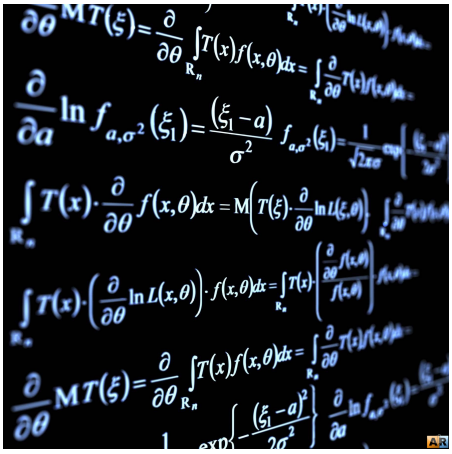

Project 3

FYS3150 - Computational physics

Quantum dots

Author:
Vidar Skogvoll

October 25, 2014

Abstract

Here is a short summary of the project.

Contents

1 Introduction

Quantum mechanics is an exciting field.

2 Theory

Here is all the theory needed to understand the project.

2.1 The physical system

This is the section explaining the physics of the system. Throughout the project, *natural units* are used ($\hbar = 1$,$c = 1$,$e = 1$,$m_e = 1$) and all energies are in so-called *atomix units* a.u.

2.1.1 The quantum mechanics and the variational principle

The quantum mechanics

In this project we will look at a system of N electrons in a so-called *quantum dot*. That is, a two dimensional harmonic oscillator with potential

$$V(\vec{r}) = \frac{1}{2}\omega^2 r^2 \tag{2.1.1}$$

This potential gives rise to a multi-particle Hamiltonian $\hat{H}$ given as the sum of an ordinary Hamiltonian and an electron repulsive part

$$\hat{H} = \sum_{i=1}^{N}\left(-\frac{1}{2}\nabla_i^2 + \frac{1}{2}\omega^2 r_i^2\right) + \sum_{i<j}\frac{1}{r_{ij}} \tag{2.1.2}$$

Where $r_{ij} = |\vec{r}_i - \vec{r}_j|$ is the distance between the electrons i and j and $r_i = |\vec{r}_i| = \sqrt{x_i^2 + y_i^2}$ when $\vec{r}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$. Our goal in this project is to find the ground eigenstate and energy of this multi-particle Hamiltonian numerically.

The variational principle

We will approach this by constructing a real test function $\Psi_T(\vec{r}_0, \vec{r}_1, ..., \vec{r}_{N-1}, \alpha, \beta)$ dependent on two parameters α and β and calculate the expextation value of the hamilton operator $\langle\hat{H}\rangle$. As we know, the eigenstates Ψ_i of the Hamiltonian forms a complete basis, so any state, including our test state Ψ_T, can be written as a linear combination of the eigenstates

$$\Psi_T = \sum_i c_i \Psi_i \tag{2.1.3}$$

Inserting this expression into the equation for the expectation value of $\hat{H}$ (remembering that Ψ_T is real) gives

$$\langle \hat{H} \rangle = \frac{\int \Psi_T \hat{H} \Psi_T d\vec{r}}{\int \Psi_T \Psi_T d\vec{r}} = \frac{\int \left(\sum_i c_i \Psi_i\right) \hat{H} \left(\sum_i c_i \Psi_i\right) d\vec{r}}{\int \left(\sum_i c_i \Psi_i\right) \left(\sum_i c_i \Psi_i\right) d\vec{r}} = \frac{\int \left(\sum_i c_i \Psi_i\right) \left(\sum_i c_i E_i \Psi_i\right) d\vec{r}}{\int \left(\sum_i c_i \Psi_i\right) \left(\sum_i c_i \Psi_i\right) d\vec{r}} \tag{2.1.4}$$

The energy of the ground state E_0 is smaller than all other E_i's so

$$\begin{aligned} \langle \hat{H} \rangle &= \frac{\int \left(\sum_i c_i \Psi_i\right) \left(\sum_i c_i E_i \Psi_i\right) d\vec{r}}{\int \left(\sum_i c_i \Psi_i\right) \left(\sum_i c_i \Psi_i\right) d\vec{r}} \\ &\geq \frac{\int \left(\sum_i c_i \Psi_i\right) \left(\sum_i c_i E_0 \Psi_i\right) d\vec{r}}{\int \left(\sum_i c_i \Psi_i\right) \left(\sum_i c_i \Psi_i\right) d\vec{r}} = E_0 \frac{\int \left(\sum_i c_i \Psi_i\right) \left(\sum_i c_i \Psi_i\right) d\vec{r}}{\int \left(\sum_i c_i \Psi_i\right) \left(\sum_i c_i \Psi_i\right) d\vec{r}} = E_0 \end{aligned} \tag{2.1.5}$$

$$\langle H \rangle \geq E_0 \tag{2.1.6}$$

This simple observation is called *the variational principle* and is what we will use to narrow our search for the optimal parameters α and β. We will look for the parameters α and β that gives us the smallest value of $\langle \hat{H} \rangle$ and this will be our estimate for the ground state.

Finding the expectation value of $\hat{H}$

We have

$$\langle \hat{H} \rangle = \frac{\int \Psi_T \hat{H} \Psi_T d\vec{r}}{\int \Psi_T \Psi_T d\vec{r}} = \int \frac{\Psi_T \Psi_T}{\int \Psi_T \Psi_T d\vec{r}} \frac{1}{\Psi_T} \hat{H} \Psi_T d\vec{r} \tag{2.1.7}$$

If we rename probability density function of the particles $\frac{\Psi_T \Psi_T}{\int \Psi_T \Psi_T d\vec{r}} = P(\vec{r})$ and $E_L(\vec{r}) = \frac{1}{\Psi_T} \hat{H} \Psi_T$, then the integral becomes

$$\langle \hat{H} \rangle = \int P(\vec{r}) E_L(\vec{r}) d\vec{r} = \langle E_L \rangle \tag{2.1.8}$$

But we could very well find a minimum of $\langle \hat{H} \rangle$ that is not an eigen energy of the system, i.e. still larger than E_0. To address this problem, let's look at the variance V_{E_L} of $\langle E_L \rangle$.

$$V_{E_L} = \langle E_L^2 \rangle - \langle E_L \rangle^2 = \int P(\vec{r}) \left(\frac{1}{\Psi_T} \hat{H} \Psi_T \right)^2 d\vec{r} - \left(\int P(\vec{r}) \frac{1}{\Psi_T} \hat{H} \Psi_T d\vec{r} \right)^2 \tag{2.1.9}$$

If the state Ψ_T is an eigenstate of $\hat{H}$ with eigenvalue E then

$$\begin{aligned} V_{E_L} &= \int P(\vec{r}) \left(\frac{1}{\Psi_T} E \Psi_T \right)^2 d\vec{r} - \left(\int P(\vec{r}) \frac{1}{\Psi_T} E \Psi_T d\vec{r} \right)^2 \\ &= E^2 \left(\int P(\vec{r}) \left(\frac{1}{\Psi_T} \Psi_T \right)^2 d\vec{r} - \left(\int P(\vec{r}) \frac{1}{\Psi_T} \Psi_T d\vec{r} \right)^2 \right) = E^2 \left(\int P(\vec{r}) (1)^2 d\vec{r} - \left(\int P(\vec{r}) \cdot 1 d\vec{r} \right)^2 \right) \\ &= E^2 \left(\int P(\vec{r}) d\vec{r} - \left(\int P(\vec{r}) d\vec{r} \right)^2 \right) = E^2 (1 - 1^2) = 0 \end{aligned} \tag{2.1.10}$$

So if Ψ_T is an eigenstate of $\hat{H}$ then the variance V_{E_L} of E_L is 0

$$V_{E_L} = \langle E_L^2 \rangle - \langle E_L \rangle^2 = 0 \tag{2.1.11}$$

This will serve as a verification that the state we have found when minimizing the expectation value of E_L is indeed an eigenstate of the hamilton operator.

2.1.2 The test function Ψ_T

We will in this project use the trial wavefunctions of $\vec{r}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$ given by

$$\Psi_T(\vec{r}_0, ..., \vec{r}_{N-1}) = \mathrm{Det}_M(\phi_0, ..., \phi_{N-1}) \cdot \mathrm{J}(\vec{r}_0, ..., \vec{r}_{N-1}) \tag{2.1.12}$$

The modified Slater determinant $\mathbf{Det}_M(\phi_0, ..., \phi_{N-1})$

$\mathrm{Det}_M(\phi_0, ..., \phi_{N-1})$ is a modified *Slater determinant* defined as

$$\mathrm{Det}_M(\phi_0, ..., \phi_{N-1}) = \begin{vmatrix} \phi_0(\vec{r}_0) & \phi_2(\vec{r}_0) & ... & \phi_{N-2}(\vec{r}_0) \\ \phi_0(\vec{r}_2) & \phi_2(\vec{r}_2) & ... & \phi_{N-2}(\vec{r}_2) \\ ... & ... & ... & ... \\ \phi_0(\vec{r}_{N-2}) & \phi_2(\vec{r}_{N-2}) & ... & \phi_{N-2}(\vec{r}_{N-2}) \end{vmatrix} \cdot \begin{vmatrix} \phi_1(\vec{r}_1) & \phi_3(\vec{r}_1) & ... & \phi_{N-1}(\vec{r}_1) \\ \phi_1(\vec{r}_3) & \phi_3(\vec{r}_3) & ... & \phi_{N-1}(\vec{r}_3) \\ ... & ... & ... & ... \\ \phi_1(\vec{r}_{N-1}) & \phi_3(\vec{r}_{N-1}) & ... & \phi_{N-1}(\vec{r}_{N-1}) \end{vmatrix} \tag{2.1.13}$$

Where $\phi_i(\vec{r}_i)$ is a wavefunction resembling one of the eigenfunctions of the Hamilton operator for *one* particle in a two dimensional harmonic oscillator, but parameterized by α in the following way:

$$\phi_i(\vec{r}_j) = H_{n_x}(\sqrt{\alpha\omega}x_j) H_{n_y}(\sqrt{\alpha\omega}y_j) \exp(-\alpha\omega(x^2 + y^2)/2) \tag{2.1.14}$$

The reason for this modified version of the Slater determinant (whose real form can be explored elsewhere[1]) is that the spin parts of the wavefunctions are not incorporated in the expressions of $\phi_i(\vec{r}_j)$. The result is that if we were to insert these into a regular Slater determinant we would get 0, everytime. The modified Slater determinant Det_M avoids this issue while still conserving some of the most important properties of the Slater determinant.

$n_x(i)$ and $n_y(i)$ corresponds to the quantum numbers needed to "fill up" the system from the lowest energy levels twice (one for each spin configuration). For $i < 12$, the explicit depence of n_x, n_y on i is given in table 2.1.1.

$i =$	0	1	2	3	4	5	6	7	8	9	10	11
$n_x =$	0	0	1	1	0	0	2	2	1	1	0	0
$n_y =$	0	0	0	0	1	1	0	0	1	1	2	2

Table 2.1.1: The explicit dependence of n_x and n_y on i in the construction of the trial wavefunctions.

[1] http://en.wikipedia.org/wiki/Slater_determinant

The Jastrow factor $\mathbf{J}(\vec{r}_0, ..., \vec{r}_{N-1})$

$\mathrm{J}(\vec{r}_0, ..., \vec{r}_{N-1})$ is a so-called *Jastrow factor*, which represents the electron repulsion part of the wavefunction, defined as

$$\mathrm{J}(\vec{r}_0, ...\vec{r}_{N-1}) = \prod_{i<j}^{N} \exp\left(\frac{a_{ij} r_{ij}}{1 + \beta r_{ij}}\right) \quad \text{where}$$

$$r_{ij} = |\vec{r}_i - \vec{r}_j| \quad \text{and} \quad a_{ij} = \begin{cases} 1/3 & \text{if spin(i) and spin(j) are parallell} \\ 1 & \text{if spin(i) and spin(j) are anti-parallell} \end{cases} \tag{2.1.15}$$

From the expression of the Jastrow factor, we see that it is zero whenever $r_{ij} = 0$ for any pair $i \neq j$. This is what we want from such a factor, namely that the wavefunction is zero whenever there is no distance between two particles (i.e. $r_{ij} = 0$).

Using the hamiltonian

It can be shown [2] that.

2.1.3 Discretization

2.1.4 The virial theorem

2.2 The numerical foundation

This is the section explaining the numerical theory upon which the project is built.

2.2.1 Monte Carlo simulations

A Monte Carlo simulation is a way of solving a mathematical or physical problem by generating a random (or pseudorandom [2]) sequence of numbers and evaluating some quantity on the assumption that our the random sequence of numbers is representative of the domain from which the quantity is evaluated. An example is evaluating the area of the unit circle by randomly placing points in a $[-1, 1] \times [-1, 1]$ grid and find the fraction points whose distance to the origin is ≤ 1 and multiply this fraction by the area of the grid (i.e. 4). Such a simple Monte Carlo simulation can give the result as shown in figure 2.2.1.

However, the method is not confined to this sort of problem, but can be applied to a variety of mathematical and physical problems. In this report, the method, through the Metropolis algorithm (see section 2.2.3) has been applied to a quantum mechanical system.

2.2.2 Importance sampling

Sometimes, functions are more important in certain domains. When we use a Monte Carlo approach to evaluate a quantity on such a function we can improve the method significantly by ensuring that the probability distribution from which we choose our random values reflect the

[2] No electronic random number generator of today is truly random. The sequence of numbers generated will repeat itself after a long period. These periods however, are increadibly long and we will for this report consider the random number generators to be truly random.

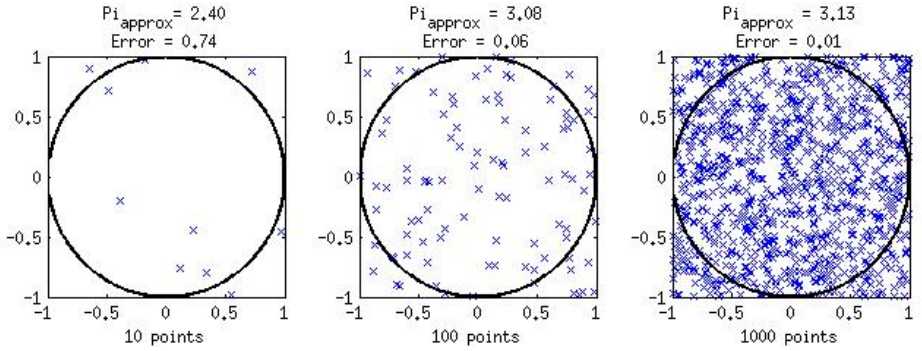

Figure 2.2.1: The results from a very simple Monte Carlo simulation of estimating the circle constant π. The precision increases with the number of points.

parts where the function is important. We do of course have to remember that our function lives on its whole domain, so to not get a biased result, we need to weigh our result with respect to the probability function we have chosen. This is called importance sampling. **INSERT EXAMPLE HERE**

2.2.3 The Metropolis algorithm

The Metropolis algorithm is a method which cleverly employs a stoichastic approach in order to quickly estimate certain mathematical objects. The method is explained at lengths elsewhere[1], but in this section we will look at an example which captures the main idea of the method.

Suppose we have a PDF[3] $P(x)$ in a domain $[a, b]$ for which we want to calculate the expectation value $\langle g \rangle$ of some function $g(x)$. The integral we need to solve is then

$$\langle g \rangle = \int_a^b P(x)g(x)dx \tag{2.2.1}$$

This integral can be approximated as follows

$$\int_a^b P(x)g(x)dx \approx \frac{b-a}{N} \sum_i P(x_i)g(x_i) \equiv I \tag{2.2.2}$$

Where x_i are some uniformly chosen values in the interval $[a, b]$. Now, imagine instead of picking values x_i uniformly and weighing them by multiplying $g(x)$ with $P(x)$ instead chose the values of $\tilde{x_i}$ from the PDF $P(x)$ and calculated the quantity $\tilde{I}$ given by

$$\tilde{I} = \frac{1}{N} \sum_i g(\tilde{x_i}) \tag{2.2.3}$$

It can be shown mathematically that for large enough N, these two quantities I and $\tilde{I}$ approach the same value. The problem with such an approach is that we need the precise expression for the PDF $P(x)$ and a robust algorithm for choosing random values from it. With the Metropolis

[3]Probability Distribution Function

algorithm however, we can use this approach *without* knowing the precise expression of the PDF and the relevant values from the domain come naturally.

The algorithm requires that we are able to calculate $\tilde{P}(x)$, an unnormalized version of $P(x)$ (i.e. some function $aP(x)$ proportional to $P(x)$). This may seem like a very strong requirement, but in many applications, as in this project, this is a much easier task than to calculate the precise PDF. The algorithm goes as follows. Starting with a position x choose a new trial position

$$x_p = x + r\Delta x \tag{2.2.4}$$

Where Δx is a predefined step length and r is a random number between zero and one. Then generate a probability criteria s, a random number between zero and one. If

$$\frac{P(x_p)}{P(x)} = \frac{aP(x_p)}{aP(x)} = \frac{\tilde{P}(x_p)}{\tilde{P}(x)} \equiv w \geq s \tag{2.2.5}$$

We accept the trial position as our new x and if not we reject it. If we choose new values of x_i in this manner, the collection of x_i's will in fact reflect the PDF $P(x)$, which was what we needed in order to use equation 2.2.3. Note how equation 2.2.5 doesn't require us to have the exact form of the probability distribution function, only a function $\tilde{P}(x)$ proportional to it.

The intuition behind the algorithm is that for each new position x_i we generate is drawn towards the part of the domain where $P(x)$ is bigger. To see this, we note that if $P(x_p) > P(x)$ then $\frac{P(x_p)}{P(x)} > 1$ which is always bigger than $s \in [0, 1]$ and the new move is always accepted. Whereas if $P(x_p) < P(x)$, the move might be rejected. This allows new values of x_i to be chosen from where $P(x)$ is big, but at the same time allows values with lower values of $P(x)$ to be chosen. Which is what we expect from a PDF. The fact that for a large number M of such steps, the values x_i picked actually reflects the PDF requires some more mathematics, and once again we refer to the lecture notes of the course [1].

As discussed in section 2.1.1 we need to solve the integral

$$\langle E_L \rangle = \int P(\vec{r}) E_L(\vec{r}) d\vec{r} \tag{2.2.6}$$

Where we have a trial function

$$\Psi_T(\vec{r}_0, ..., \vec{r}_{N-1}, \alpha, \beta) \tag{2.2.7}$$

dependent on 2 trial parameters α and β where $\vec{r}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$. This is exactly the kind of problem the Metropolis algorithm can solve and the explicit algorithm for calculating $\langle E_L \rangle$ and $\langle E_L^2 \rangle$ is given in algorithm 1.

3 Results and discussion

Section listing results and discussing them.

Data:
A number M of Monte Carlo simulations to be performed
A predefined steplength Δr
An initial position matrix $\mathbf{r} = \begin{pmatrix} x_0 & x_2 & ... & x_{N-1} \\ y_0 & y_2 & ... & y_{N-1} \end{pmatrix}$
Result:
The expectation value of the local energy: $\langle E_L \rangle$
The expectation value of the local energy squared: $\langle E_L^2 \rangle$

begin
  $cumulative\_local\_energy = 0$ // Initialization
  $cumulative\_local\_energy\_squared = 0$
  $counter = 0$
  while $counter < M$ **do**
    $i = \text{randint}(0, 1, ..., N-1)$ // Choose random element index
    $\Delta\vec{r} = \Delta r \cdot \begin{pmatrix} \text{rand}(0,1) \\ \text{rand}(0,1) \end{pmatrix}$ // Create a random two-dimensional step
    $\mathbf{r_p} = \begin{pmatrix} x_0 & x_1 & ... & x_i & ... & x_{N-1} \\ y_0 & y_1 & ... & y_i & ... & y_{N-1} \end{pmatrix} + \Delta\vec{r}$ // Create a trial position matrix
    $s = \text{randint}(0,1)$ // Generate a probability criteria
    $w = |\psi(\alpha, \beta, \mathbf{r_p})|^2 / |\psi(\alpha, \beta, \mathbf{r})|^2$ // Calulate the probability ratio
    if $w \geq s$ **then**
      $\vec{r} = \vec{r}_p$
      $E_L(\mathbf{r}, \alpha, \beta) = \frac{1}{\Psi_T(\mathbf{r}, \alpha, \beta)} \hat{H} \psi_T(\mathbf{r}, \alpha, \beta)$ // Calculate the local energy
      $cumulative\_local\_energy \overset{+}{=} E_L(\mathbf{r}, \alpha, \beta)$ // Update *cumulative_local_energy*
      $cumulative\_local\_energy\_squared \overset{+}{=} E_L(\mathbf{r}, \alpha, \beta)^2$ // Update *cumulative_local_energy_squared*
      $counter \overset{+}{=} 1$ // Update *counter*
    end
  end
  Calculate $\langle E_L \rangle = \frac{cumulative\_local\_energy}{M}$
  Calculate $\langle E_L^2 \rangle = \frac{cumulative\_local\_energy\_squared}{M}$
end

Algorithm 1: The metropolis algorithm used for finding the expecation value of the local energy and the expecation value of the local energy squared.

3.1 Subsection

The same as the method section.

4 Conclusion

References

[1] Morten Hjorth-Jensen. *Computational Physics - Lecture Notes Fall 2014*. August 2014.

[2] Jørgen Høgberget. Quantum monte-carlo studies of generalized many-body systems, June 2013.