UNIVERSITAT JAUME I

Escola Superior de Tecnologia i Ciències Experimentals

HOLES AND ELECTRONS IN CdSe NANOPLATELETS

Roser Fernadez Climent

Supervisor: Juan Ignacio Climente Plasencia

Department of Physical and Analytical Chemistry

Degree in Chemistry

Course 2015-2016

1

INDEX

1. INTRODUCTION

This report will talk about the Colloidal Metal Chalcogenides Semiconductors, specifically quasi two-dimensional ones, the properties that make them different from the 0D quantum dots (QDs) or 1D quantum wires and consequently the interest of their study and the need of theoretical models in order to simulate the optoelectronics response.

We shall develop two methods to program with Mathematica which calculate energies, wave functions and radiative recombination rates for the quasi 2D semiconductors and at last, the reason why only one of these two methods can work with complex conditions while the other one only works in simple cases.

About 30 years ago the optical properties of semiconductor colloidal QD, also known as nanocrystals, were discovered. Since then, the synthesis of these colloidal semiconductors nanoparticles is becoming mature enough so that they have started to be incorporated in devices[1].

When an electron in the valence band (VB) of a semiconductor nanostructure is excited it is transferred to the energy level of the conduction band (CB) leaving a hole in the valence band, see on Figure 1 panel (**1**). Once in the excited state, the electron relaxes non-radioactively down to the lowest CB level in a very fast time scale (ps) panel (**2**) and next back to the VB ground state by emitting photon energy (*hv*) panel (**3**).

How to change to change the color of the emission? The photon emission energy depends on the semiconductor gap size. Then, it is necessary modify the gap to change the energy and consequently the color.
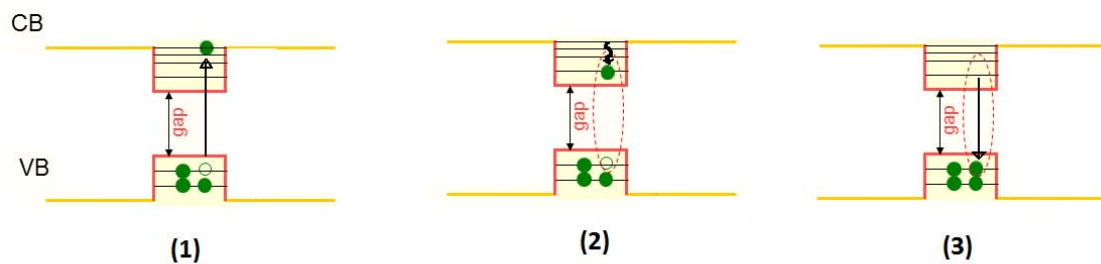

Figure 1. Schematic of the electron fluorescence in a semiconductor nanocrystal.

The QD gap, $E_{gap}{}^{QD}$, is the energy difference between the lowest CB and the highest VB states:

$$E_{gap}{}^{QD} = E_{gap} + E_{gs}{}^{CB} + E_{gs}{}^{VB} \qquad (1.1)$$

where E_{gap} is the energy difference between the bottom of the CB and the top of the VB, i.e. the bulk semiconductor energy gap (denoted as *gap* in Fig.1). $E_{gs}{}^{CB}$ is the energy of the ground state in the CB and $E_{gs}{}^{VB}$ is the ground state energy of the VB.
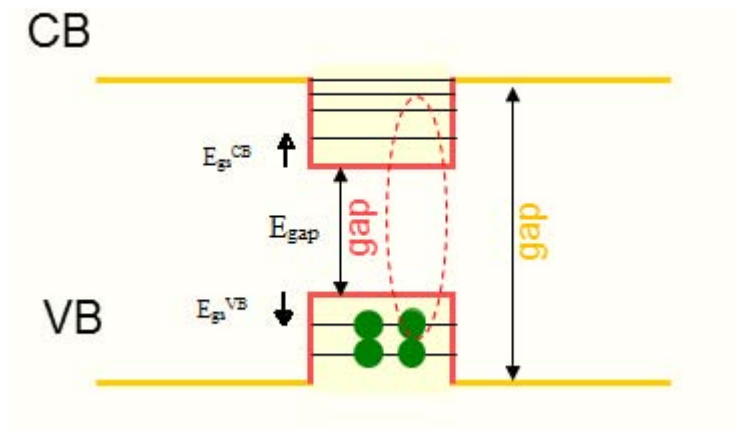
**Figure 2.** Diagram showing the energy terms of Eq. (1.1)

Because colloidal QDs are embedded in organic media, they can be seen as quantum boxes. Then, the electrons and holes energy are:

$$E_{gs}{}^{CB} = \frac{\hbar^2 \pi^2}{2m_e}\left(\frac{1}{L_x{}^2} + \frac{1}{L_y{}^2} + \frac{1}{L_z{}^2}\right) \qquad (1.2)$$

$$E_{gs}{}^{VB} = \frac{\hbar^2 \pi^2}{2m_h}\left(\frac{1}{L_x{}^2} + \frac{1}{L_y{}^2} + \frac{1}{L_z{}^2}\right) \qquad (1.3)$$

where Lx, Ly and Lz are the QD dimensions and m_e and m_h are the masses of the electron and the hole respectively.

Taking colloidal nanocrystals of different sizes one can form boxes of different sizes too in which the emission energy and the emitted color change (see Fig. 3).

$$E \propto 1/L^2$$
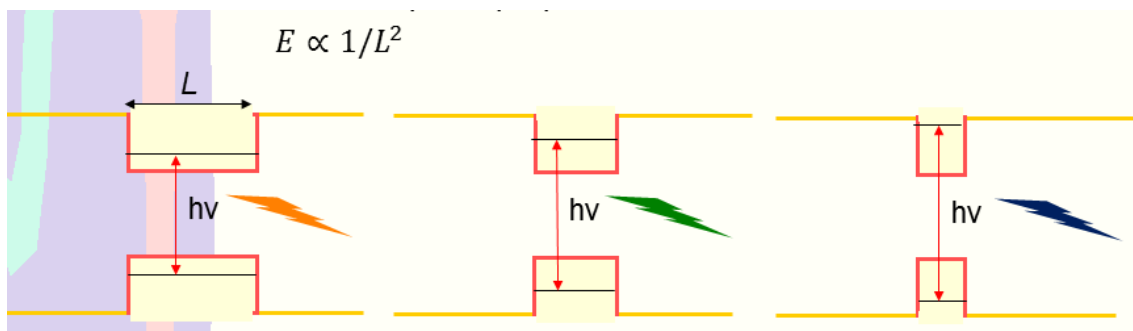
Figure 3. The energy of emitted photons scales inversely proportional to the nanocrystal size.

An important development that recently took place in the field of colloidal quantum dots is the synthesis of two-dimensional (2D) semiconductor nanoplatelets (NPLs) that appear as free-standing nanosheets.
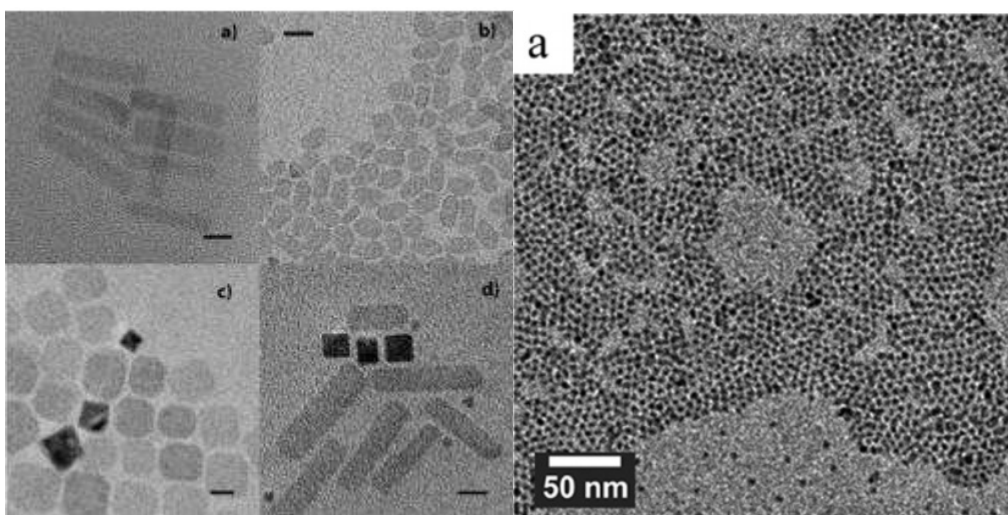
Figure 4. On the left, the transmission electron microscope images of CdSe colloidal nanoplatelets. Different lateral shapes (a-d) can be obtained using different precursors. Scale bars: b) 20nm, others, 10nm [2]. On the right, TEM images of bare CdSe quantum dots of average size 5 nm. [3]

From the physical point of view, NPLs are of particular interest because it has been shown that their thickness (Lz) can be controlled with atomic precision, so that no inhomogeneous broadening is observed. These large planar facets can be precisely defined chemically and as a consequence, NPLs can serve as a model system to better understand the effect of surface chemistry, of composition, and of confinement in semiconductors systems.

Because they have two large free interfaces, mirror charges play an important role, and the electron-hole Coulomb interaction due to the small dielectric constant of the surrounding media is very high. These two effects almost perfectly compensate each other; it results in particles with unique spectroscopic properties such as fast fluorescence lifetime and extreme color purity (narrow full width at half-maximum of their emission spectra).
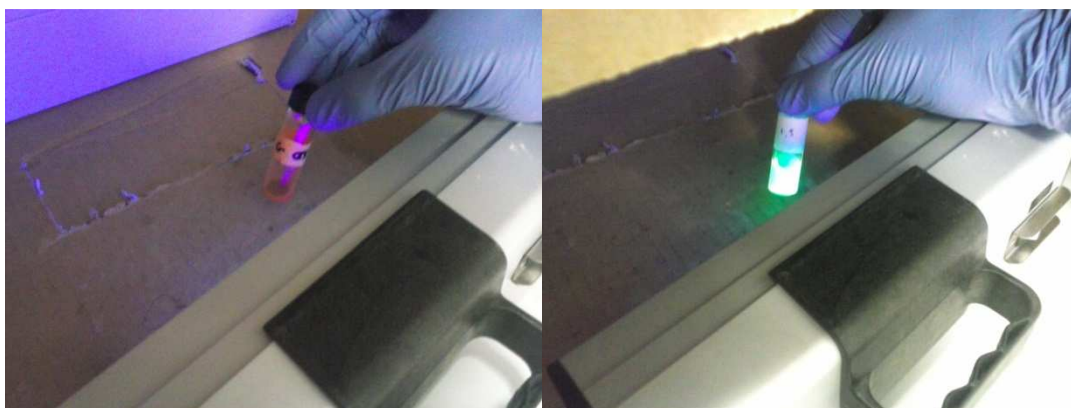
Figure 5. On the left is shown the fluorescence emitted by QDs and on the right the fluorescence emitted by NPLs. We can observe that the second one is more powerful and presents a more pure color.

From the chemical point of view, these colloidal particles are model system to study the role of ligands since they have precisely defined facets. In addition, the synthesis of these highly anisotropic objects triggered new research to understand at a mechanistic level how this strong anisotropy could be generated.

From the application point of view, 2D colloidal NPLs, offer interesting perspectives when color purity, charge conductivity, or field tunable absorption are required[1].

Colloidal NPLs exhibit three striking features compared with QDs: a narrow optical feature, fast photoluminescence (PL) lifetime, and almost no Stokes's shift.

The radiative fluorescent lifetime measured in CdSe nanoplatelets decreases with temperature, reaching 1 ns at 6 K, two orders of magnitude less than for spherical CdSe nanoparticles.[4] This makes the nanoplatelets the fastest colloidal fluorescent emitters and strongly suggests that they show a giant oscillator strength transition.

This work focuses on the simulation of excitons in NPLs. The goal is to predict the emission energy and radiative recombination rate as a function of the NPL dimension for different materials.

To this end, we develop two computational models. In the first model we consider the NPL can be defined as an ideal quantum box. This allows us to use the well-known analytical solutions of the textbook 'particle-in-a-box' model. In the second model, we use finite differences method in the (x,y) directions. While such a numerical method model is more computationally demanding, we shall see that it allows us to investigate regimes where the former model is not valid.

2. THEORETICAL FOUNDATIONS

The system consisting of a particle confined in a box of potential with impenetrable walls constitutes one of the most paradigmatic problems of quantum mechanics. This is mainly due to two reasons. On the one hand, it presents a simple and manageable analytical solution, while on the other hand it manifests, despite its simplicity, the main aspects of physics that govern the microscopic world. In fact, sometimes it allows to qualitatively interpreting the behavior of quantum systems considerably more complex, some of whom can be found among the objects of study in various fields of most current research. This is precisely the case of NPLs, which we deal with in the present work.

For the reason we shall briefly review here the main aspects of the particle-in-a-box model and the adapt it to the case of our study. Firstly 1D, for its simplicity and next 3D (2D+1D).
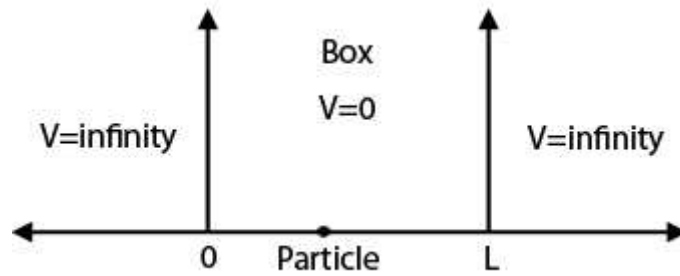

Figure 6

2.1 ANALYTICAL METHOD

The analytical method is, in execution terms, an easy and fast method to obtain the energies and own eigenfunctions of the system.

Based on the expression of the Hamiltonian:

$$H = \frac{p^2}{2m} + V \tag{2.1}$$

And being p the moment:

$$p_x = -i\hbar \frac{\partial}{\partial x} \tag{2.2}$$

Substituting (2.2) in (2.1) and applying it to the Schrodinger equation $H\psi = E\psi$, the standing wave equation describing the particle inside the box is:

$$-\frac{\hbar^2}{2m}\frac{d^2}{dx^2}f(x) = Ef(x) \tag{2.3}$$

where, without loss of generality , we have assumed that the potential inside the box is zero V=0. The solutions f(x) of this equation must also satisfy the boundary conditions imposed by the impenetrability of the potential barriers:

$$f(0) = f(L) = 0 \tag{2.4}$$

The analytical integration of equation (1.1) under the imposition of (1.2), which can be found in any introductory text of quantum mechanics, provides solutions:

$$f_n(x) = \sqrt{\frac{2}{L}}Sin[\frac{n\pi}{L}x] , n = 1,2,3... \tag{2.5}$$

$$E_n = \frac{\hbar^2\pi^2}{2mL^2}n^2, n = 1,2,3... \tag{2.6}$$

for the wave functions (standard) and allowed energies. The quantum number n=1,2,3,..., which quantifies label functions and their associated energies, occurs naturally in imposing the boundary conditions.
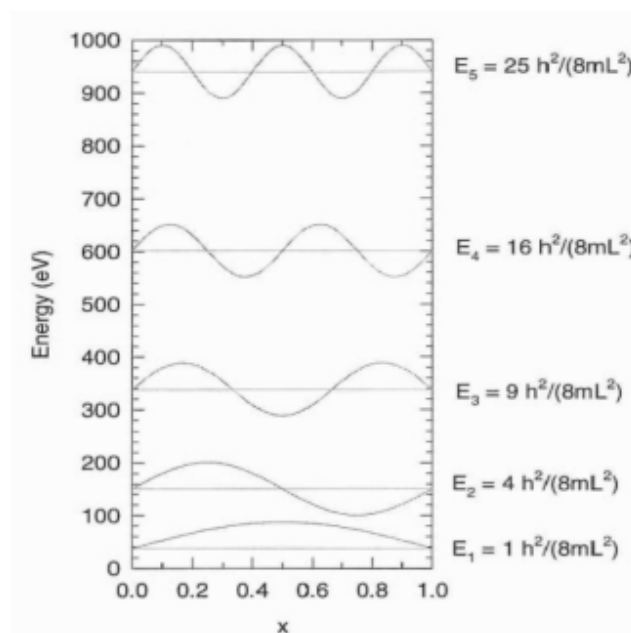
Figure 7

2.2 NUMERICAL METHOD: FINITE DIFFERENCE NUMERICAL INTEGRATION

Another way to get the energy and functions of the system is to use numerical methods for integrating equation (1.1). In our case (particle in the box impenetrable walls) it is additional method to analytical, but in more complex systems we will see later, the numerical method is the only way able to offer a satisfactory solution to the problem.

Numerical methods are based on approximating each own function f(x) by estimating the value it takes only in certain discrete values of the variable f(x), obtaining a set of values

$$\{f(x1), f(x2),....,f(xN)\}$$

(which we abbreviate as $\{f1, f2,..., f_N\}$) at the points $\{x1, x2,...,xn\}$ (which we abbreviate as $\{1, 2,..., n\}$).

The arrangement in space of this set of points is known as mesh discretization (see Figure 1.3). As outlined below, obtaining numerical functions in turn allows for an estimate of their energies.

Given a specific function f(x), equation (1.1) must be satisfied at all points of the variable, and in particular, at all points of the mesh discretization. For a generic point xi it is satisfied therefore:

$$-\frac{\hbar^2}{2m} f i'' = E f i'' \tag{2.7}$$

Since the function f is defined only for certain values of the coordinate, calculating the derivative appearing in (1.7) is not trivial, because we cannot apply the limits shown in the usual definition of derivative of a function. It is therefore necessary to adapt the definition of a derivative. As shown in Figure 1.2, a good strategy is to approximate the tangent line to the function at point i by secant defined by its two adjacent points i+1 and i-1. Thus, the derivative is approximated by

$$f'(x_i) = f_i' = \frac{f_{i+1} + f_{i-1}}{2h} \tag{2.8}$$

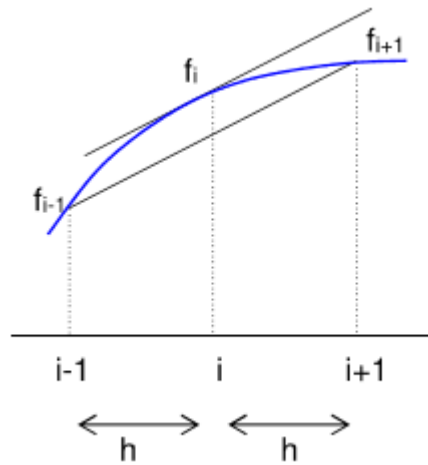Where h is the called *discretization step*. In consequence

$$f'(x_i) = f_i' = \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} \qquad (2.9)$$

Substituting (1.8) and (1.9) in (1.7) and rearranging terms, one obtains:

$$\left(-\frac{\hbar^2}{2mh^2}\right)f_{i-1} + \left(\frac{\hbar^2}{mh^2}\right)f_i + \left(-\frac{\hbar^2}{2mh^2}\right)f_{i+1} = E f_i \qquad (2.10)$$

which has the form:

$$b\, f_{i-1} + a\, f_i + b\, f_{i+1} = E\, f_i \qquad (2.11)$$

Figure 8

We can write as many equations of type (1.11) as points exist in the mesh discretization, the ones corresponding to i=1 and i=N are trivial (as f1=fn=0) and can be ignored (see Figure 1.3). Thus we obtain a system of equations N-2 can be expressed in matrix form as:

$$\begin{bmatrix} a & b & & & & \\ b & a & b & & & \\ & \ddots & \ddots & \ddots & & \\ & & b & a & b \\ & & & b & a \end{bmatrix} \cdot \begin{bmatrix} f_2 \\ f_3 \\ \vdots \\ f_{N-2} \\ f_{N-1} \end{bmatrix} = E \begin{bmatrix} f_2 \\ f_3 \\ \vdots \\ f_{N-2} \\ f_{N-1} \end{bmatrix} \qquad (2.12)$$

The above expression represents a matrix equation of eigenvalues M*f*=E*f*. The diagonalization of the matrix M allows for the eigenvalues and eigenvectors E f. These correspond, respectively, with the allowed energies of the particle and the discretized eigenfunctions.
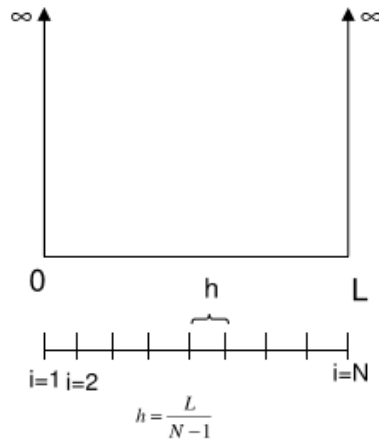
Figure 9

Now we consider a 2D system with dimensions Lx Ly.

From the discretization in the case of a single dimension it is easy to obtain it for a two-dimensional system.

Now, the eigenfunction depends on x and y coordinates so for f(xy) we obtain a set of values

$$\{f(x1y1), f(x1y2),....,f(xNyN)\}$$

Which we abbreviate as $\{f_{11}, f_{12},....,fnn\}$ at the points $\{x1y1, x1y2,...,xnyn\}$ (which we abbreviate as $\{11, 12,...,n\,n\}$).

The point of the variable x will be i and the points of the variable y will be j.

But as we have defined the boundary conditions at the ends of the box function is 0
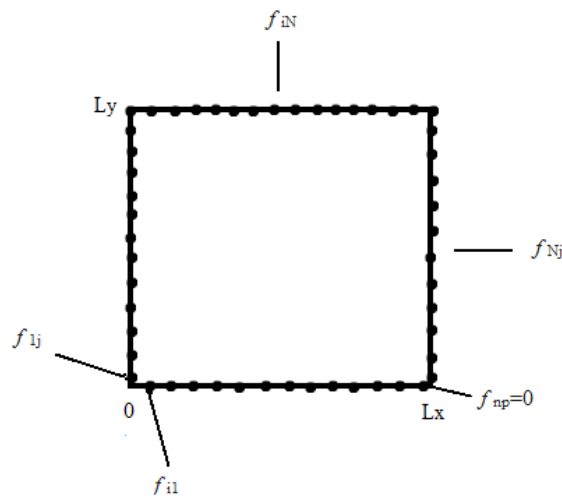
Figure 10

when the i and j worth the first of the last point, 1 or np (from now N 'll call np, number of points), will not consider these functions: $f_{1j} = f_{i1} = f_{npj} = f_{inp} = 0$, so:

$$\{f_{22}, f_{23}, \ldots, f_{np-1,np-1}\}$$

Now for a generic point ij it is satisfied:

$$-\frac{\hbar^2}{2m}\left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right)f'' = Ef'' \tag{2.13}$$

And the derivative is approximated by:

$$\frac{\partial^2 f_{ij}}{\partial x^2} = \frac{f_{i-1,j} - 2f_{ij} + f_{i+1,j}}{h^2{}_x} \tag{2.14}$$

$$\frac{\partial^2 f_{ij}}{\partial y^2} = \frac{f_{i,j-1} - 2f_{ij} + f_{i,j+1}}{h^2{}_y} \tag{2.15}$$

Being hx and hy:

$$h_x = \frac{L_x}{np-1} \qquad h_y = \frac{L_y}{np-1}$$

Then, substituting (2.14) and (2.15) in (2.13) we obtain the equation:

$$\left(-\frac{\hbar^2}{2m_\perp h^2{}_x}\right)f_{i-1,j} + \left(-\frac{\hbar^2}{2m_\perp h^2{}_y}\right)f_{i,j-1} + \left(\frac{\hbar^2}{m_\perp h^2{}_x}\right)f_{ij} + \left(\frac{\hbar^2}{m_\perp h^2{}_y}\right)f_{ij} +$$

$$\left(-\frac{\hbar^2}{2m_\perp h^2{}_y}\right)f_{i,j+1} + \left(-\frac{\hbar^2}{2m_\perp h^2{}_x}\right)f_{i+1,j} = Ef_{ij} \tag{2.16}$$

which we rewrite as:

$$a_x f_{i-1,j} + a_y f_{i,j-1} + (b_x + b_y)f_{ij} + a_y f_{i,j+1} + a_x f_{i+1,j} = Ef_{ij} \tag{2.17}$$

Where, then we will define for the matrix that $b = b_x + b_y$

With the point i and j of our 2D system we write as many equations (2.17) as possible. With all of them we build the Hamiltonian matrix, which this time will not have a fixed format for all np values. For each np change not only the dimensions of the matrix (np-2)(np-2), as was in 1D case, but also appear some 0 in the ay diagonals (upper and lower).

For example if we take np=5: the dimensions of the matrix will be (np-2)(np-2) = 9, the 0 in the ay diagonals will appear each (np-2) = 3 rows and the matrix would have this form:

$$
\begin{bmatrix}
b & ay & 0 & ax & 0 & 0 & 0 & 0 & 0 \\
ay & b & ay & 0 & ax & 0 & 0 & 0 & 0 \\
0 & ay & b & 0 & 0 & ax & 0 & 0 & 0 \\
ax & 0 & 0 & b & ay & 0 & ax & 0 & 0 \\
0 & ax & 0 & ay & b & ay & 0 & ax & 0 \\
0 & 0 & ax & 0 & ay & b & 0 & 0 & ax \\
0 & 0 & 0 & ax & 0 & 0 & b & ay & 0 \\
0 & 0 & 0 & 0 & ax & 0 & ay & b & ay \\
0 & 0 & 0 & 0 & 0 & ax & 0 & ay & b
\end{bmatrix}
\begin{vmatrix}
f_{22} \\ f_{23} \\ f_{24} \\ f_{32} \\ f_{33} \\ f_{34} \\ f_{42} \\ f_{43} \\ f_{44}
\end{vmatrix}
= E
\begin{vmatrix}
f_{22} \\ f_{23} \\ f_{24} \\ f_{32} \\ f_{33} \\ f_{34} \\ f_{42} \\ f_{43} \\ f_{44}
\end{vmatrix}
\qquad (2.18)
$$

To set these zeros, we build matrix for different np and see that is always true that the 0 in the lower diagonal begins at row (np-1) and each row+(np- 2) is repeated while the upper diagonal 0 begin in row (np -2) row and each row+(np- 2) is repeated.

Another thing that also changes when we build the Hamiltonian nuance 2D is the difference the diagonal ax ay regarding, for each np is a different distance. The upper diagonal always begins in the first row and ends at the last less np-2 while the lower row begins in np- 1 and ends at the last.

2.3 HAMILTONIAN OF ELECTRONS AND HOLES

As mentioned in the introduction, the emission energy of a NPL is given by

$$
E_{gap}{}^{QD} = E_{gap} + E_{gs}{}^{CB} + E_{gs}{}^{VB} \qquad (2.19)
$$

where E_{gap} is the gap energy of the semiconductor material that forms the NPL, which can be found in tables[5] while $E_{gs}{}^{CB}$ is the lowest (electron) state of the conduction band and $E_{gs}{}^{VB}$ is the highest (hole) state of the valence band.

Neglecting Coulomb interaction between the electron and holes, the latter energies can be obtained using effective mass (single-band k-p) theory using particle-in-the-box Hamiltonians:

$$H_e{}^{CB} = \frac{p_x{}^2}{2m_e} + \frac{p_y{}^2}{2m_e} + \frac{p_z{}^2}{2m_e} \tag{2.20}$$

$$H_h{}^{VB} = \frac{p_x{}^2}{2m_{\perp h}} + \frac{p_y{}^2}{2m_{\perp h}} + \frac{p_z{}^2}{2m_{zh}} \tag{2.21}$$

Here m_e is the effective mass of the electron, which for a given material can be obtained from tables.[5] Likewise m_h is the effective mass of holes. Notice that hole masses are highly anisotropic, so different values are used in the NPL plane $(m_\perp)$ and in the thin direction $(m_\parallel)$.

In the analytical model, the solution of equations (2.20) and (2.21) are straightforward. For the ground state (nodeless state, with $n_x=n_y=n_z=1$) the eigenenergies and eigenfunctions are:

$$E_{gs}{}^{CB} = \frac{\hbar^2\pi^2}{2m_e L_x{}^2} + \frac{\hbar^2\pi^2}{2m_e L_y{}^2} + \frac{\hbar^2\pi^2}{2m_e L_z{}^2} \tag{2.22}$$

$$E_{gs}{}^{VB} = \frac{\hbar^2\pi^2}{2m_{\perp h} L_x{}^2} + \frac{\hbar^2\pi^2}{2m_{\perp h} L_y{}^2} + \frac{\hbar^2\pi^2}{2m_{zh} L_z{}^2} \tag{2.23}$$

$$\psi_{gs}{}^{CB} = \psi_{gs}{}^{VB} = \sqrt{\frac{2}{L_x}} Sin(\frac{\pi x}{L_x}) \cdot \sqrt{\frac{2}{L_y}} Sin(\frac{\pi y}{L_y}) \cdot \sqrt{\frac{2}{L_z}} Sin(\frac{\pi z}{L_z}) \tag{2.24}$$

In the numerical model, we separate in-plane and vertical motion. The wave function is taken of the form $\psi_{gs}{}^{CB} = f(x,y) \cdot \phi(z)$ is obtained via finite differences as shown in Section 2.2, while $\phi(z)$ is the analytical solution

$$\phi(z) = \sqrt{\frac{2}{L_z}} Sin(\frac{\pi}{L_z}) \tag{2.25}$$

The corresponding energies are:

$$E_{gs}{}^{CB} = E_{xy} + E_z \tag{2.26}$$

Again, E_{xy} is obtained from 2D numerical integration method, while $E_z = \frac{\hbar^2\pi^2}{2m_z} \frac{1}{L_z{}^2}$

The same approximations are used to obtain $\psi_{gs}{}^{VB}$ and $E_{gs}{}^{VB}$.

2.3.1 HAMILTONIAN WITH AN ELECTRIC FIELD

We consider the case in which we add to our system an electric field F in a certain direction, in the x-axis. We build the Hamiltonian matrix for the electron and for the hole for this new case containing that field.

Now, the expressions for the particle-in-the-box Hamiltonians takes the form:

$$H_e{}^{CB} = \frac{p_x{}^2}{2m_e} + \frac{p_y{}^2}{2m_{\perp h}} + V + q_e \cdot F \cdot x \tag{2.27}$$

$$H_h{}^{VB} = \frac{p_x{}^2}{2m_{\perp h}} + \frac{p_y{}^2}{2m_{\perp h}} + V + q_h \cdot F \cdot x \tag{2.28}$$

where the potential V is cero, as it has been defined previously, and q_e and q_h the electron and hole charge respectively and F the intensity.

When we apply the Schrödinger Equation in each Hamiltonian we obtain:

$$\left(\frac{p_x{}^2}{2m_e} + \frac{p_y{}^2}{2m_e} + q_e \cdot F \cdot x\right) \cdot \psi = E\psi \tag{2.29}$$

$$\left(\frac{p_x{}^2}{2m_{\perp h}} + \frac{p_y{}^2}{2m_{\perp h}} + q_h \cdot F \cdot x\right) \cdot \psi = E\psi \tag{2.30}$$

If we operate them according Section 2.2 including the camp in the f_{ij} term and considering that when we discretize x it is (i-1)h_x:

$$\left(-\frac{\hbar^2}{2m_e h^2{}_x}\right)f_{i-1,j} + \left(-\frac{\hbar^2}{2m_e h^2{}_y}\right)f_{i,j-1} + \left(\frac{\hbar^2}{m_e h^2{}_x}\right)f_{ij} + \left(\frac{\hbar^2}{m_e h^2{}_y}\right)f_{ij} + ((i-1)q_e F h_x)f_{ij} +$$

$$(-\frac{\hbar^2}{2m_e h^2{}_y})f_{i,j+1} + (-\frac{\hbar^2}{2m_e h^2{}_x})f_{i+1,j} = E f_{ij} \tag{2.31}$$

$$\left(-\frac{\hbar^2}{2m_{\perp h} h^2{}_x}\right)f_{i-1,j} + \left(-\frac{\hbar^2}{2m_{\perp h} h^2{}_y}\right)f_{i,j-1} + \left(\frac{\hbar^2}{m_{\perp h} h^2{}_x}\right)f_{ij} + \left(\frac{\hbar^2}{m_{\perp h} h^2{}_y}\right)f_{ij} + ((i-1)q_h F h_x)f_{ij} +$$

$$(-\frac{\hbar^2}{2m_{\perp h} h^2{}_y})f_{i,j+1} + (-\frac{\hbar^2}{2m_{\perp h} h^2{}_x})f_{i+1,j} = E f_{ij} \tag{2.32}$$

which we rewrite as:

$$a_x f_{i-1,j} + a_y f_{i,j-1} + (b_x + b_y + kw)f_{ij} + a_y f_{i,j+1} + a_x f_{i+1,j} = E f_{ij} \tag{2.33}$$

being k=(i-1) and w=qFh_x

For the points ij we write as many equations (2.33) as possible and then we build with all of them the Hamiltonian matrix. It is similar to (2.18) but the electric field is added to the main diagonal. Besides this there is another difference, the term k does not change his value of row into row, instead of, k changes every (np-2) rows. It will be no easy to program that, it will be necessary a double loop (see it in annex).

For example, for 5 points we obtain a (np-2)(np-2) matrix like this:

$$
\begin{bmatrix}
b+w & ay & 0 & ax & 0 & 0 & 0 & 0 & 0 \\
ay & b+w & ay & 0 & ax & 0 & 0 & 0 & 0 \\
0 & ay & b+w & 0 & 0 & ax & 0 & 0 & 0 \\
ax & 0 & 0 & b+2w & ay & 0 & ax & 0 & 0 \\
0 & ax & 0 & ay & b+2w & ay & 0 & ax & 0 \\
0 & 0 & ax & 0 & ay & b+2w & 0 & 0 & ax \\
0 & 0 & 0 & ax & 0 & 0 & b+3w & ay & 0 \\
0 & 0 & 0 & 0 & ax & 0 & ay & b+3w & ay \\
0 & 0 & 0 & 0 & 0 & ax & 0 & ay & b+3w
\end{bmatrix}
\begin{bmatrix}
f_{22} \\ f_{23} \\ f_{24} \\ f_{32} \\ f_{33} \\ f_{34} \\ f_{42} \\ f_{43} \\ f_{44}
\end{bmatrix}
= E
\begin{bmatrix}
f_{22} \\ f_{23} \\ f_{24} \\ f_{32} \\ f_{33} \\ f_{34} \\ f_{42} \\ f_{43} \\ f_{44}
\end{bmatrix}
$$

(2.34)

To build the matrix we consider two matrixes, one like (2.34) but without b terms and another one with b in the main diagonal and then we add them.

This matrix (2.34) will be built twice, for the electron and for the hole. To calculate the energy the vertical motion will be added analytically, see in Section 2.3.

2.4 ELECTRONIC TRANSITION PROBABILITY IN SEMICONDUCTORS

When an electron is excited from the valence band (VB) to the conduction band (CB) it leaves a hole behind. Then, the electron returns to the VB emitting a photon with energy *hv.*
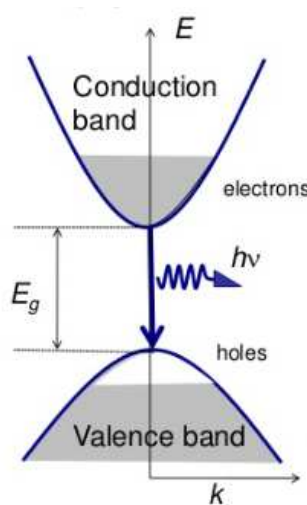
Figure 11 Electronic transition

The life-time of an electron before it recombines with the hole is ζ_{ij}.

The transition rate is calculated using Fermi's golden rule:

$$\zeta_{ij}^{-1} = \frac{2\pi}{\hbar} \left| <\psi_i | \mu | \psi_j> \right|^2 \tag{2.35}$$

where $\mu = e \cdot r$

When increase the transition rate ζ_{ij} increase the probability of the transition.

In this case, the electron transition occurs from the CB to the hole in the VB, so i will be the electron and j will be the hole. The equation (1.17) changes to:

$$\zeta_{e-h}^{-1} = \frac{2\pi}{\hbar} | <\psi_e | \mu | \psi_h> |^2 \tag{2.36}$$

By the approximation of the envelope function approx.:

$$<\psi_e | \mu | \psi_h> \simeq <S | \mu | P><f_e | f_h> + <S | P><f_e | \mu | f_h> \tag{2.37}$$

where $\psi_e = f_e | S >$, $\psi_h = f_h | P >$ being f_e and f_h the envelope function and $<S | P>$ the Bloch functions of conduction and valence band respectively.[4]

Since $<S | P> = 0$ by symmetry, with the equations (1.18) and (1.19) we obtain:

$$\zeta_{e-h}^{-1} = \frac{2\pi}{\hbar} <S | \mu | P> | <f_e | f_h> |^2 = k | <f_e | f_h> |^2 \tag{2.38}$$

One can see from equation (2.22) that the electron-hole recombination rate ζ_{e-h}^{-1} is proportional to the electron-hole overlap $<f_e | f_h>$. We will study how the different size and shape of NPLs can be used to enhance recombination rates based on such a fact.

3. RESULTS

In this section we are going to show the results obtained for the energy and for the transition probability rates. They have been calculated by two methods, as we have explained that we would do it, the analytical and the numerical method. Finally we will work in the system with an electric field, which is solve numerically, the only possible way.

3.1 ENERGIES

Firstly, we represent the energy as function of the length of NPLs by analytical method. We take CdSe and CdS NPL with typical experimental dimensions.[6]
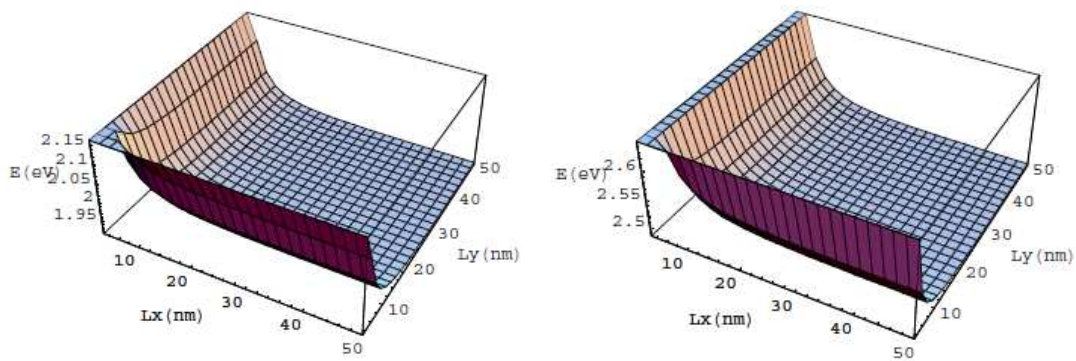
Figure 12. On the left the representation of the energy against to Lx and Ly for CdSe's NPL and on the right the representation of the energy against to Lx and Ly for CdS's NPL.

Since it is possible to observe in the Figure 12, for values of Lx and Ly higher than 15nm the variation of the energy is very slightly sensitive at the rate of dimensions, whereas for small values of Lx and Ly (between 0nm and 10nm), the energy experiences sudden changes. It is very sensitive managing to overcome in both cases the energy of the bulk semiconductor gap which is 1.732eV for CdSe NPLs and 2.482eV for CdS ones.

The graph is like a ramp that it begins with a very big slope for small NPLs and that keeps a slope increasingly small with the increase of the NPLs' size.

Then, we proceed to calculate the energies both for the analytical method and for the numerical one. We obtain the first four energies but the one that we are really interested in is the grown state energy, since; it is in the grown state of the valence band where the electron-hole recombination occurs.

The energy is obtained in the numerical method from the Hamiltonian matrix described in section 2.2. We program in Mathematica (look at annex) and we obtain the energy in-plane, then we add the vertical motion by the analytical solution $(z) = \sqrt{\frac{2}{L_z}} Sin(\frac{\pi}{L_z})$.

In the analytical method to obtain the energy we employ the equation:

$$E_{xyz} = \frac{\hbar^2 \pi^2}{2mL_x^2} n_x^2 + \frac{\hbar^2 \pi^2}{2mL_y^2} n_y^2 + \frac{\hbar^2 \pi^2}{2mL_z^2} n_z^2 \qquad (3.1)$$

Considering the dimensions for the NPL Lx=Ly= 10nm and np=50, the first four energies in each method are:

Analytical energies (eV)	Numerical energies (eV)
E_{111}= 0.233043	E_1= 0.23302
E_{121}= 0.333695	E_2= 0.3335
E_{211}= 0.333695	E_3= 0.3335
E_{221}= 0.434347	E_4= 0.43398

Table1. The analytical and numerical energies for the first four states.

Note that the energies are similar in both methods but they are not exactly the same. This is because, while the analytical method with the application of the Equation (3.1) only gives rise to a unique solution for each n_x, n_y and n_z, by the method of finite differences the result changes significantly with the number of points. The higher number of points are taken the more accurate the result.

Moreover, we can see that the second and third energies are equal for each method, they are degenerated.

Then we calculate the energy only for the grown state both for the numerical method and for the analytical one. For it we keep Lx's value constant to 10nm and we are giving to Ly different values to see the trend of the changes in the energy.

| Ly (nm)	E_{analyt} (eV)	E_{num} (eV)
5	0.333695	0.331998
10	0.233043	0.232364
15	0,214403	0.213913
20	0,207879	0.207865
25	0,20956	0.207455
30	0,20322	0.204466
35	0,202231	0.202843

Table 2. The energies for the ground state.

Again we see that the energies are very similar but are not exactly equal, they differ from the third decimal, and the reason is the same that we have said in the page before.

We see that the energy diminishes often that we enlarge Ly, nevertheless, from values of Ly superiors to 10nm the energy changes smaller (as we have seen in it Figures 12).

We represent the energies in Figure 13 where we can see in red the curve for the energy for the analytical method and in blue for the numerical method.

As expected seeing the values in the second table, the curves were going to be almost coincidental. If we compare it with the graphs of the Figure 12, we find seemed between these curves and the profile of the graphs, for small values of Ly the energy results higher and with the increase of the Ly the energy decreases till values almost constant.
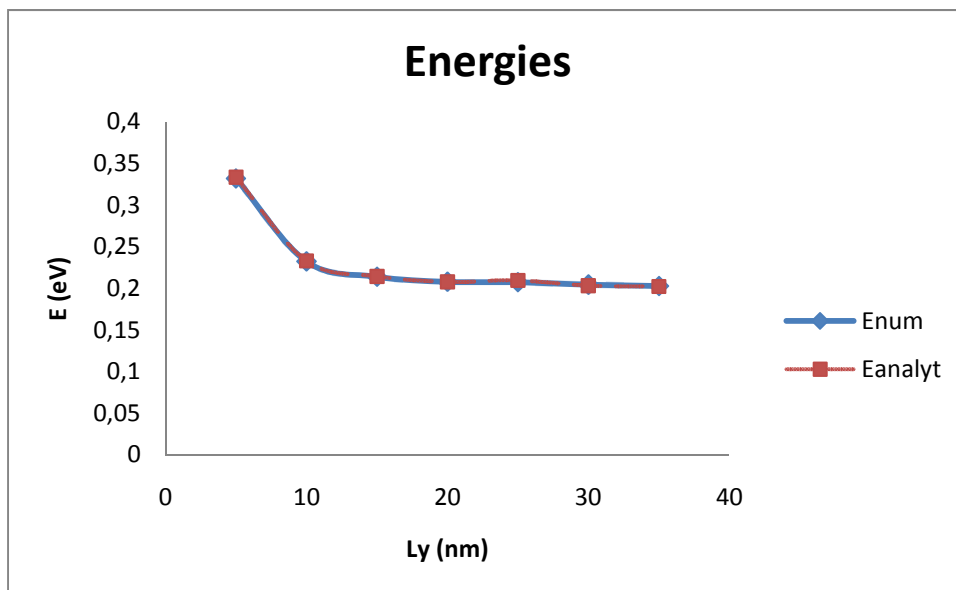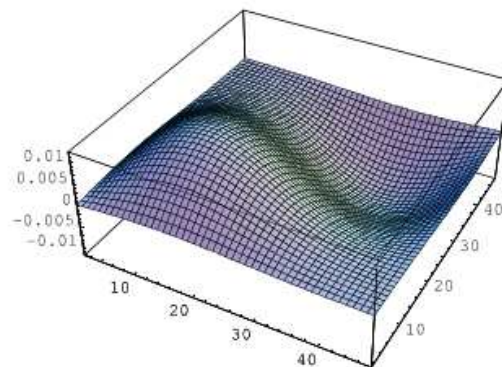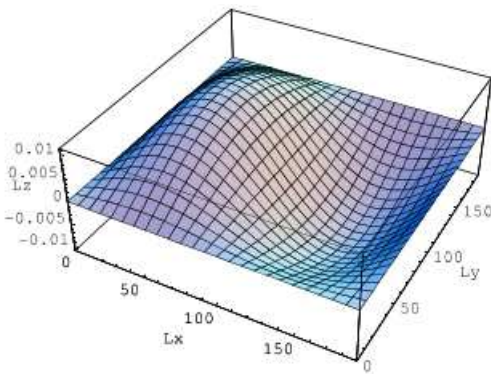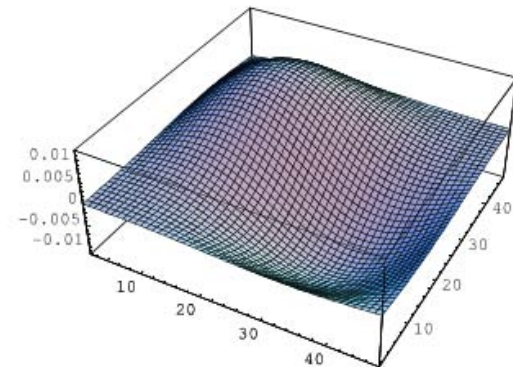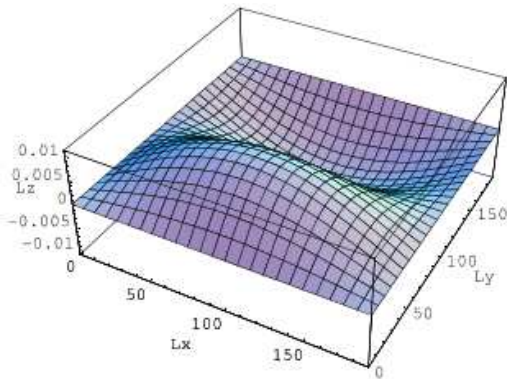

Figure 13. The ground state energies for different values of Ly.

3.2 WAVEFUNCTIONS

In addition, we have represented the wavefunctions for the first four states with each method, in the numerical method by the command eigenvectors and in the analytical method by the equation (look at annex):

$$\psi_{gs}{}^{CB} = \psi_{gs}{}^{VB} = \sqrt{\frac{2}{L_x}} Sin(\frac{\pi x}{L_x} nx) \cdot \sqrt{\frac{2}{L_y}} Sin(\frac{\pi y}{L_y} ny) \cdot \sqrt{\frac{2}{L_z}} Sin(\frac{\pi z}{L_z} nz)$$

(3.2)
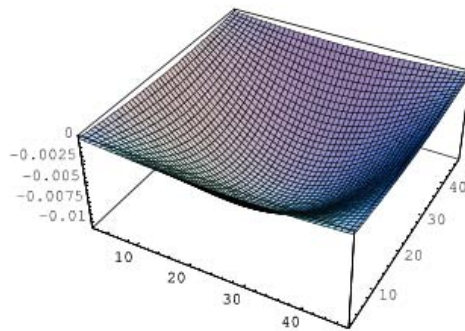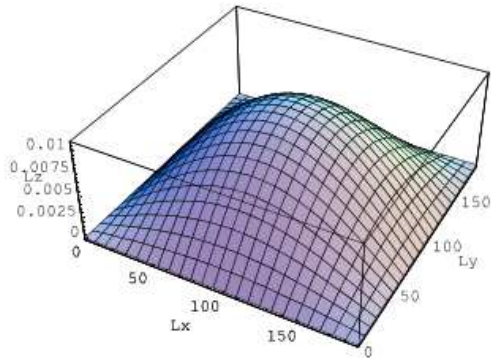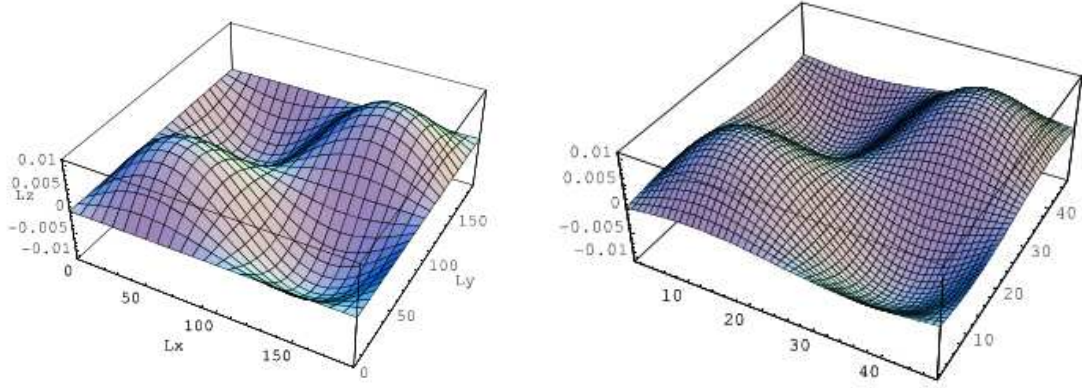
Figure 14. On the left the wavefunctions obtained by the analytical method and on the right the wavefunction by the numerical one.

We can observe two facts typically to simple sight: firstly we see that for the analytical method the representations contain in its surface less squares that those of the numerical method, this is because in the numerical method more points have been taken for the integration, increasing the number of points a more exact surface is obtained. On the other hand, there is the same number of nodes for every wavefunction in every method but them signs do not correspond, both are correct because it does not import the sign of the phase.

3.3 TRANSITION PROBABILITY RATE

Another point in this work has been to calculate the rate of the electronic transition probability. One more time, we have done that analytical and numerically.

As it has been explained in the section 2.4, at last we have to represent the equation:

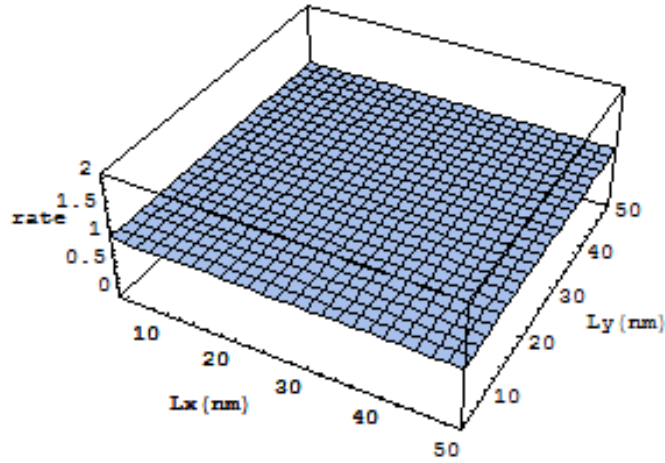$$\zeta_{e-h}{}^{-1} = k|{<}f_e|f_h{>}|^2 \qquad (3.3)$$

Were the electron-hole recombination rate is proportional to the electron-hole overlap and the term k, being k:

$$k = \frac{2\pi}{\hbar} {<}S|\boldsymbol{\mu}|P{>} \qquad (3.4)$$

According to the Equation (3.3) the electron-hole recombination rate is proportional to k.

If we represent it:

Figure 15. The electron-hole recombination rate $\zeta_{e-h}{}^{-1}$.

The Figure 15 is the same for both methods, because the function is the same (but obtained by different methods) and it only need to apply the Equation (3.3) with Mathematica (look at annex).

Looking at the Figure 15, we can conclude about the transition rate that they are constant for any Lx and Ly dimensions and its values is one, so the size of the NPLs will not affect its electron-hole recombination rates.

3.4 THE EFFECT OF AN ELECTRIC FIELD

We take two experimental values for F (0 and 100 kV/cm) [6] and calculate the ground energy, wavefunction and electron-hole recombination rate for each case.

F=0 (without field)		F=100 (with field)	
$E_{electron}$ **(eV)**	E_{hole} **(eV)**	$E_{electron}$ **(eV)**	E_{hole} **(eV)**
0.23302	24.5575	0.181115	24.5961
0.3335	24.525	0.281595	24.6211
0.3335	24.525	0.2855	24.633
0.43398	24.6075	0.385981	24.658

Table 3. Energies for electric field

As it is shown in Table 3, the electric field affect the system, its energies and its wavefuctions modifying them:
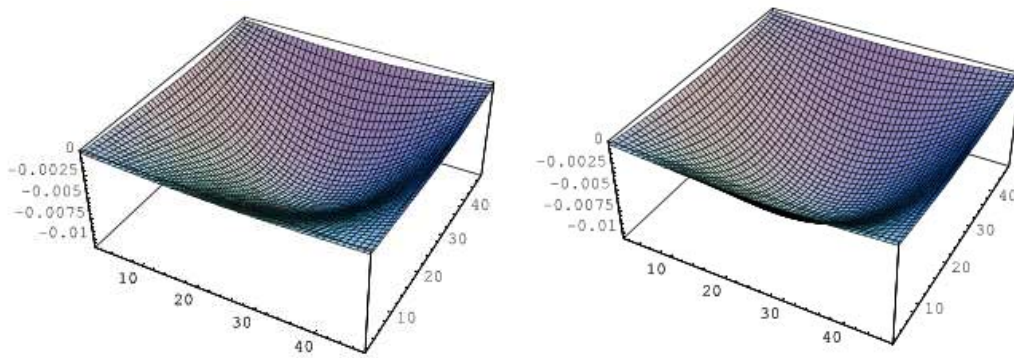
Figure 16. The representations of the wave functions for the electron F=0 and F=100 respectively.
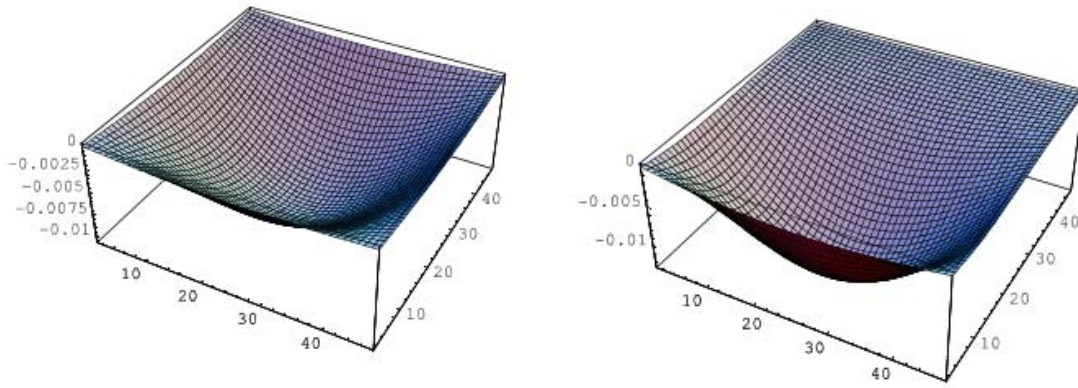
Figure 17. The representations of the wave functions for the hole F=0 and F=100 respectively.

There are differences between them, so there is an effect produced by the electric field, but with so many points it is difficult to appreciate, so we try with less points (this way is less exact is only to see the effect):
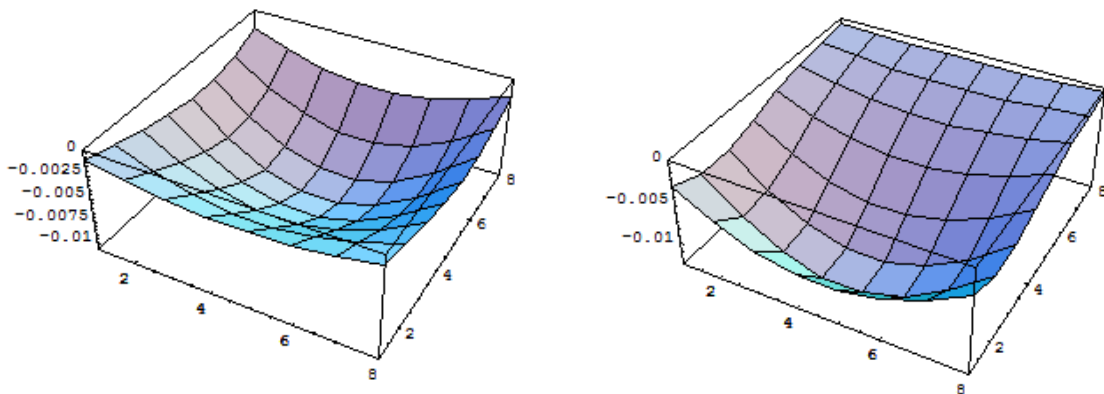
Figure 18. The representations of the wave functions for the electron and hole with electric field

Finally we cheek the electron-hole transition rate. Without camp we have proved that it is constant and its value is one, but when we add an electric field (F=100), the value is not constant and it is around 0.5.

4. CONCLUSIONS

To conclude this work we are going to speak about the concepts that we have extracted in clearly. In the introduction we were speaking about the properties of NPLs such as his short lifetime, his high luminescence or his pure color that become them in interesting subjects of study. For it we needed to simulate his optoelectronic response.

When we have studied the trend of the energy depending on the dimensions of the NPL, we have seen that, if we take the experimental values to define Lx and Ly, from values around 10nm we find a point of change in the trend of energy. While, for Lx and Ly minors than 10nm the energy suffers very sudden changes managing to reach high energies, if we take major dimensions of them, the energy suffers changes very small and as the L grows these changes become smaller.

Therefore, we can say that for the lengths of the sides Lx and Ly upper to 10nm, the energy is kept almost equal, for what it does not import the size of the NPL.

On the other hand, when we represent the electron-hole recombination rate we obtain in every case the same response. The transition rate is always the same and proportional to k Equation (3.3) and (3.4). It is independent of the size of the NPLs. So for any NPL we will obtain always the same transition probability.

When finally we add an electric field to the system, we perceive some changes in its energy, wavefunction, the field modifies them. Moreover, we get a lower transition probability rate than without electric field. It shows that the action of the electric field is to reduce overlap between electron and hole (directly proportional to the rate) and thus extend the lifetime of electrons in the conduction band.

Note that we have been using analytical and numerical methods until now but for the last case we have only used the numerical method because only with it we can work with more complex systems as is the case of the electric field.

Personally, this work has allowed me to learn more things and with more depth of nanotechnology. Also I have dealt with researches about quantum mechanics and scientific articles in which I have supports my work. And I have improve my knowledge in the use of Mathematica too.

5. BIBLIOGRAPHY

(1) Lhuiller, E.; Pedetti, S.; Ithurria, S.; Nadal, B.; Heuclin, H.; Dubertret, B. Two-Dimensional Colloidal Semiconductors: Synthesis, Spectroscopy, and Apllications. Acc. Chem. **2015**, 48, 22-30.

(2) McKinney, D.; Faster Colloidal Fluorescence emitters: Nanoplatelets. Phy.org **2011**

(3) Dayneko, S.; Linkov, P.; Martynov, I.; Cristyakov, A. Photoconductivity of composites based on CdSe quantum dots and low-band-gap polymers. **2016**

(4) Achtstein, AW.; Scott, R.; Kickhöfel, S.; Jagsch, ST.; Christodoulou, s.; Bertrand, G.; Prudnikau, A.; Antanovich, A.; Artemyev, M.; Moreels, I.; Schliwa, A.; Woggon, U. p-State Luminescence in CdSe Nanoplatelets: Role of Lateral Confinement and a Longitudinal Optical Phonon Bottleneck. Phys. Reb. **2016**

(5) Valenzuela, C.; Química general. Salamanca, **1995**

(6) Kovalenco, M. *et. al.* Prospects of Nanoscience with Nanocrystals. *ACS Nano*, **2015** *9*, pp 1012–1057

6. ANNEXES

VALUES FOR CdSe AND CdS NANOPLATELETS

Cristalline structure

Wurtzite

$a = 4.2999\text{Å}$

$c = 7.0109\text{Å}$

lx= ly = 3-50 nm

lz = 4.5 nm

CdSe

$E_g = 1.732 eV$

$m_n *= 0.112 m_o$

$m_p \perp *= 0.45 m_o$

$m_{p\parallel} * \geq 1 m_o$

CdS

$E_g = 2.482 \ eV$

$m_n *= 0.25 m_o$

$m_p \perp *= 0.7 m_o$

$m_{p\parallel} * \geq 5 m_o$

q_e= -1

q_h= 1

---

1 bohr = 0.529 Å = 0.529·10⁻¹⁰ m

1 hartree = 27.2107 eV

In[108]:=

```
(* CdSe *)
Clear["Global`*"]
hb = 1;
m = 1;
mn = 0.112;
mpp = 0.45;
mp = 1;
```

In[114]:=

```
ban = 0.529;
```

In[115]:=

```
lz = 4.5;
```

In[116]:=

```
egap = 0.06365;
```

$$eh[lx\_, ly\_] := \frac{hb^2 * \pi^2}{2 * mpp * (\frac{lx*10}{ban})^2} + \frac{hb^2 * \pi^2}{2 * mpp * (\frac{ly*10}{ban})^2} + \frac{hb^2 * \pi^2}{2 * mp * (\frac{lz*10}{ban})^2};$$

In[118]:=

$$ee[lx\_, ly\_] := \frac{hb^2 * \pi^2}{2 * mn * (\frac{lx*10}{ban})^2} + \frac{hb^2 * \pi^2}{2 * mn * (\frac{ly*10}{ban})^2} + \frac{hb^2 * \pi^2}{2 * mn * (\frac{lz*10}{ban})^2};$$

```
(* La energia en hartree *)
```

In[119]:=

```
Ehart[lx_, ly_] := egap + eh[lx, ly] + ee[lx, ly]
```

```
(* La energia en eV *)
```

In[120]:=

```
Eev[lx_, ly_] := Ehart[lx, ly] * 27.2107
```

In[121]:=

```
Plot3D[Eev[lx, ly], {lx, 3, 50}, {ly, 3, 50}, AxesLabel → {"Lx(nm)", "Ly(nm)", "E(eV)"}]
```
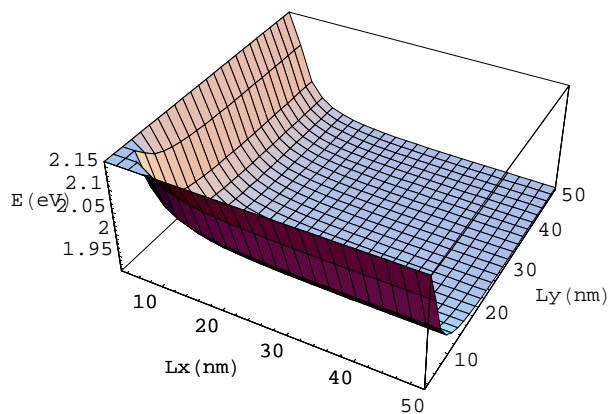

Out[121]=

```
- SurfaceGraphics -
```

```
In[93]:=  (* CdS *)
          Clear["Global`*"]
          hb = 1;
          m = 1;
          mn = 0.25;
          mpp = 0.7;
          mp = 5;
```

```
In[99]:=  ban = 0.529;
```

```
In[100]:=
          egap = 0.09121;
          lz = 4.5;
```

```
In[103]:=
```

$$eh[lx\_, ly\_] := \frac{hb^2 * \pi^2}{2 * mpp * (\frac{lx*10}{ban})^{\wedge}2} + \frac{hb^2 * \pi^2}{2 * mpp * (\frac{lx*10}{ban})^{\wedge}2} + \frac{hb^2 * \pi^2}{2 * mp * (\frac{lz*10}{ban})^{\wedge}2}$$

```
In[104]:=
```

$$ee[lx\_, ly\_] := \frac{hb^2 * \pi^2}{2 * mn * (\frac{lx*10}{ban})^{\wedge}2} + \frac{hb^2 * \pi^2}{2 * mn * (\frac{ly*10}{ban})^{\wedge}2} + \frac{hb^2 * \pi^2}{2 * mn * (\frac{lx*10}{ban})^{\wedge}2};$$

```
In[105]:=

          Ehart[lx_, ly_] := egap + eh[lx, ly] + ee[lx, ly]
```
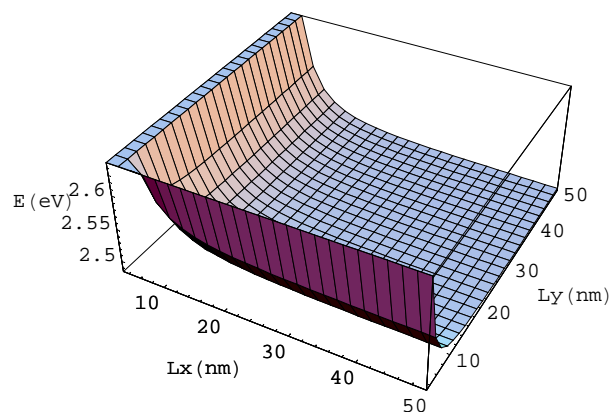
```
In[106]:=

          Eev[lx_, ly_] := Ehart[lx, ly] * 27.2107
```

```
In[107]:=
          Plot3D[Eev[lx, ly], {lx, 3, 50}, {ly, 3, 50}, AxesLabel → {"Lx(nm)", "Ly(nm)", "E(eV)"}]
```



```
Out[107]=
          - SurfaceGraphics -
```

In[1]:= **(* ANALITICAL METHOD *)**

Clear["Global`*"]

In[2]:= **hb = 1;**
m = 1;
mn = 0.112;
mpp = 0.45;
mp = 1;
ban = 0.529;
lx = 100 / ban;
ly = 100 / ban;
lz = 85 (*ua*);
elv = 27.2107;

In[12]:= **egap = 0.064;**

(* we calculate the first four energies *)

In[13]:= **enexy11 =** $\left(\dfrac{hb^2 * \pi^2}{2 * mn * lx^2} + \dfrac{hb^2 * \pi^2}{2 * mn * ly^2} \right)$ *** elv**

Out[13]= 0.0671016

In[14]:= **enexy12 =** $\left(\dfrac{hb^2 * \pi^2 * 1^2}{2 * mn * lx^2} + \dfrac{hb^2 * \pi^2 * 2^2}{2 * mn * ly^2} \right)$ *** elv**

Out[14]= 0.167754

In[16]:= **enexy21 =** $\left(\dfrac{hb^2 * \pi^2 * 2^2}{2 * mn * lx^2} + \dfrac{hb^2 * \pi^2 * 1^2}{2 * mn * ly^2} \right)$ *** elv**

Out[16]= 0.167754

In[17]:= **enexy22 =** $\left(\dfrac{hb^2 * \pi^2 * 2^2}{2 * mn * lx^2} + \dfrac{hb^2 * \pi^2 * 2^2}{2 * mn * ly^2} \right)$ *** elv**

Out[17]= 0.268406

(* and we calculate the subtractions *)

In[18]:= **enexy12 – enexy11**

Out[18]= 0.100652

In[19]:= **enexy21 – enexy12**

Out[19]= 0.

In[20]:= **enexy22 – enexy21**

Out[20]= 0.100652

(* we calculate and add z axis to xy z *)

In[21]:= **enez1 =** $\dfrac{hb^2 * \pi^2}{2 * mn * lz^2}$ *** elv**

Out[21]= 0.165941

In[22]:= **enexyz111 = enexy11 + enez1**

Out[22]= 0.233043

In[23]:= **enexyz121 = enexy12 + enez1**

Out[23]= 0.333695

In[25]:= **enexyz211 = enexy21 + enez1**

Out[25]= 0.333695

In[26]:= **enexyz221 = enexy22 + enez1**

Out[26]= 0.434347

In[27]:= **(* The new subtractions *)**

enexyz121 - enexyz111

Out[27]= 0.100652

In[28]:= **enexyz211 - enexyz121**

Out[28]= 0.

In[29]:= **enexyz221 - enexyz211**

Out[29]= 0.100652

(* We write the wavefunctions and represent them *)

$$\mathbf{f1[x\_]} := \sqrt{\frac{2}{lx}} * \mathbf{Sin}\left[\frac{\pi * x}{lx}\right];$$

$$\mathbf{f2[x\_]} := \sqrt{\frac{2}{lx}} * \mathbf{Sin}\left[\frac{2 * \pi * x}{lx}\right];$$

$$\mathbf{fu1[y\_]} := \sqrt{\frac{2}{ly}} * \mathbf{Sin}\left[\frac{\pi * y}{ly}\right];$$

$$\mathbf{fu2[y\_]} := \sqrt{\frac{2}{ly}} * \mathbf{Sin}\left[\frac{2 * \pi * y}{ly}\right];$$

(* Estado fundamental *)

f1[x_, y_] := f1[x] * fu1[y]

```
Plot3D[f1[x, y], {x, 0, lx}, {y, 0, ly}, AxesLabel → {"Lx", "Ly", "Lz"}]
```
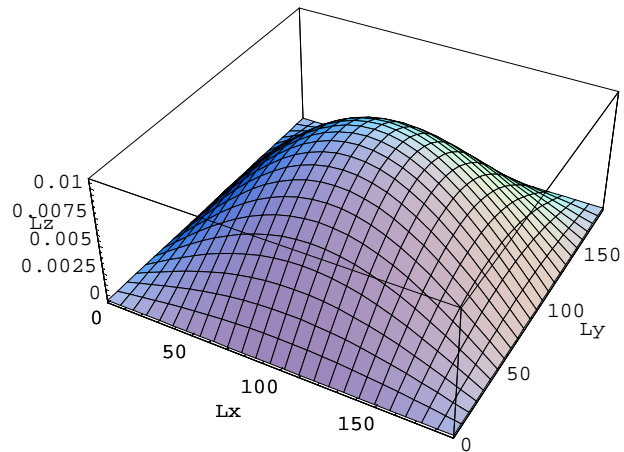

- SurfaceGraphics -

```
f2[x_, y_] := f1[x] * fu2[y]
```

```
Plot3D[f2[x, y], {x, 0, lx}, {y, 0, ly}, AxesLabel → {"Lx", "Ly", "Lz"}]
```


- SurfaceGraphics -

```
f3[x_, y_] := f2[x] * fu1[y]
```

Plot3D[f3[x, y], {x, 0, lx}, {y, 0, ly}, AxesLabel → {"Lx", "Ly", "Lz"}]

◾ SurfaceGraphics ◾

f4[x_, y_] := f2[x] * fu2[y]
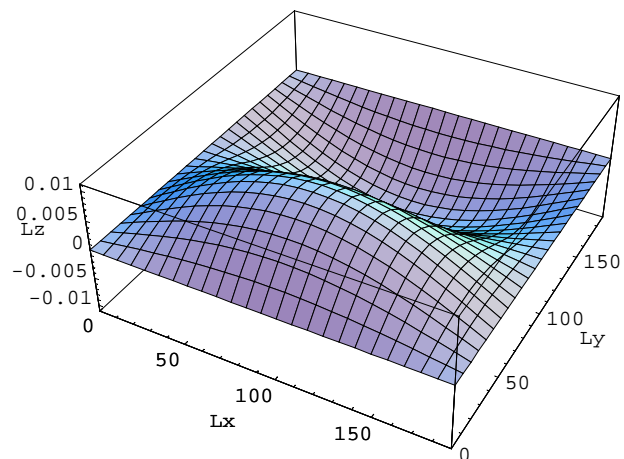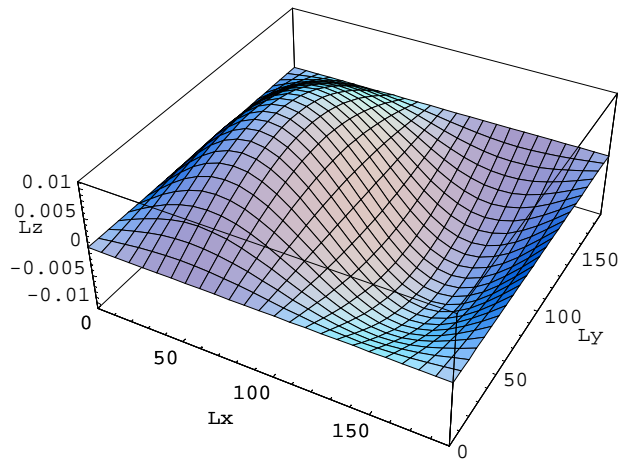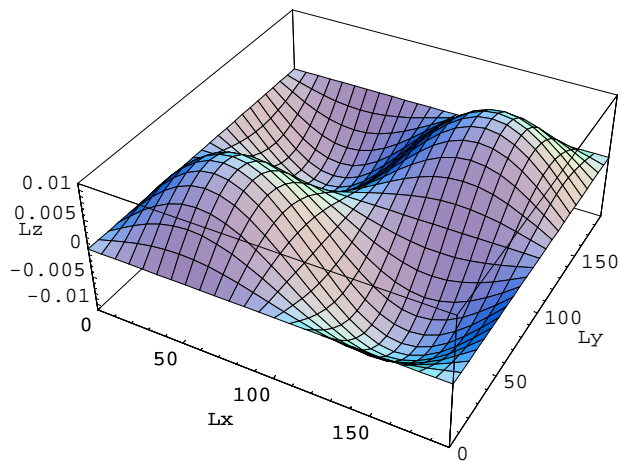
Plot3D[f4[x, y], {x, 0, lx}, {y, 0, ly}, AxesLabel → {"Lx", "Ly", "Lz"}]

◾ SurfaceGraphics ◾

```
In[1]:=  Clear["Gobal`*"]

In[2]:=  hb = 1;
         ban = 0.529;
         lx = 100 / ban;
         ly = 100 / ban;
         m = 0.112;
         elv = 27.2107;
         np = 50;
         hx = ——————— ;
               np - 1
         hy = ——————— ;
               np - 1
         dif = np - 2;
         fin = dif * dif;

In[13]:=  matH = Table[0, {fil, 1, fin}, {col, 1, fin}];
          For[fil = 1, fil ≤ fin, fil = fil + 1, matH[[fil, fil]] = b];
          For[fil = 1, fil ≤ fin - 1, fil = fil + 1, matH[[fil, fil + 1]] = aa];
          For[fil = dif, fil ≤ fin - 1, fil = fil + dif, matH[[fil, fil + 1]] = 0];
          For[fil = 2, fil ≤ fin, fil = fil + 1, matH[[fil, fil - 1]] = aa];
          For[fil = np - 1, fil ≤ fin, fil = fil + dif, matH[[fil, fil - 1]] = 0];
          For[fil = 1, fil ≤ fin - dif, fil = fil + 1, matH[[fil, fil + dif]] = a];
          For[fil = np - 1, fil ≤ fin, fil = fil + 1, matH[[fil, fil - dif]] = a]
          matH // MatrixForm;

In[22]:=  bx = ———————— ;
                m * hx²
          by = ———————— ;
                m * hy²
          b = bx + by;
               -hb²
          a = ———————— ;
              2 * m * hx²
                -hb²
          aa = ———————— ;
               2 * m * hy²

In[27]:=  (* Energies *)

In[28]:=  energ = Eigenvalues[matH, -4] // N

Out[28]=  {0.00985049, 0.00615782, 0.00615782, 0.00246516}

In[29]:=  energia1 = energ[[4]] * elv
          energia2 = energ[[3]] * elv
          energia3 = energ[[2]] * elv
          energia4 = energ[[1]] * elv

Out[29]=  0.0670786

Out[30]=  0.167559

Out[31]=  0.167559

Out[32]=  0.268039
```

In[33]:= **(* we add analitically the z-axis *)**

lz = 85;
mn = 0.112;
$$enez1 = \frac{hb^2 * \pi^2}{2 * mn * lz^2} * elv$$

Out[35]= 0.165941

In[36]:= **energia11 = energia1 + enez1**

Out[36]= 0.23302

In[38]:= **energia21 = energia2 + enez1**

Out[38]= 0.3335

In[40]:= **energia31 = energia3 + enez1**

Out[40]= 0.3335

In[42]:= **energia41 = energia4 + enez1**

Out[42]= 0.43398

In[43]:= **energia2 - energia1**
energia3 - energia2

Out[43]= 0.10048

Out[44]= 1.21014×10^{-14}

In[45]:= **energia4 - energia3**

Out[45]= 0.10048

(* Wavefunctions *)

In[46]:= **func = Eigenvectors[matH, -4];**

In[47]:= **func1 = func[[4]];**

In[48]:= **func2 = func[[3]];**
func3 = func[[2]];
func4 = func[[1]];

(* Represent the wavefunctions *)

In[51]:= **listabis = Table[{0, 0}, {i, 1, np - 2}, {j, 1, np - 2}];**
listax =
elem = 0;
For[i = 2, i ≤ np - 1, i = i + 1,
For[j = 2, j ≤ np - 1, j = j + 1,
elem = elem + 1; listabis[[i - 1, j - 1]] = func1[[elem]]]];

In[54]:= **ListPlot3D[listabis]**
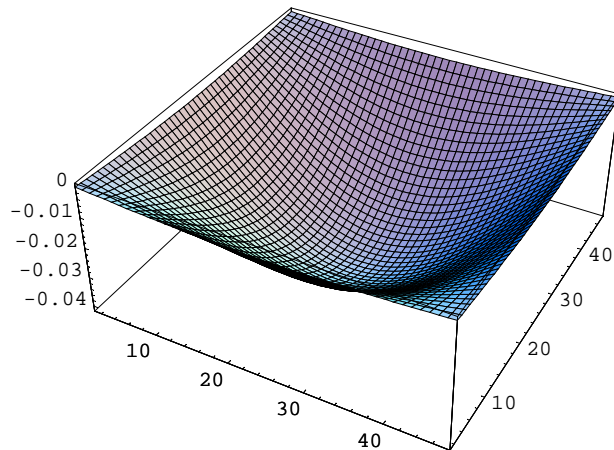
Out[54]= ◾ SurfaceGraphics ◾

In[55]:= **(* We normalize the function and return to represent *)**

  gij = Dot[func1, func1]

Out[55]= 1.

In[56]:= **int = hx * hy * gij**

Out[56]= 14.8832

In[58]:= **norm = $\dfrac{1}{\sqrt{\text{int}}}$**

Out[58]= 0.25921

In[59]:= **func1norm = norm * func1;**

In[60]:= **listabis = Table[{0, 0}, {i, 1, np - 2}, {j, 1, np - 2}];**

```
In[61]:=  elem = 0;
          For[i = 2, i ≤ np - 1, i = i + 1,
            For[j = 2, j ≤ np - 1, j = j + 1,
              elem = elem + 1; listabis[[i - 1, j - 1]] = func1norm[[elem]]]];
          ListPlot3D[listabis]
```
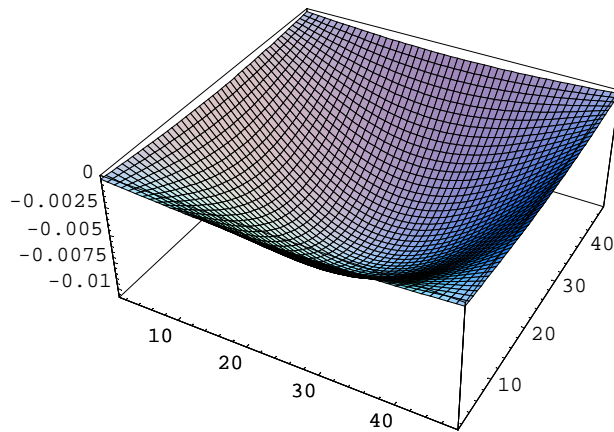


```
Out[63]=  ◼ SurfaceGraphics ◼
```

```
          (* Now we prove it *)
```

```
In[64]:=  hx * hy * Dot[func1norm, func1norm]
```

```
Out[64]=  1.
```

```
In[65]:=  gij2 = Dot[func2, func2]
```

```
Out[65]=  1.
```

```
In[66]:=  int2 = hx * hy * gij2
```

```
Out[66]=  14.8832
```

```
In[67]:=  norm2 = ─────────
                   √int2
```

```
Out[67]=  0.25921
```
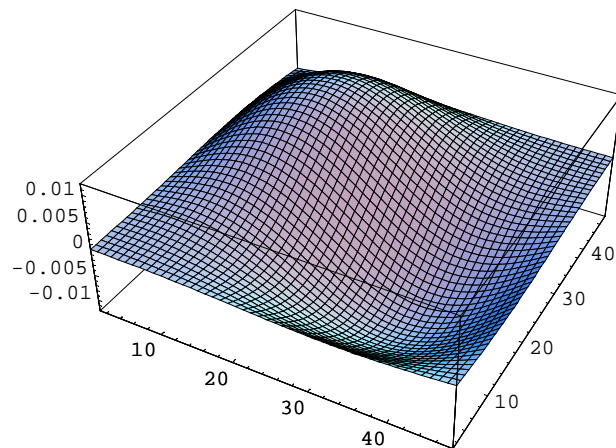
```
In[68]:=  func2norm = norm2 * func2;
```

```
In[69]:=  listabis2 = Table[{0, 0}, {i, 1, np - 2}, {j, 1, np - 2}];
```

```
In[70]:=  elem = 0;
          For[i = 2, i ≤ np - 1, i = i + 1,
            For[j = 2, j ≤ np - 1, j = j + 1,
              elem = elem + 1; listabis2[[i - 1, j - 1]] = func2norm[[elem]]]];
```

In[72]:= **ListPlot3D[listabis2]**

Out[72]= ▪ SurfaceGraphics ▪

In[73]:= **gij3 = Dot[func3, func3]**

Out[73]= 1.

In[74]:= **int3 = hx * hy * gij3**

Out[74]= 14.8832

In[75]:= **norm3 = $\dfrac{1}{\sqrt{\text{int3}}}$**

Out[75]= 0.25921

In[76]:= **func3norm = norm3 * func3;**

In[77]:= **listabis3 = Table[{0, 0}, {i, 1, np - 2}, {j, 1, np - 2}];**

In[78]:= **elem = 0;**
  For[i = 2, i ≤ np - 1, i = i + 1,
    For[j = 2, j ≤ np - 1, j = j + 1,
     elem = elem + 1; listabis3[[i - 1, j - 1]] = func3norm[[elem]]]];

In[80]:= **ListPlot3D[listabis3]**
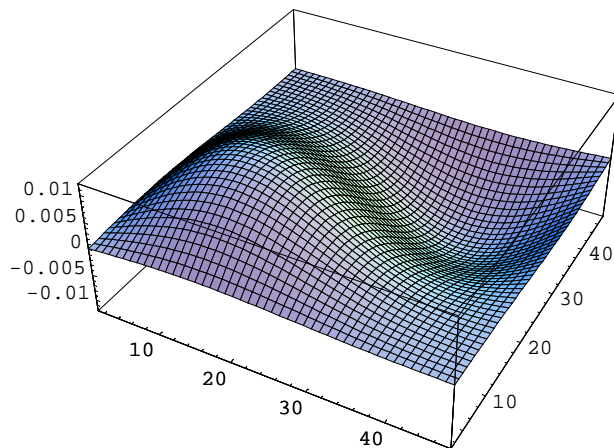
Out[80]= ‑ SurfaceGraphics ‑

In[81]:= **gij4 = Dot[func4, func4]**

Out[81]= 1.

In[82]:= **int4 = hx * hy * gij4**

Out[82]= 14.8832

In[83]:= **norm4 =** $\dfrac{1}{\sqrt{\texttt{int4}}}$

Out[83]= 0.25921

In[84]:= **func4norm = norm4 * func4;**

In[85]:= **listabis4 = Table[{0, 0}, {i, 1, np - 2}, {j, 1, np - 2}];**

In[86]:= **elem = 0;**
 For[i = 2, i ≤ np - 1, i = i + 1,
  For[j = 2, j ≤ np - 1, j = j + 1,
   elem = elem + 1; listabis4[[i - 1, j - 1]] = func4norm[[elem]]]];

In[88]:= **ListPlot3D[listabis4]**
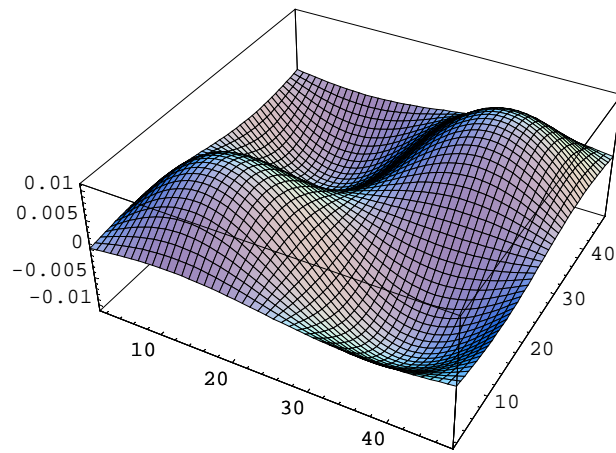
Out[88]= ▪ SurfaceGraphics ▪

```
Clear["Global`*"]
```

$$f1[x\_] := \sqrt{\frac{2}{lx}} * Sin\left[\frac{\pi * x}{lx}\right];$$

$$ful[y\_] := \sqrt{\frac{2}{ly}} * Sin\left[\frac{\pi * y}{ly}\right];$$

$$fun1[z\_] := \sqrt{\frac{2}{lz}} * Sin\left[\frac{\pi * z}{lz}\right];$$

```
lz = 4.5;

felec[x_, y_, z] := f1[x] * ful[y] * fun1[z]

fhueco[x_, y_, z_] := f1[x] * ful[y] * fun1[z]

rate[lx_, ly_] :=
  (Integrate[felec[x, y, z] * fhueco[x, y, z], {x, 0, lx}, {y, 0, ly}, {z, 0, lz}])^2

Plot3D[rate[lx, ly], {lx, 3, 50}, {ly, 3, 50}, PlotRange → {0, 2}]
```
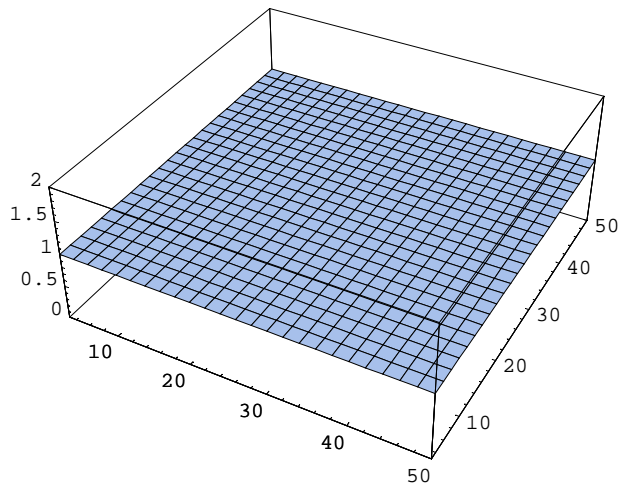

- SurfaceGraphics -

```
Clear["Global`*"]

func1norm = {0.0012376248959005888`,
   0.0023259739639574398`, 0.0031337762442413164`, 0.003563598859858037`,
   0.0035635988598580488`, 0.0031337762442413242`, 0.0023259739639574567`,
   0.0012376248959006099`, 0.002325973963957417`, 0.004371401140141917`,
   0.005889572823815496`, 0.006697375104099385`, 0.0066973751040993925`,
   0.00588957282381551`, 0.004371401140141935`, 0.002325973963957458`,
   0.0031337762442413164`, 0.005889572823815496`, 0.007934999999999987`,
   0.009023349068056841`, 0.009023349068056864`, 0.007935000000000008`,
   0.005889572823815529`, 0.0031337762442413338`, 0.0035635988598580522`,
   0.006697375104099375`, 0.009023349068056838`, 0.010260973963957462`,
   0.010260973963957467`, 0.009023349068056869`, 0.006697375104099388`,
   0.0035635988598580687`, 0.0035635988598580496`, 0.0066973751040994`,
   0.009023349068056874`, 0.010260973963957459`, 0.010260973963957455`,
   0.009023349068056867`, 0.0066973751040993726`, 0.0035635988598580574`,
   0.003133776244241319`, 0.005889572823815527`, 0.007935000000000011`,
   0.009023349068056869`, 0.009023349068056857`, 0.007935000000000003`,
   0.005889572823815518`, 0.0031337762442413177`, 0.0023259739639574614`,
   0.004371401140141946`, 0.005889572823815532`, 0.006697375104099391`,
   0.0066973751040993856`, 0.005889572823815518`, 0.004371401140141925`,
   0.0023259739639574562`, 0.0012376248959006066`, 0.00232597396395746`,
   0.0031337762442413307`, 0.0035635988598580652`, 0.003563598859858062`,
   0.003133776244241313`, 0.0023259739639574545`, 0.0012376248959006042`};

lz = 4.5;
np = 10;
ban = 0.529;
lx = 100 / ban;
ly = 100 / ban;
hx = lx / (np - 1);
hy = ly / (np - 1);
rate[lx_, ly_] :=
  ((hx * hy * Dot[func1norm, func1norm]) * ∫₀^lz (√(2/lz) * Sin[(π * z)/lz] * √(2/lz) * Sin[(π * z)/lz]) ⅆz)^:
```

```
Plot3D[rate[x, y], {lx, 3, 50}, {ly, 3, 50}]
```
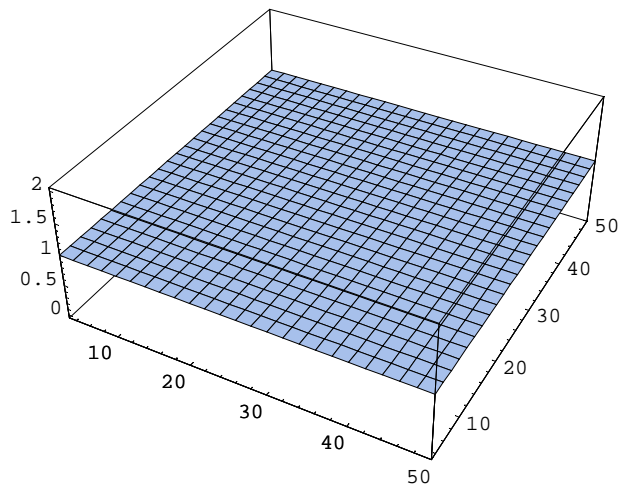
- SurfaceGraphics -

$$fz[z\_] := \sqrt{\frac{2}{lz}} * Sin\left[\frac{\pi * z}{lz}\right];$$

```
rate[lx_, ly_] :=
  ((hx * hy * Dot[func1norm, func1norm]) * Integrate[fz[z] * fz[z], {z, 0, lz}]) ^ 2
```

```
Plot3D[rate[x, y], {lx, 3, 50}, {ly, 3, 50}]
```
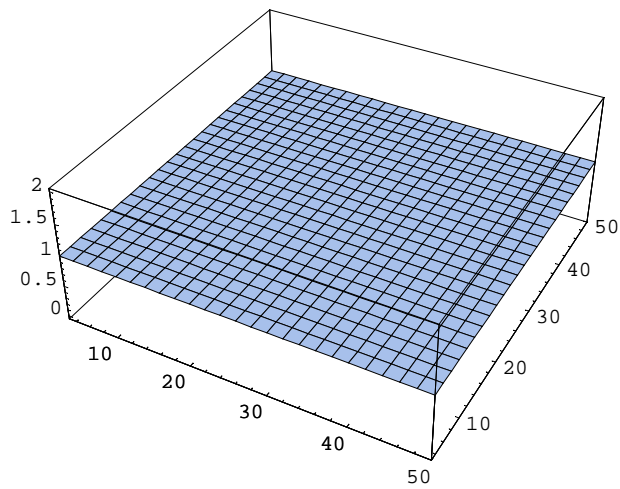
- SurfaceGraphics -

```
In[1]:= Clear["Gobal`*"]

In[2]:= hb = 1;
       ban = 0.529;
       lx = 100 / ban;
       ly = 100 / ban;
       m = 0.112;
       elv = 27.2107;
       np = 50;
              lx
       hx = ──────── ;
            np - 1
              ly
       hy = ──────── ;
            np - 1
       dif = np - 2;
       fin = dif * dif;

In[13]:= (* we define the Hamiltonian matrix for the electron *)

In[14]:= matH = Table[0, {fil, 1, fin}, {col, 1, fin}];
        fil = 1;
        For[k = 1, k ≤ dif, k = k + 1,
          For[fil = fil, fil ≤ k * dif, fil = fil + 1, matH[[fil, fil]] = k * w]];
        For[fil = 1, fil ≤ fin - 1, fil = fil + 1, matH[[fil, fil + 1]] = aa];
        For[fil = dif, fil ≤ fin - 1, fil = fil + dif, matH[[fil, fil + 1]] = 0];
        For[fil = 2, fil ≤ fin, fil = fil + 1, matH[[fil, fil - 1]] = aa];
        For[fil = np - 1, fil ≤ fin, fil = fil + dif, matH[[fil, fil - 1]] = 0];
        For[fil = 1, fil ≤ fin - dif, fil = fil + 1, matH[[fil, fil + dif]] = a];
        For[fil = np - 1, fil ≤ fin, fil = fil + 1, matH[[fil, fil - dif]] = a]
        matH // MatrixForm;

In[24]:= matdebH = Table[If[fil == col, b, 0], {fil, 1, fin}, {col, 1, fin}];
        matdebH // MatrixForm;

In[26]:= matrizH = matH + matdebH;
        matrizH // MatrixForm;

                hb²
In[28]:= bx = ──────── ;
              m * hx²
                hb²
        by = ──────── ;
             m * hy²
        b = bx + by;
              -hb²
        a = ─────────── ;
            2 * m * hx²
               -hb²
        aa = ─────────── ;
             2 * m * hy²
        w = qe * F * hx;

In[34]:= qe = -1;
        F = 0;

        (* Now we calculate the energy in xy for the electron *)

In[36]:= energiasxy = Eigenvalues[matrizH, -4] // N

Out[36]= {0.00985049, 0.00615782, 0.00615782, 0.00246516}
```

```
In[37]:=  enexy1 = energiasxy[[4]] * elv
          enexy2 = energiasxy[[3]] * elv
          enexy3 = energiasxy[[2]] * elv
          enexy4 = energiasxy[[1]] * elv
```

```
Out[37]=  0.0670786
```

```
Out[38]=  0.167559
```

```
Out[39]=  0.167559
```

```
Out[40]=  0.268039
```

```
          (* Now we calculate the energy z and add t to the energy in xy for the electron *)
```

```
In[41]:=  lz = 85;
          mn = 0.112;
          enez1 = ─────────── * elv;
                  2 * mn * lz ^ 2
```

$$enez1 = \frac{hb^2 * \pi^2}{2 * mn * lz\text{^}2} * elv;$$

```
In[45]:=  enexyz1 = enexy1 + enez1
```

```
Out[45]=  0.23302
```

```
In[47]:=  enexyz2 = enexy2 + enez1
```

```
Out[47]=  0.3335
```

```
In[49]:=  enexyz3 = enexy3 + enez1
```

```
Out[49]=  0.3335
```

```
In[51]:= enexyz4 = enexy4 + enez1

Out[51]= 0.43398
```

(* we define the Hamiltonian matrix for the hole *)

```
In[52]:= mathuecoH = Table[0, {fil, 1, fin}, {col, 1, fin}];
        fil = 1;
        For[k = 1, k ≤ dif, k = k + 1,
           For[fil = fil, fil ≤ k * dif, fil = fil + 1, mathuecoH[[fil, fil]] = k * v]];
        For[fil = 1, fil ≤ fin - 1, fil = fil + 1, mathuecoH[[fil, fil + 1]] = oo];
        For[fil = dif, fil ≤ fin - 1, fil = fil + dif, mathuecoH[[fil, fil + 1]] = 0];
        For[fil = 2, fil ≤ fin, fil = fil + 1, mathuecoH[[fil, fil - 1]] = oo];
        For[fil = np - 1, fil ≤ fin, fil = fil + dif, mathuecoH[[fil, fil - 1]] = 0];
        For[fil = 1, fil ≤ fin - dif, fil = fil + 1, mathuecoH[[fil, fil + dif]] = o];
        For[fil = np - 1, fil ≤ fin, fil = fil + 1, mathuecoH[[fil, fil - dif]] = o]
        mathuecoH // MatrixForm;
```

```
In[64]:= matdepH = Table[If[fil == col, p, 0], {fil, 1, fin}, {col, 1, fin}];
        matdepH // MatrixForm;
```

```
In[66]:= matrizH2 = mathuecoH + matdepH;
        matrizH2 // MatrixForm;
```

```
In[86]:= mhz = 1;
        mh = 0.45;
        qh = 1;
```

```
In[89]:= px = hb²/(mh * hx²);

        py = hb²/(mh * hy²);

        p = bx + by;

        o = -hb²/(2 * mh * hx²);

        oo = -hb²/(2 * mh * hy²);

        v = qh * F * hx;
```

(* Now we calculate the energy in xy for the hole *)

```
In[95]:= energiashuecoxy = Eigenvalues[matrizH2, -4] // N

Out[95]= {0.903649, 0.902729, 0.902729, 0.90181}
```

```
In[96]:= enhxy1 = energiashuecoxy[[4]] * elv
         enhxy2 = energiashuecoxy[[3]] * elv
         enhxy3 = energiashuecoxy[[2]] * elv
         enhxy4 = energiashuecoxy[[1]] * elv
```

```
Out[96]= 24.5389
```

```
Out[97]= 24.5639
```

```
Out[98]= 24.5639
```

```
Out[99]= 24.5889
```

```
(* Now we calculate the energy z and add t to the energy in xy for the hole *)
```

```
In[101]:=
```

$$enhz1 = \frac{hb^2 * \pi^2}{2 * mhz * lz^2} * elv;$$

```
In[106]:=
         (* The total energy *)

         enhxyz1 = enhxy1 + enhz1
         enhxyz2 = enhxy2 + enhz1
         enhxyz3 = enhxy3 + enhz1
         enhxyz4 = enhxy4 + enhz1
```

```
Out[106]=
         24.5575
```

```
Out[107]=
         24.5825
```

```
Out[108]=
         24.5825
```

```
Out[109]=
         24.6075
```

```
In[110]:=
         energia1 = enexyz1 + enhxyz1
```

```
Out[110]=
         24.7905
```

```
In[111]:=
         energia2 = enexyz2 + enhxyz2
```

```
Out[111]=
         24.916
```

```
In[112]:=
         energia3 = enexyz3 + enhxyz3
```

```
Out[112]=
         24.916
```

```
In[113]:=
        energia4 = enexyz4 + enhxyz4
Out[113]=
        25.0415

In[114]:=

        (* WAVEFUNCTIONS *)
        (* For holes *)

        holefunc = Eigenvectors[matrizH2, -4];

In[115]:=
        funchole1 = holefunc[[4]];
        funchole2 = holefunc[[3]];
        funchole3 = holefunc[[2]];
        funchole4 = holefunc[[1]];

        (* we normalize the functions *)

        gij = Dot[funchole1, funchole1]
Out[119]=
        1.

In[120]:=
        int = hx * hy * gij
Out[120]=
        14.8832

In[122]:=
        norm = 1/√int
Out[122]=
        0.25921

In[123]:=
        funchole1norm = norm * funchole1;

        (* we can prove it *)
```

```
In[124]:=
        hx * hy * Dot[funchole1norm, funchole1norm]
```

```
Out[124]=
        1.
```

```
        (* For electrons*)
```

```
In[125]:=
        elecfunc = Eigenvectors[matrizH, -4];
```

```
In[126]:=
        funcelec1 = elecfunc[[4]];
        funcelec2 = elecfunc[[3]];
        funcelec3 = elecfunc[[2]];
        funcelec4 = elecfunc[[1]];
```

```
In[130]:=
        (* now we normalize the functions for electrons *)

        jij = Dot[funcelec1, funcelec1]
```

```
Out[130]=
        1.
```

```
In[131]:=
        integral = hx * hy * jij
```

```
Out[131]=
        14.8832
```

```
In[132]:=
        cten = ─────────────
                √integral
```

```
Out[132]=
        0.25921
```

```
In[133]:=
        funcelec1norm = cten * funcelec1;
```

```
In[134]:=
        (* prove it *)

        hx * hy * Dot[funcelec1norm, funcelec1norm]
```

```
Out[134]=
        1.
```
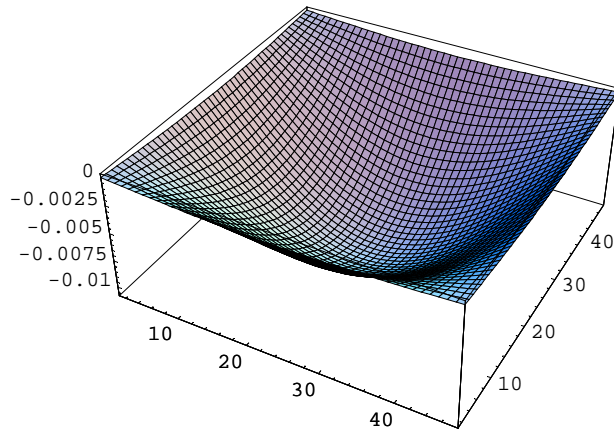
```
        (* we build a list with the funtion for the ground state *)
```

```
In[135]:=
        electronlist1 = Table[{0, 0}, {i, 1, np - 2}, {j, 1, np - 2}];
        listax =
          elem = 0;
        For[i = 2, i ≤ np - 1, i = i + 1,
          For[j = 2, j ≤ np - 1, j = j + 1,
           elem = elem + 1; electronlist1[[i - 1, j - 1]] = funcelec1norm[[elem]]]];
```

In[138]:=

```
ListPlot3D[electronlist1]
```
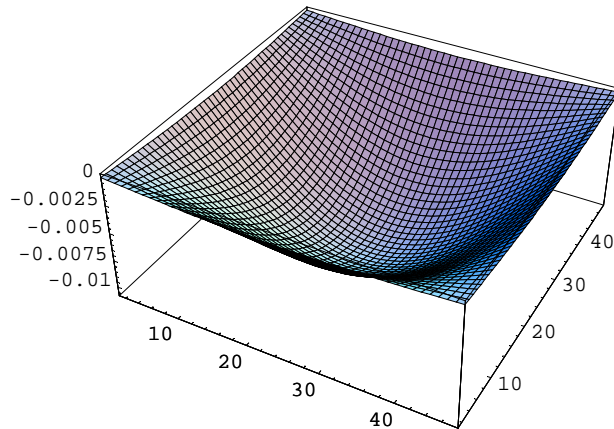

Out[138]=

- SurfaceGraphics -

In[139]:=

```
holelist1 = Table[{0, 0}, {i, 1, np - 2}, {j, 1, np - 2}];
listax =
  elem = 0;
For[i = 2, i ≤ np - 1, i = i + 1,
  For[j = 2, j ≤ np - 1, j = j + 1,
   elem = elem + 1; holelist1[[i - 1, j - 1]] = funchole1norm[[elem]]]];
```

In[142]:=
ListPlot3D[holelist1]

Out[142]=
▪ SurfaceGraphics ▪

In[143]:=
(* Now we calculate the rate of the transition probability *)

$$fz[z\_] := \sqrt{\frac{2}{lz}} * Sin\left[\frac{\pi * z}{lz}\right];$$

rate[lx_, ly_] :=
 ((hx * hy * Dot[funcelec1norm, funchole1norm]) * Integrate[fz[z] * fz[z], {z, 0, lz}]) ^ 2

In[145]:=
Plot3D[rate[lx, ly], {lx, 3, 50}, {ly, 3, 50}, AxesLabel → {"Lx(nm)", "Ly(nm)", "rate"}]
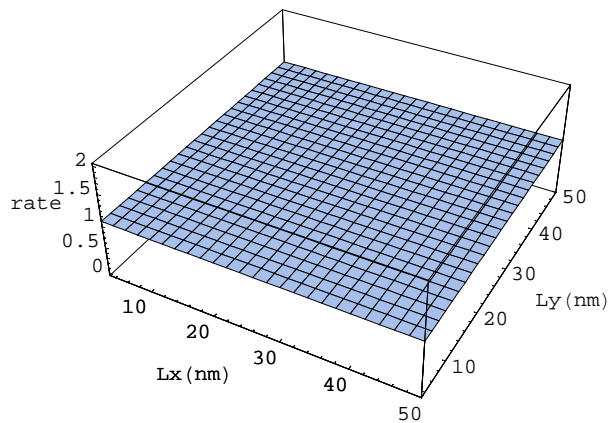
Out[145]=
▪ SurfaceGraphics ▪

```
In[1]:= Clear["Gobal`*"]

In[2]:= hb = 1;
        ban = 0.529;
        lx = 100 / ban;
        ly = 100 / ban;
        m = 0.112;
        elv = 27.2107;
        np = 50;
             lx
        hx = ────── ;
            np - 1
             ly
        hy = ────── ;
            np - 1
        dif = np - 2;
        fin = dif * dif;

        (* we define the Hamiltonian matrix for the electron *)

In[13]:= matH = Table[0, {fil, 1, fin}, {col, 1, fin}];
         fil = 1;
         For[k = 1, k ≤ dif, k = k + 1,
           For[fil = fil, fil ≤ k * dif, fil = fil + 1, matH[[fil, fil]] = k * w]];
         For[fil = 1, fil ≤ fin - 1, fil = fil + 1, matH[[fil, fil + 1]] = aa];
         For[fil = dif, fil ≤ fin - 1, fil = fil + dif, matH[[fil, fil + 1]] = 0];
         For[fil = 2, fil ≤ fin, fil = fil + 1, matH[[fil, fil - 1]] = aa];
         For[fil = np - 1, fil ≤ fin, fil = fil + dif, matH[[fil, fil - 1]] = 0];
         For[fil = 1, fil ≤ fin - dif, fil = fil + 1, matH[[fil, fil + dif]] = a];
         For[fil = np - 1, fil ≤ fin, fil = fil + 1, matH[[fil, fil - dif]] = a]
         matH // MatrixForm;

In[23]:= matdebH = Table[If[fil == col, b, 0], {fil, 1, fin}, {col, 1, fin}];
         matdebH // MatrixForm;

In[25]:= matrizH = matH + matdebH;
         matrizH // MatrixForm;

              hb²
In[27]:= bx = ────── ;
             m * hx²
              hb²
         by = ────── ;
             m * hy²
         b = bx + by;
              -hb²
         a = ──────── ;
            2 * m * hx²
              -hb²
         aa = ──────── ;
             2 * m * hy²
         w = qe * F * hx;

In[33]:= qe = -1;
         F = 1.9 * 10⁻⁵;

        (* Now we calculate the energy in xy for the electron *)

In[35]:= energiasxy = Eigenvalues[matrizH, -4] // N

Out[35]= {0.00808651, 0.00439384, 0.00425031, 0.000557645}
```

In[36]:= **enexy1 = energiasxy[[4]] * elv**
enexy2 = energiasxy[[3]] * elv
enexy3 = energiasxy[[2]] * elv
enexy4 = energiasxy[[1]] * elv

Out[36]= 0.0151739

Out[37]= 0.115654

Out[38]= 0.119559

Out[39]= 0.22004

(* Now we calculate the energy z and add t to the energy in xy for the electron *)

In[40]:= **lz = 85;**
mn = 0.112;
enez1 = $\dfrac{\text{hb}^2 * \pi^2}{2 * \text{mn} * \text{lz}^\wedge 2}$ * elv;

In[44]:= **enexyz1 = enexy1 + enez1**

Out[44]= 0.181115

In[46]:= **enexyz2 = enexy2 + enez1**

Out[46]= 0.281595

In[48]:= **enexyz3 = enexy3 + enez1**

Out[48]= 0.2855

```
In[50]:=  enexyz4 = enexy4 + enez1

Out[50]=  0.385981
```

(* we define the Hamiltonian matrix for the hole *)

```
In[51]:=  mathuecoH = Table[0, {fil, 1, fin}, {col, 1, fin}];
          fil = 1;
          For[k = 1, k ≤ dif, k = k + 1,
            For[fil = fil, fil ≤ k * dif, fil = fil + 1, mathuecoH[[fil, fil]] = k * v]];
          For[fil = 1, fil ≤ fin - 1, fil = fil + 1, mathuecoH[[fil, fil + 1]] = oo];
          For[fil = dif, fil ≤ fin - 1, fil = fil + dif, mathuecoH[[fil, fil + 1]] = 0];
          For[fil = 2, fil ≤ fin, fil = fil + 1, mathuecoH[[fil, fil - 1]] = oo];
          For[fil = np - 1, fil ≤ fin, fil = fil + dif, mathuecoH[[fil, fil - 1]] = 0];
          For[fil = 1, fil ≤ fin - dif, fil = fil + 1, mathuecoH[[fil, fil + dif]] = o];
          For[fil = np - 1, fil ≤ fin, fil = fil + 1, mathuecoH[[fil, fil - dif]] = o]
          mathuecoH // MatrixForm;
```

```
In[63]:=  matdepH = Table[If[fil == col, p, 0], {fil, 1, fin}, {col, 1, fin}];
          matdepH // MatrixForm;
```

```
In[65]:=  matrizH2 = mathuecoH + matdepH;
          matrizH2 // MatrixForm;
```

```
In[67]:=  mhz = 1;
          mh = 0.45;
          qh = 1;
```

```
In[70]:=  px = hb² / (mh * hx²);

          py = hb² / (mh * hy²);

          p = bx + by;

          o = -hb² / (2 * mh * hx²);

          oo = -hb² / (2 * mh * hy²);

          v = qh * F * hx;
```

(* Now we calculate the energy in xy for the hole *)

```
In[76]:=  energiashuecoxy = Eigenvalues[matrizH2, -4] // N

Out[76]=  {0.905506, 0.904587, 0.904147, 0.903228}
```

```
In[81]:=  enhxy1 = energiashuecoxy[[4]] * elv;
          enhxy2 = energiashuecoxy[[3]] * elv;
          enhxy3 = energiashuecoxy[[2]] * elv;
          enhxy4 = energiashuecoxy[[1]] * elv;
```

(* Now we calculate the energy z and add t to the energy in xy for the hole *)

```
In[86]:=  enhz1 = hb² * π² / (2 * mhz * lz ^ 2) * elv;
```

```
In[91]:=  (* The total energy *)

          enhxyz1 = enhxy1 + enhz1
          enhxyz2 = enhxy2 + enhz1
          enhxyz3 = enhxy3 + enhz1
          enhxyz4 = enhxy4 + enhz1
```

Out[91]= 24.5961

Out[92]= 24.6211

Out[93]= 24.633

Out[94]= 24.658

```
In[95]:=  energia1 = enexyz1 + enhxyz1
```

Out[95]= 24.7772

```
In[96]:=  energia2 = enexyz2 + enhxyz2
```

Out[96]= 24.9027

```
In[97]:=  energia3 = enexyz3 + enhxyz3
```

Out[97]= 24.9185

```
In[98]:=  energia4 = enexyz4 + enhxyz4
```

Out[98]= 25.044

```
In[99]:=
          (* WAVEFUNCTIONS *)
          (* For holes *)

          holefunc = Eigenvectors[matrizH2, -4];
```

```
In[100]:=
          funchole1 = holefunc[[4]];
          funchole2 = holefunc[[3]];
          funchole3 = holefunc[[2]];
          funchole4 = holefunc[[1]];

          (* we normalize the functions *)

          gij = Dot[funchole1, funchole1]
```

Out[104]=
          1.

```
In[105]:=
          int = hx * hy * gij
```

Out[105]=
          14.8832
```

*In[107]:=*
```
norm = 1 / √int
```

*Out[107]=*
```
0.25921
```

*In[108]:=*
```
funchole1norm = norm * funchole1;

(* we can prove it *)
```

*In[109]:=*
```
hx * hy * Dot[funchole1norm, funchole1norm]
```

*Out[109]=*
```
1.
```

```
(* For electrons*)
```

In[110]:=
```
elecfunc = Eigenvectors[matrizH, -4];
```

In[111]:=
```
funcelec1 = elecfunc[[4]];
funcelec2 = elecfunc[[3]];
funcelec3 = elecfunc[[2]];
funcelec4 = elecfunc[[1]];
```

In[115]:=
```
(* now we normalize the functions for electrons *)

jij = Dot[funcelec1, funcelec1]
```

Out[115]=
```
1.
```

In[116]:=
```
integral = hx * hy * jij
```

Out[116]=
```
14.8832
```

In[117]:=
```
cten = 1 / √integral
```

Out[117]=
```
0.25921
```

In[118]:=
```
funcelec1norm = cten * funcelec1;
```

```
In[119]:=
        (* prove it *)

        hx * hy * Dot[funcelec1norm, funcelec1norm]
Out[119]=
        1.

In[120]:=
        (* we build a list with the funtion for the ground state *)

In[121]:=
        electronlist1 = Table[{0, 0}, {i, 1, np - 2}, {j, 1, np - 2}];
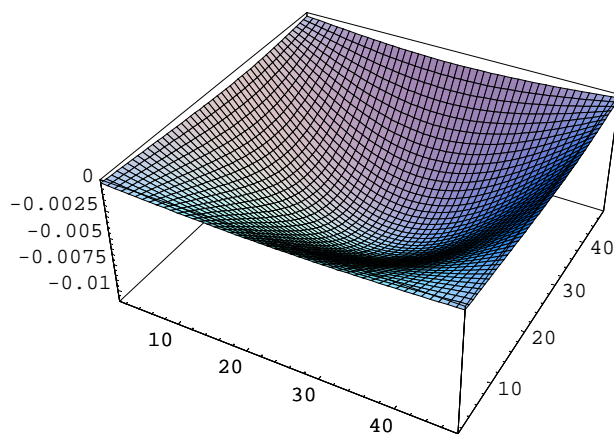        listax =
          elem = 0;
        For[i = 2, i ≤ np - 1, i = i + 1,
          For[j = 2, j ≤ np - 1, j = j + 1,
           elem = elem + 1; electronlist1[[i - 1, j - 1]] = funcelec1norm[[elem]]]];

In[124]:=
        ListPlot3D[electronlist1]
```



```
Out[124]=
        - SurfaceGraphics -

In[125]:=
        holelist1 = Table[{0, 0}, {i, 1, np - 2}, {j, 1, np - 2}];
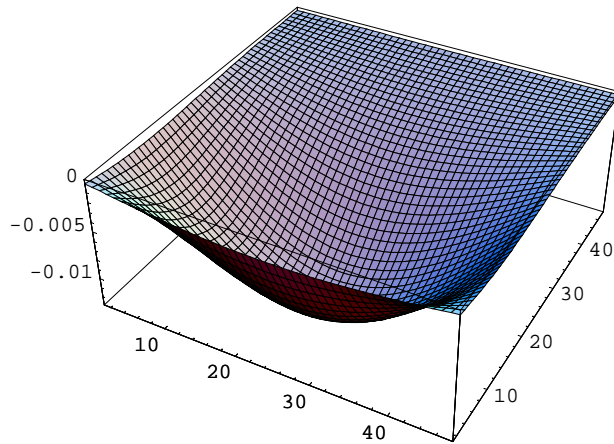        listax =
          elem = 0;
        For[i = 2, i ≤ np - 1, i = i + 1,
          For[j = 2, j ≤ np - 1, j = j + 1,
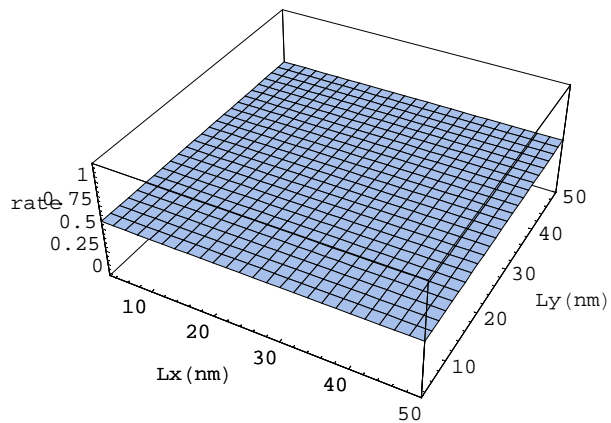           elem = elem + 1; holelist1[[i - 1, j - 1]] = funchole1norm[[elem]]]];
```

In[128]:=

ListPlot3D[holelist1]

Out[128]=

  ▪ SurfaceGraphics ▪

In[129]:=

(∗ Now we calculate the rate of the transition probability ∗)

$$fz[z\_] := \sqrt{\frac{2}{lz}} * Sin\left[\frac{\pi * z}{lz}\right];$$

rate[lx_, ly_] :=
 ((hx ∗ hy ∗ Dot[funcelec1norm, funchole1norm]) ∗ Integrate[fz[z] ∗ fz[z], {z, 0, lz}]) ^ 2

In[131]:=

Plot3D[rate[lx, ly], {lx, 3, 50}, {ly, 3, 50}, AxesLabel → {"Lx(nm)", "Ly(nm)", "rate"}]

Out[131]=

  ▪ SurfaceGraphics ▪