, t)$$

This fact can be seen in the simulations. When a wave packet arrives to the boundary, due to this imaginary potential term, small rebounding waves appear.

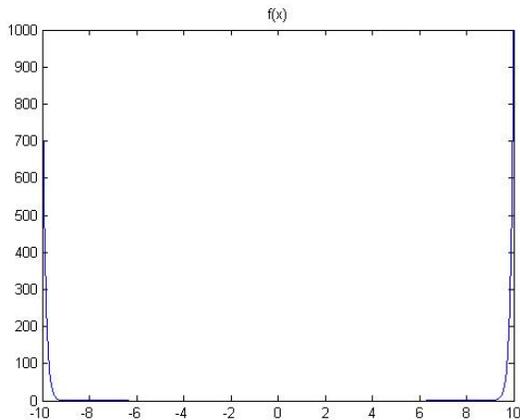


FIG. 1. The shape of the potential used to attenuate the boundaries.

### III. RESULTS

Given a potential shape and strength, and the energy, amplitude and mass of the particle, our program shows the real part of the temporal evolution of the wave function.

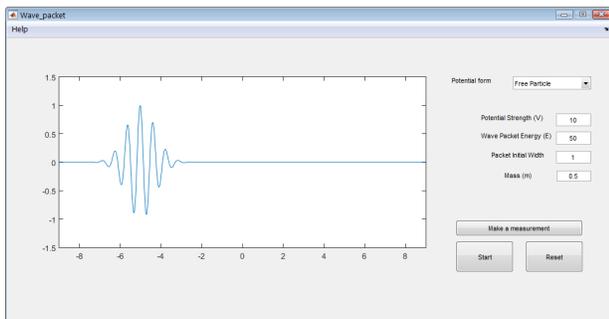


FIG. 2. The interface of the program.

In it, there is a pop-up menu where the potential shape can be selected. The available potential shapes are:

- Free particle. The potential is  $V=0$  everywhere.
- Step potential. When this option is selected, a cursor appears. That cursor allows to select the position where the step will take place. The potential strength can be selected writing its value in the corresponding box.
- Potential barrier. The difference between this option and the previous one is that, in the potential barrier, two positions have to be selected. The smaller one will correspond to the beginning of the barrier, and the greater one to the ending of it.
- Double step. This option is like the potential barrier, but it allows to set a potential different than 0 after the greater position of the two selected.

- Harmonic potential. It sets a potential of the form  $V(x) = V_0x^2$ , where  $V_0$  is the proportional to the potential strength selected.

The program calculates the transmission and reflection coefficients in the potential steps. In order to do an approximation to the coefficients, we could have used the formulas for a plane wave of energy  $E$ , but the results obtained do not match with the simulation, since the wave packet has a distribution of energies, so we have used the following formula instead:

$$T = \frac{\int_0^\infty T(k)\phi(k)dk}{\int_0^\infty \phi(k)dk}$$

where  $\phi(k)$  is the wavenumber distribution of the wave packet. This acts as a mean of the transmission coefficient for all the energies of the packet.

The wave packet shape can also be modified. It has always a gaussian distribution initially, but the energy, amplitude and mass can be selected.

Once you have selected the wave packed characteristics and the desired potential, the simulation can be runned, using the “Start” button. Once it is pressed, the simulation starts and the button changes to a “Stop” button. The simulation can also be reseted, using the “Reset” button.

As the 3rd postulate of Quantum Mechanics tells us, if we measure the position of the particle, immediately the system will collapse to the resulting point of the measurement. Then, the system will continue evolving according to the Schrödinger equation. In our program, this effect is the produced when the button “Make a measurement” is pressed.

#### A. Ressonance. Ramsauer-Townsend effect

Let us consider a plane wave under the action of a potential barrier of length  $a$  and potential strength  $V_0$ . It can be shown that the transmission and reflection coefficients are given by:

$$T = \left(1 + \frac{1}{4\varepsilon(\varepsilon - 1)} \sin^2(\lambda\sqrt{\varepsilon - 1})\right)^{-1}$$

$$R = \left(1 + \frac{4\varepsilon(\varepsilon - 1)}{\sin^2(\lambda\sqrt{\varepsilon - 1})}\right)^{-1}$$

where  $\lambda = a\sqrt{\frac{2mV_0}{\hbar^2}}$  and  $\varepsilon = \frac{E}{V_0}$

It is important to remark that if the energy of the plane wave is  $E = V_0 + \frac{n^2\pi^2\hbar^2}{2ma^2}$ , where  $n \in \mathbb{N}$ , the transmission coefficient is  $T = 1$ . This effect is known as Ramsauer-Townsend effect.

In the case of a wave packet we have a superposition of plane waves of different energies. If we set the central

energy to the one that produces Ramsauer-Townsend effect, there will be other energies that do not fulfill the condition  $T = 1$ . Due to that, it is difficult to observe this effect in our simulation. To observe it we could increase the initial packet width, since the central energy will have more weight.

Looking at the transmission coefficient it can be seen that it tends to 1 when we increase the packet. However, in the simulation is not appreciated due the contour limits.

### B. Changing the mass

It is also well known that in quantum mechanics, one must recover the classical result if the mass of the particle is large enough. Therefore, in that case there should not be an increase of the uncertainty in the position of a really massive particle while time passes.

That can be checked using the program developed. Let us consider two different particles, both with the same characteristics but the first one with a mass fourty times larger than the second. As it can be seen in the figure, in the first case there is practically no difference between the original particle and the one we have at  $t_0 + dt$  (in blue) while in the other case the difference is pretty clear:

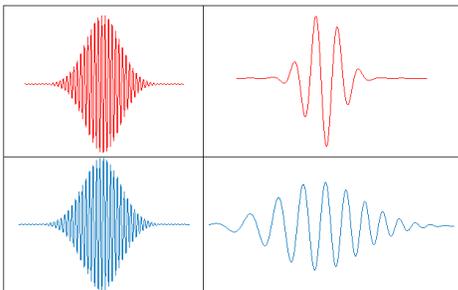


FIG. 3. In red, the original particles, in blue after  $dt$ .

### C. Changing the potential

One would not expect that a particle that has less energy than a potential barrier could overcome it. We would neither expect at first that a particle that has more energy than that of the barrier could be reflected. Even so, it is well known that it can occur in quantum mechanics and so, we can verify that is also takes place in our program (see figures 4 and 5).

### D. Zeno effect

The Zeno effect [2] suggests that if a particle is measured frequently enough, even if it is unstable, one can

freeze its evolution.

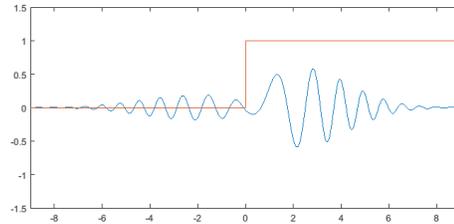


FIG. 4. In the left part of the figure, it can be seen the reflected part of the particle and in the right the transmitted in the case  $E > V_0$ .

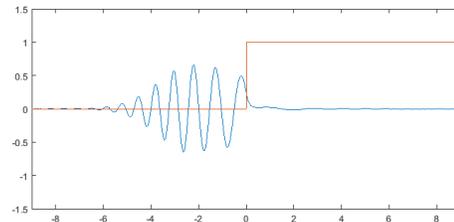


FIG. 5. The same as in the last figure but for the case  $E < V_0$ .

It has been checked that effect occurs using our program's button "Make a measurement".

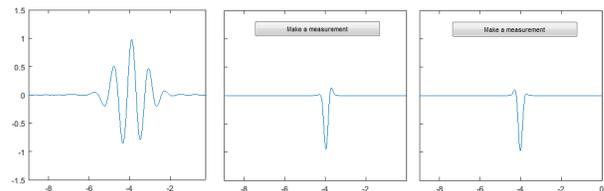


FIG. 6. If measures are performed frequently enough, the particle seems to be nearly in the same position.

## IV. CONCLUSIONS

That program has allowed us to understand better the behavior of a quantum system under the action of some potentials. Using it, we have checked experimentally some results learned at the Quantum Mechanics course.

Due to the simplicity of its use, that program will also be helpful to the future Quantum Mechanics students.

## V. ACKNOWLEDGEMENTS

We would like to acknowledge Jordi Boronat and Sonia Fernández, for their contributions to this work.

---

**VI. REFERENCES**

- [1] C. Cohen-Tannoudji. *Quantum mechanics*, Vol 1, pg 79-85
- [2] T. Fromhold. *Quantum Mechanics for Applied Physics and Engineering*
- [3] Ch.4 Stability of Finite Difference Methods  
<http://web.mit.edu/16.90/BackUp/www/pdfs/Chapter14.pdf>