

Simulating a two dimensional particle in a square quantum box with CUDA

George Zakhour

August 17, 2013

Contents

1	Background, Motive and Experiment	3
2	Expected Output	3
3	The mathematics of a particle in a box	4
3.1	In one dimension	4
3.1.1	Finding the Wave Function from Schrodinger's Equation .	4
3.1.2	Energy at each quantum level n	4
3.1.3	Probability function of the position	5
3.2	In two dimensions	5
3.2.1	The time-independent solution	5
3.2.2	Energy in two dimensions	5
3.2.3	Probability function of the position	6
3.3	Generalizing the time-independent solution	6
3.4	Generalizing the probability function	6
4	Implementation	7
4.1	Conventions	7
5	API	8
5.1	Structs	8
5.1.1	<code>_wave_function</code>	8
5.1.2	<code>particle</code>	8
5.2	Functions	8
5.2.1	<code>create_particle</code>	8
5.2.2	<code>probability</code>	8
6	Results	9
7	Code	10
7.1	Code	10
7.2	License: GNU General Public License	10

1 Background, Motive and Experiment

The particle in box problem is one of the first problems given to undergraduate students in a course about quantum mechanics. It showcases how the energy of particles is quantized and it highlights the probabilistic nature of quantum mechanics, especially with the idea that the particle is not located in a fixed position but it has a likelihood of existing at any point in space.

The particle in a box is an experiment in which a particle is stuck inside a box and cannot escape. The energy outside this box is ∞ while inside it is 0. The particle thus moves inside the box of dimensions $L \times L$.

Trying to imagine how these probabilities are scattered in the box is hard and one can't do without a simulation of the physical properties of the particle (position, energy and momentum).

In this simulation I try and simulate the "particle in a box" problem to display the probabilities of the position and the energy of the particle at each state.

2 Expected Output

Searching online, I have found a Youtube video that simulates the particle in a box¹, but lacks essential information on the state of the system, for example the dimensions of the box and the energy levels of the particle at each frame. Although lacking these essential variables I will base my results on the video.

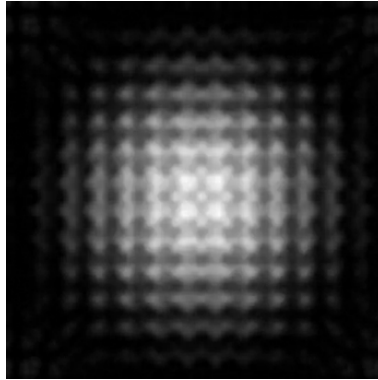


Figure 1: A screenshot of a "particle in a box" simulation found on Youtube

¹ Particle in a Box - Youtube <http://www.youtube.com/watch?v=jevKmFfcaxE>

3 The mathematics of a particle in a box

3.1 In one dimension

3.1.1 Finding the Wave Function from Schrodinger's Equation

The time dependent Schrodinger equation is given as:

$$\left(\frac{-\hbar^2}{2m}\nabla^2 + V\right)\Psi(x, t) = i\hbar\frac{\partial}{\partial t}\Psi(x, t) \quad (1)$$

The time independent Schrodinger equation is given as:

$$\left(\frac{-\hbar^2}{2m}\nabla^2 + V\right)\Phi(x) = E\Phi(x) \quad (2)$$

Where Ψ is the wave equation. In terms of Φ , Ψ is denoted as:

$$\Psi(x, t) = e^{-i(E/\hbar)t}\Phi(x) \quad (3)$$

Assuming the following is a solution to (2):

$$\Phi(x) = A\cos(kx) + B\sin(kx) \quad (4)$$

And given these two conditions that arise from the experiment:

$$\Phi(0) = \Phi(L) = 0$$

We plug them in the (4) and find the following:

$$\begin{cases} A &= 0 \\ k &= \frac{n}{L}\pi \quad (\text{n is the energy level}) \end{cases} \quad (5)$$

To find the value of B we need to normalize the equation.

The probability of finding the particle inside $[0; L]$ is 1 because it cannot escape.

$$\int_0^L |\Phi|^2 dx = 1 \quad (6)$$

$$B^2 \int_0^L \sin^2\left(\frac{n}{L}\pi x\right) dx = 1 \quad (7)$$

And thus $B = \sqrt{\frac{2}{L}}$.

Finally we get the solution to the time independent Schrodinger equation

$$\Phi(x) = \sqrt{\frac{2}{L}} \sin\left(\frac{n}{L}\pi x\right) \quad (8)$$

3.1.2 Energy at each quantum level n

We note the following

$$\frac{\partial^2}{\partial x^2}\Phi = -\left(\frac{n\pi}{L}\right)^2\Phi \quad (9)$$

If we replace the results in (2) we find:

$$E_n = \frac{\hbar^2 n^2 \pi^2}{2mL^2}$$

Or simply:

$$\begin{aligned} E_n &= n^2 E_1 \\ E_1 &= \frac{\hbar^2 \pi^2}{2mL^2} \end{aligned}$$

3.1.3 Probability function of the position

The probability of finding the particle between a and b is the following:

$$\begin{cases} \int_a^b |\Phi|^2 dx & \text{if } a, b \in [0, L] \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

If we wish to find the position inside a box of width ϵ and center a we would integrate between $a - \epsilon/2$ and $a + \epsilon/2$ where both ends are between 0 and L

The integration will lead to the following:

$$P(x) = \frac{\epsilon}{L} + \frac{1}{2n\pi} \left[\sin\left(\frac{2n\pi}{L}(a - \epsilon/2)\right) - \sin\left(\frac{2n\pi}{L}(a + \epsilon/2)\right) \right]$$

Which can be reduced more to the following form:

$$P(x) = \frac{\epsilon}{L} - \frac{1}{n\pi} \sin\left(\frac{n\pi}{L}\epsilon\right) \cos\left(\frac{2n\pi}{L}x\right)$$

3.2 In two dimensions

3.2.1 The time-independent solution

This time we suppose the solution is

$$\Phi(x, y) = X(x)Y(y) \quad (11)$$

$$X(x) = A \cos(k_x x) + B \sin(k_x x)$$

$$Y(y) = C \cos(k_y y) + D \sin(k_y y)$$

Plugging (11) in (2) and doing similar operations as 3.1.1 we get the following solution:

$$\Phi(x, y) = \frac{2}{L} \sin\left(\frac{n_x \pi}{L} x\right) \sin\left(\frac{n_y \pi}{L} y\right) \quad (12)$$

3.2.2 Energy in two dimensions

Doing the same steps as 3.1.2 we find that the energy has become:

$$E_{n_x, n_y} = \frac{\hbar^2 \pi^2}{2mL^2} (n_x^2 + n_y^2) \quad (13)$$

Or simply:

$$\begin{aligned} E_{n_x, n_y} &= (n_x^2 + n_y^2) E_{1,1} \\ E_{1,1} &= \frac{\hbar^2 \pi^2}{2mL^2} \end{aligned}$$

3.2.3 Probability function of the position

Similar to section 3.1.3 to find the probability of finding the particle inside the box of size $\epsilon \times \epsilon$ we integrate the wave function inside a box of center (x, y) and width ϵ .

$$P(x, y) = \int_{y-\epsilon/2}^{y+\epsilon/2} \int_{x-\epsilon/2}^{x+\epsilon/2} |\Phi(x, y)|^2 dx dy$$

After evaluating the integral we get the following function:

$$P(x, y) = \frac{1}{L^2} p(x) p(y) \quad (14)$$

$$p(\alpha) = \epsilon - \frac{L}{n_\alpha \pi} \cos\left(\frac{2n_\alpha \pi}{L} \alpha\right) \sin\left(\frac{n_\alpha \pi}{L} \epsilon\right)$$

3.3 Generalizing the time-independent solution

The equation that we have found so far depends on two quantum numbers n_x and n_y and describes the particle for these energy levels only. However the final time-dependant equation is a combination of many of these equations. The final time-independent equation is:

$$\Phi(x, y) = \sum_n c_n \Phi_{n_x, n_y}(x, y) \quad (15)$$

Where the constant c_n is the square root of the probability of getting the equation Φ_{n_x, n_y} that describes the particle in the box.

3.4 Generalizing the probability function

Since we have a more general wave function we need to have a more general probability function for the position. The new definition is generated from integrating Φ between $a - \epsilon/2$ and $a + \epsilon/2$. The result is:

$$P(x, y) = \sum_n c_n^2 P_n(x, y)$$

4 Implementation

4.1 Conventions

Some of the conventions adopted through the code are

- words in function and struct names are separated by an `_` (underscore)
- words in variable names are glued together with the first letter capitalized. Except in the case of the first character.
- functions that start with `_` are helper functions and should not be used directly. They may be considered like private methods and properties in object-oriented code. No major documentation is provided for these functions and the author only uses inline code comments to document them
- Tiny mathematical and physical constants, such as \hbar are expressed as float numbers without their orders. For example $\hbar = 1.054 \cdot 10^{-34}$ is defined the following way `#define HBAR 1.054 // without e-34`

5 API

5.1 Structs

5.1.1 `_wave_function`

This struct is used by the `particle` struct and describes a wave function

- `int n` the energy level of the particle described by the wave function
- `float energy` the energy of the particle at that energy level

5.1.2 `particle`

This struct represents a particle and have the following elements

- `float mass` the mass of the particle
- `_wave_function* wave` the wave function that describes the state of the particle

5.2 Functions

5.2.1 `create_particle`

This function is used to create a new particle.

```
void create_particle(particle* p, int n, float mass);
```

- `particle* p` the particle to create
- `int n` the energy level of the particle
- `float mass` the mass of the particle

5.2.2 `probability`

This function finds the probability of finding the particle in the box of center (x, y) and of dimensions $\epsilon \times \epsilon$

The function returns a `float`; the probability of finding the particle at points (x, y)

```
float probability(particle *p, float x, float y);
```

- `particle* p` the particle to find the probability of
- `float x` the x-coordinate of the center of the box
- `float y` the y-coordinate of the center of the box

6 Results

7 Code

7.1 Code

7.2 License: GNU General Public License

Copyright (C) 2013 George Zakhour

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.