

# Project 2 in FYS4411: Computational Physics 2

Mathias M. Vege

May 29, 2017

## Abstract

NAN

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Theory</b>	<b>2</b>
2.1	Variational Monte Carlo . . . . .	3
2.2	The Metropolis Algorithm . . . . .	4
2.2.1	Uniform sampling . . . . .	6
2.2.2	Importance sampling . . . . .	6
2.2.3	Steepest descent . . . . .	9
2.3	Electron wave function . . . . .	10
2.3.1	Slater determinants . . . . .	10
2.3.2	The Jastrow factor . . . . .	11
2.4	Two electron case . . . . .	11
2.4.1	Unperturbed local energy . . . . .	11
2.4.2	Local energy . . . . .	12
2.4.3	Quantum force . . . . .	12
2.5	$N$ electron case . . . . .	12
2.5.1	Unperturbed local energy . . . . .	12
2.5.2	Local energy . . . . .	12
2.5.3	Quantum force . . . . .	12
<b>3</b>	<b>Implementation</b>	<b>12</b>
3.0.1	Slater determinants . . . . .	12
<b>4</b>	<b>Results</b>	<b>12</b>

<b>5</b>	<b>Discussion and conclusion</b>	<b>12</b>
<b>6</b>	<b>Appendix</b>	<b>12</b>
6.1	Hermite polynomials . . . . .	12

## 1 Introduction

The goal of this project is to study closed shell systems of electrons confined in a harmonic oscillator potential - a quantum dot. Within this scope we are investigating the ground state energies, **exception values kinetic and potential energies, single particle densities and one-body densities** . The system we are interested in is a two dimensional system of  $N$  electrons, and since we have closed shell systems we will look at  $N = 2, 6$  and 12 electrons.

## 2 Theory

As tradition demands we begin by looking at the Hamiltonian of the system we are to solve,

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

In order to accommodate a modern notational benefits and simplifications, we use natural units( $\hbar = c = e = m_e = 1$ ). We can also observe that  $N$  is the number of particles we are using, and the  $\omega$  is the oscillator frequency for the harmonic oscillator part. The first part, we recognize as the unperturbed part of the Hamiltonian,

$$\hat{H}_0 = \sum_{i=1}^N \left( -\frac{1}{2} \nabla_i^2 + \frac{1}{2} \omega^2 r_i^2 \right), \quad (2)$$

and the last part is the perturbation to our system,

$$\hat{H}_1 = \sum_{i<j}^N \frac{1}{r_{ij}} \quad (3)$$

such that  $\hat{H} = \hat{H}_0 + \hat{H}_1$ . The distance between two electrons is defined as following,

$$r_{ij} = |\mathbf{r}_i - \mathbf{r}_j| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (4)$$

## 2.1 Variational Monte Carlo

In order to go through the basic steps of variational Monte Carlo, we begin by deriving the basic formulas needed. Let  $\psi_T$  be our trial wave function, which we can expand into an energy basis which we assume is normalized,

$$\psi_T = \sum_{i=0} c_i |\psi_i\rangle$$

The expectation value of the energy given by the Hamiltonian  $\hat{H}$  (from now on omitting the hat) is,

$$\begin{aligned} \langle E \rangle &= \langle \psi_T | H | \psi_T \rangle \\ &= \sum_{i=0} \sum_{j=0} c_i^* c_j \langle \psi_i | H | \psi_j \rangle \\ &= \sum_{i=0} \sum_{j=0} c_i^* c_j E_j \langle \psi_i | \psi_j \rangle \\ &= \sum_{i=0} |c_i|^2 E_i \end{aligned}$$

From the variational principle in quantum mechanics, we have that the energy of the ground state can be approximated by a trial wave function,

$$E_0 \geq E[\psi_T] = \frac{\langle \psi_T | H | \psi_T \rangle}{\langle \psi_T | \psi_T \rangle} \quad (5)$$

Writing this on integral form, we get

$$\begin{aligned} E[H] &= \langle H \rangle = \frac{\int d\mathbf{r} \psi_T^*(\mathbf{r}) H(\mathbf{r}) \psi_T(\mathbf{r})}{\int d\mathbf{r} \psi_T^*(\mathbf{r}) \psi_T(\mathbf{r})} \\ &= \frac{\sum_{ij} c_i^* c_j \int d\mathbf{r} \psi_i^*(\mathbf{r}) H(\mathbf{r}) \psi_j(\mathbf{r})}{\sum_{ij} c_i^* c_j \int d\mathbf{r} \psi_i^*(\mathbf{r}) \psi_j(\mathbf{r})} \\ &= \frac{\sum_i |c_i|^2 E_i}{\sum_i |c_i|^2} \\ &\geq E_0 \end{aligned}$$

In order to make any progress with variational Monte Carlo, we need to get ourselves a wave function. This wave function will have to take  $N$  particles  $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$ , and have can be varied by a set of variational parameters  $\{\alpha_i\}$ . The integral we wish to solve is now,

$$E[H] = \frac{\int d\mathbf{r} \psi_T^*(\mathbf{r}, \alpha) H(\mathbf{r}) \psi_T(\mathbf{r}, \alpha)}{\int d\mathbf{r} \psi_T^*(\mathbf{r}, \alpha) \psi_T(\mathbf{r}, \alpha)} \quad (6)$$

As with Monte Carlo methods, we need to define ourselves a probability distribution function(PDF) which will be based off our trial wave function  $\psi_T(\mathbf{r})$ ,

$$P(\mathbf{r}) = \frac{|\psi_T(\mathbf{r})|^2}{\int |\psi_T(\mathbf{r})|^2 d\mathbf{r}} \quad (7)$$

We can use this to rewrite our  $E[H]$  which now also depends on  $\alpha$ ,

$$E[H(\alpha)] = \int P(\mathbf{r}) E_L(\mathbf{r}) d\mathbf{r} \approx \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} P(\mathbf{r}_i, \alpha) E_L(\mathbf{r}_i, \alpha)$$

where  $N_{MC}$  is the number of Monte Carlo cycles which we run for. We now have an estimate for the ground state energy from the average of our Monte Carlo samples. The algorithm for the VMC(Variational Monte Carlo) method will be based of the Metropolis algorithm **CITE METROPOLIS!!** , such that a quick review of this is in order.

## 2.2 The Metropolis Algorithm

The Metropolis algorithm **SITER** is a method for obtaining a sequence of samples when direct sampling is problematic. Given a PDF  $P_i^{(n)}$ , where  $n$  is the time step, and  $i$  is the current state, the algorithm is such that a transitioning probability to a new state  $j$  is given by  $T_{i \rightarrow j}$ . Then, the probability of accepting this state is given by  $A_{i \rightarrow j}$ . If rejected, no move is performed and we remain at state  $i$ . We will require that  $A$  and  $T$  is such that  $P_i^{(n \rightarrow \infty)} \rightarrow p_i$ . This gives rise to the relation

$$\begin{aligned} P_i^{(n)} &= \sum_j \left[ P_j^{(n-1)} T_{j \rightarrow i} A_{j \rightarrow i} + P_i^{(n-1)} T_{i \rightarrow j} (1 - A_{i \rightarrow j}) \right] \\ &= \sum_j \left[ P_j^{(n-1)} T_{j \rightarrow i} A_{j \rightarrow i} - P_i^{(n-1)} T_{i \rightarrow j} A_{i \rightarrow j} + P_i^{(n-1)} T_{i \rightarrow j} \right] \\ &= \sum_j \left[ P_j^{(n-1)} T_{j \rightarrow i} A_{j \rightarrow i} - P_i^{(n-1)} T_{i \rightarrow j} A_{i \rightarrow j} \right] + P_i^{(n-1)} \sum_j T_{i \rightarrow j} \\ &= \sum_j \left[ P_j^{(n-1)} T_{j \rightarrow i} A_{j \rightarrow i} - P_i^{(n-1)} T_{i \rightarrow j} A_{i \rightarrow j} \right] + P_i^{(n-1)} \cdot 1 \end{aligned}$$

Since we require our system to make some transition into a new state  $j$ ,  $\sum_j T_{i \rightarrow j} = 1$ . Now using the requirement  $P_i^{(n \rightarrow \infty)} \rightarrow p_i$ , we get

$$\begin{aligned} p_i &= \sum_j [p_j T_{j \rightarrow i} A_{j \rightarrow i} - p_i T_{i \rightarrow j} A_{i \rightarrow j}] + p_i \\ &\downarrow \\ 0 &= \sum_j [p_j T_{j \rightarrow i} A_{j \rightarrow i} - p_i T_{i \rightarrow j} A_{i \rightarrow j}] \end{aligned}$$

The requirement applies for the system as a whole when time tends to infinity. This is a rather weak requirement, so we seek to enforce a similar condition to each state. Namely,

$$p_j T_{j \rightarrow i} A_{j \rightarrow i} = p_i T_{i \rightarrow j} A_{i \rightarrow j}$$

which we observe implies that transitioning and accepting a state  $i \rightarrow j$  is equal to that of  $j \rightarrow i$ . By rearranging we get,

$$\frac{A_{j \rightarrow i}}{A_{i \rightarrow j}} = \frac{p_j T_{j \rightarrow i}}{p_i T_{i \rightarrow j}} \quad (8)$$

This is a much stronger requirement, and is called *detailed balance*. The Metropolis algorithm is now all about maximizing  $A$ , such that

$$A_{i \rightarrow j} = \min \left( 1, \frac{p_i T_{i \rightarrow j}}{p_j T_{j \rightarrow i}} \right)$$

The ratio which we use to accept or reject a step is defined as

$$q(\mathbf{r}^{\text{new}}, \mathbf{r}^{\text{old}}) = \frac{p_i T_{i \rightarrow j}}{p_j T_{j \rightarrow i}} \quad (9)$$

Let us then recapitulate what the Metropolis algorithm becomes as applied to our VMC problem.

---

**Algorithm 1** Metropolis algorithm for Variational Monte Carlo.

---

- 1: Set up initial conditions for our system.
  - 2: Sample a new state  $j$  with transition probability  $T_{i \rightarrow j}$  by the desired sampling method.
  - 3: Accept new state  $j$  with acceptance probability  $A_{i \rightarrow j}$ .
  - 4: If rejected, we go back to state  $i$ .
  - 5: Sample the system.
  - 6: Repeat step 2-5  $N_{MC}$  times.
-

### 2.2.1 Uniform sampling

For sampling a new state (by which we mean a new position), we use

$$\mathbf{r}^{\text{new}} = \mathbf{r}^{\text{old}} + \xi \Delta \mathbf{r}^{\text{old}} \quad (10)$$

where  $\xi$  is a random number chosen from a uniform distribution, and  $\Delta \mathbf{r}$  is the step length which we update for. For this kind of uniform sampling, we seek to tune  $\Delta \mathbf{r}$  to such that we accept roughly 50% all suggested moves.

The acceptance ratio (9) as applied to our VMC calculation, is (dubbing it as  $R$  instead of  $q(\mathbf{r}^{\text{new}}, \mathbf{r}^{\text{old}})$ ),

$$R \equiv \frac{\psi_T^{\text{new}}}{\psi_T^{\text{old}}} \quad (11)$$

### 2.2.2 Importance sampling

If we wish to improve our VMC Metropolis algorithm, we can use something called importance sampling. It is based upon the Fokker-Planck equation, which characterizes a move through coordinate space. For one dimension, it reads

$$\frac{\partial P}{\partial t} = D \frac{\partial}{\partial x} \left( \frac{\partial}{\partial x} - F \right) P(x, t) \quad (12)$$

The  $D$  is referred to as the diffusion constant since it is derived from the Fokker-Planck equation, and the  $F$  is called the drift term. A new position in coordinate space is chosen from solving the Langevin equation,

$$\frac{\partial x(t)}{\partial t} = DF(x(t)) + \eta \quad (13)$$

$\eta$  is a random variable. This is solved using Euler's method. For our problem, the solution and sampling of a new position is defined by

$$\mathbf{r}^{\text{new}} = \mathbf{r}^{\text{old}} + D\mathbf{F}(\mathbf{r}^{\text{old}})\Delta t + \xi\sqrt{\Delta t} \quad (14)$$

Where  $D = 0.5$  in atomic units, and  $\mathbf{F}$  is the quantum force as defined by,

$$\mathbf{F} = \frac{2}{\psi_T} \nabla \psi_T \quad (15)$$

We typically choose  $\Delta t \in [0.001, 0.01]$ . The random variable  $\xi$  is chosen from a Gaussian distribution around 0 with a standard deviation of 1. Further, the acceptance ratio  $q(\mathbf{r}^{\text{new}}, \mathbf{r}^{\text{old}})$  now becomes,

$$q(\mathbf{r}^{\text{new}}, \mathbf{r}^{\text{old}}) = \frac{G(\mathbf{r}^{\text{old}}, \mathbf{r}^{\text{new}})|\psi_T(\mathbf{r}^{\text{new}})|^2}{G(\mathbf{r}^{\text{new}}, \mathbf{r}^{\text{old}})|\psi_T(\mathbf{r}^{\text{old}})|^2} \quad (16)$$

where the  $G(\mathbf{r}^{\text{new}}, \mathbf{r}^{\text{old}})$  is a Green's function which is a solution to the Fokker-Planck equation. This is a transition probability, and is defined by

$$G(\mathbf{r}^{\text{new}}, \mathbf{r}^{\text{old}}) = \frac{1}{(4\pi D\Delta t)^{3N/2}} \exp\left(-(\mathbf{r}^{\text{new}} - \mathbf{r}^{\text{old}} - D\delta t \mathbf{F}(\mathbf{r}^{\text{old}}))^2 / 4D\Delta t\right) \quad (17)$$

A more detailed derivation of importance sampling can be found in [SITER MORTEN KOMPENDIUM HER](#). As we can already observe, it is possible to simplify the Greens ratio (16) due to the exponential. We will begin by using that the squaring of the exponential implies a dot product for vectors. For simplicity, we dub  $\mathbf{x} = \mathbf{r}^{\text{old}}$  and  $\mathbf{y} = \mathbf{r}^{\text{new}}$ . Through a lot of patience, we

get **REMOVE THIS???**

$$\begin{aligned}
\frac{G(\mathbf{x}, \mathbf{y})}{G(\mathbf{y}, \mathbf{x})} &= \exp \left[ \frac{1}{4D\Delta t} \left\{ - (x_x^2 - x_x y_x - x_x D\Delta t F_x(\mathbf{y}) + y_x^2 - y_x x_x \right. \right. \\
&\quad + y_x D\Delta t F_x(\mathbf{y}) - x_x D\Delta t F_x(\mathbf{y}) + y_x D\Delta t F_x(\mathbf{y}) + D^2 \Delta t^2 F_x^2(\mathbf{y})) \\
&\quad + (y_x^2 - y_x x_x - y_x D\Delta t F_x(\mathbf{x}) + x_x^2 - x_x y_x \\
&\quad + x_x D\Delta t F_x(\mathbf{x}) + D^2 \Delta t^2 F_x^2(\mathbf{x}) - y_x D\Delta t F_x(\mathbf{x}) + x_x D\Delta t F_x(\mathbf{x})) \\
&\quad - (x_y^2 - x_y y_y - x_y D\Delta t F_y(\mathbf{y}) - x_y y_y + y_y^2 \\
&\quad + y_y D\Delta t F_y(\mathbf{y}) - x_y D\Delta t F_y(\mathbf{y}) + y_y D\Delta t F_y(\mathbf{y} + D^2 \Delta t^2 F_y^2(\mathbf{y}))) \\
&\quad + (y_y^2 - y_y x_y - y_y D\Delta t F_y(\mathbf{x}) + x_y^2 - x_y y_y + x_y D\Delta t F_y(\mathbf{x}) \\
&\quad \left. \left. + D^2 \Delta t^2 F_y^2(\mathbf{x}) - y_y D\Delta t F_y(\mathbf{x}) + x_y D\Delta t F_y(\mathbf{x})) \right\} \right] \\
&= \exp \left[ \frac{1}{4D\Delta t} \left\{ - y_x D\Delta t F_x(\mathbf{x}) + x_x D\Delta t F_x(\mathbf{x}) + D^2 \Delta t^2 F_x^2(\mathbf{x}) \right. \right. \\
&\quad - y_x D\Delta t F_x(\mathbf{x}) + x_x D\Delta t F_x(\mathbf{x}) + x_x D\Delta t F_x(\mathbf{y}) - y_x D\Delta t F_x(\mathbf{y}) \\
&\quad + x_x D\Delta t F_x(\mathbf{y}) - y_x D\Delta t F_x(\mathbf{y}) - D^2 \Delta t^2 F_x^2(\mathbf{y}) - y_y D\Delta t F_y(\mathbf{x}) \\
&\quad + x_y D\Delta t F_y(\mathbf{x}) + D^2 \Delta t^2 F_y^2(\mathbf{x}) - y_y D\Delta t F_y(\mathbf{x}) + x_y D\Delta t F_y(\mathbf{x}) \\
&\quad + x_y D\Delta t F_y(\mathbf{y}) - y_y D\Delta t F_y(\mathbf{y}) + x_y D\Delta t F_y(\mathbf{y}) - y_y D\Delta t F_y(\mathbf{y}) \\
&\quad \left. \left. - D^2 \Delta t^2 F_y^2(\mathbf{y}) \right\} \right] \\
&= \exp \left[ \frac{1}{4D\Delta t} \left\{ - y_x D\Delta t [F_x(\mathbf{x}) + F_x(\mathbf{x}) + F_x(\mathbf{y}) + F_x(\mathbf{y})] \right. \right. \\
&\quad - D^2 \Delta t^2 [F_x^2(\mathbf{y}) - F_x^2(\mathbf{x})] + 2x_x D\Delta t [F_x(\mathbf{x}) + F_x(\mathbf{y})] \\
&\quad - 2y_y D\Delta t [F_y(\mathbf{x}) + F_y(\mathbf{y})] + 2x_y D\Delta t [F_y(\mathbf{x}) + F_y(\mathbf{y})] \\
&\quad \left. \left. - D^2 \Delta t^2 [F_y^2(\mathbf{y}) - F_y^2(\mathbf{x})] \right\} \right] \\
&= \exp \left[ - \frac{y_x}{2} (F_x(\mathbf{x}) + F_x(\mathbf{y})) - \frac{D\Delta t}{4} (F_x^2(\mathbf{y}) - F_x^2(\mathbf{x})) \right. \\
&\quad + \frac{x_x}{2} (F_x(\mathbf{x}) + F_x(\mathbf{y})) - \frac{y_y}{2} (F_y(\mathbf{x}) + F_y(\mathbf{y})) \\
&\quad \left. - \frac{D\Delta t}{4} (F_y^2(\mathbf{y}) - F_y^2(\mathbf{x})) + \frac{x_y}{4} (F_y(\mathbf{x}) + F_y(\mathbf{y})) \right]
\end{aligned}$$



Through the magic of simplification (but mostly patience), we arrive at

$$\begin{aligned} \frac{G(\mathbf{x}, \mathbf{y})}{G(\mathbf{y}, \mathbf{x})} = \exp & \left[ \frac{1}{2} (x_x - y_x) (F_x(\mathbf{x}) + F_x(\mathbf{y})) \right. \\ & + \frac{1}{2} (x_y - y_y) (F_y(\mathbf{x}) + F_y(\mathbf{y})) \\ & \left. - \frac{D\Delta t}{4} (F_x^2(\mathbf{y}) - F_x^2(\mathbf{x}) + F_y^2(\mathbf{y}) - F_y^2(\mathbf{x})) \right] \end{aligned}$$

Summing over dimensions, and we get

$$\frac{G(\mathbf{x}, \mathbf{y})}{G(\mathbf{y}, \mathbf{x})} = \exp \left[ \sum_i^{N_{\text{dim}}} \frac{1}{2} (F_i(\mathbf{x}) + F_i(\mathbf{y})) \left( \frac{D\Delta t}{2} (F_i(\mathbf{x}) - F_i(\mathbf{y})) - y_i + x_i \right) \right] \quad (18)$$

### 2.2.3 Steepest descent

The steepest descent method is a simple iterative method for finding a local minimum in parameters. It is used when we are tasked to solve systems of linear equations on the form of

$$A\mathbf{x} = \mathbf{b}$$

where we through an iterative process seeks to minimize the following,

$$\mathbf{r} = \mathbf{b} - A\mathbf{x}$$

where the exact solution implies that  $\mathbf{r} = 0$ . The process for solving this through steepest descent can be set up as,

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \nabla f(\mathbf{x}_i) \delta t \quad (19)$$

Applied to our VMC problem, where we will have one or two variational parameters,  $\alpha$  and  $\beta$ , the equations will take on the form of (19),

$$\begin{pmatrix} \alpha_{i+1} \\ \beta_{i+1} \end{pmatrix} = \begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} - \delta t \begin{pmatrix} \frac{d\langle E_L[\alpha] \rangle}{d\alpha} \\ \frac{d\langle E_L[\beta] \rangle}{d\beta} \end{pmatrix} \quad (20)$$

Where the  $\langle E_L \rangle$  is the expectation value of the local energy and  $\delta t$  is the step size of the method. As is evident, this method may oscillate around the solution, since the step size is fixed. Other more optimal methods such as the conjugate gradient method exist, but are not implemented in the project.

## 2.3 Electron wave function

Our wave function will consist of two parts: one that comes from the harmonic oscillator potential and is built up based on the fermionic nature of the system, and one that proved a correlation between two particles - the so-called Jastrow factor. The wave function we construct, will be called our *trial wave function*.

$$\psi_T(\mathbf{r}) = \psi_J(\mathbf{r})\psi_{SD}(\mathbf{r}) \quad (21)$$

### 2.3.1 Slater determinants

The wave function for an electron in a two dimensional harmonic oscillator potential can be written as a Hermite polynomial,

$$\phi_{n_x, n_y}(x, y) = A H_{n_x}(\sqrt{\omega\alpha}x) H_{n_y}(\sqrt{\omega\alpha}y) \exp\left(-\frac{\omega\alpha}{2}(x^2 + y^2)\right) \quad (22)$$

The details surrounding the mysterious variational parameter  $\alpha$  will be explained further on. For now, we will place these  $\phi$ 's into a Slater determinant. The Slater determinant is a creature that describes the wave function of a fermionic system, while also enforcing anti-symmetrization and thus the Pauli principle. Our Slater determinant will take the following form, when we describe the specific state of a system by  $n_x, n_y$  and a specific particle  $\mathbf{r}_i = x_i\mathbf{i} + y_i\mathbf{j}$ ,

$$Det(\Phi(\mathbf{r})) \equiv \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_1(\mathbf{r}_1) & \phi_2(\mathbf{r}_1) & \dots & \phi_N(\mathbf{r}_1) \\ \phi_1(\mathbf{r}_2) & \phi_2(\mathbf{r}_2) & \dots & \phi_N(\mathbf{r}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{r}_N) & \phi_2(\mathbf{r}_N) & \dots & \phi_N(\mathbf{r}_N) \end{vmatrix} \quad (23)$$

Note that we have defined  $\mathbf{r} \equiv (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$ . To pull this definition back to our trial wave function (21), we get

$$\psi_{SD}(\mathbf{r}) = Det(\Phi(\mathbf{r})) \quad (24)$$

Where the *OB* stands for one body, as in one body wave function. We now need to look into the part that accounts for many body effects, the Jastrow factor.

### 2.3.2 The Jastrow factor

The correlation term is called a *Jastrow factor* is as mentioned here to take into account many-body effects of our system. The general shape of it is

$$\psi_J(\mathbf{r}) = \prod_{i < j}^N \exp\left(\frac{ar_{ij}}{1 + \beta r_{ij}}\right) = \prod_{i=1}^N \prod_{j=i+1}^N \exp\left(\frac{ar_{ij}}{1 + \beta r_{ij}}\right) \quad (25)$$

where the  $C$  stands for correlation. The  $a$  is a parameter that is 1 for parallel spin, and 1/3 for anti-parallel spin. The  $\beta$  is another variational parameter and the  $r_{ij}$  is defined by the equation (4) as the distance between two electrons. For now, we shall begin by looking closer at the two-electron case.

## 2.4 Two electron case

For the two electron case the Hamiltonian takes on familiar form,

$$H = -\frac{1}{2} \left( \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial y_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial y_2^2} \right) + \frac{1}{2} \omega^2 (x_1^2 + y_1^2 + x_2^2 + y_2^2) + \frac{1}{r_{12}} \quad (26)$$

### 2.4.1 Unperturbed local energy

We begin by looking at the unperturbed local energy, in which we will try to convince ourselves that the local energy comes out to be  $2\omega$  in atomic units.

#### 2.4.2 Local energy

#### 2.4.3 Quantum force

### 2.5 $N$ electron case

#### 2.5.1 Unperturbed local energy

#### 2.5.2 Local energy

#### 2.5.3 Quantum force

## 3 Implementation

### 3.0.1 Slater determinants

An efficient way of updating the inverse of the Slater matrix can be found by [SITER](#) .

$$D_{kj}^{-1}(\mathbf{x}^{\text{new}}) = \begin{cases} D_{kj}^{-1}(\mathbf{x}^{\text{old}}) - \frac{D_{ki}^{-1}(\mathbf{x}^{\text{old}})}{R} \sum_{l=1}^N D_{il}(\mathbf{x}^{\text{new}}) D_{lj}^{-1}(\mathbf{x}^{\text{old}}) & \text{if } j \neq i \\ \frac{D_{ki}^{-1}(\mathbf{x}^{\text{old}})}{R} \sum_{l=1}^N D_{il}(\mathbf{x}^{\text{old}}) D_{lj}^{-1}(\mathbf{x}^{\text{old}}) & \text{if } j = i \end{cases} \quad (27)$$

## 4 Results

## 5 Discussion and conclusion

## 6 Appendix

### 6.1 Hermite polynomials

[REFERANSER!!](#)