Project 1, deadline March 10

Computational Physics I FYS4411/FYS9411

Department of Physics, University of Oslo, Norway

Spring semester 2017

Introduction

The two projects this year focus on properties of confined and interacting systems of electrons in two dimensions, so-called quantum dots.

We will deal only with systems where all possible single-particle states below a certain level are filled up. Such systems are called closed shell systems, a naming inspired from atomic and nuclear physics. These closed shell systems define what is frequently named **magic numbers**. Quantum dots exhibit also magic numbers, meaning that the addition or removal of one eletron requires more energy than systems where the lowest-lying shells are not filled. Using the harmonic oscillator in two dimensions as basis functions (with degenerate single-particle energies) the magic numbers are $N = 2$, $N = 6$, $N = 12$, $N = 20$ etc, where N is the number of electrons. See the table below for more details.

The first project deals with setting up a Hartree-Fock program that is flexible enough to handle different magic numbers. Hartree-Fock is the simplest possible many-body method. We will use the harmonic oscillator functions as basis functions and via the Hartree-Fock method modify the potential which confines the motion of the electrons by accounting for the Coulomb repulsion between electrons. The Hartree-Fock method leads to a set of coupled single-particle equations. If properly written, the system can easily be extended to three-dimensional electronic systems or other fermionic many-body systems.

The second project deals with so-called post Hartree-Fock methods, where additional correlations are introduced via different methods. We will focus on the variational Monte Carlo (VMC) approach, using the optimal Hartree-Fock basis from project 1 as input. If somebody wants to focus on other methods like Coupled-Cluster theory or in-medium Similarity renormalization group method (see chapters 8 and 11 of this reference), this can be tailored to your specific interests. In general, the VMC is much easier to implement and gives excellent approximations to the ground state properties.

We consider a system of electrons confined in a pure two-dimensional isotropic harmonic oscillator potential, with an idealized total Hamiltonian given by

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

where natural units ($\hbar = c = e = m_e = 1$) are used and all energies are in so-called atomic units a.u. We will study systems of many electrons N as functions of the oscillator frequency ω using the above Hamiltonian. The Hamiltonian includes a standard harmonic oscillator part

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

and the repulsive interaction between two electrons given by

$$\hat{H}_I = \sum_{i<j}^{N}\frac{1}{r_{ij}},$$

with the distance between electrons given by $r_{ij} = \sqrt{\boldsymbol{r}_1 - \boldsymbol{r}_2}$. We define the modulus of the positions of the electrons (for a given electron i) as $r_i = \sqrt{r_{i_x}^2 + r_{i_y}^2}$. There is no spin-orbit part in the two-body Hamiltonian and the different parts of the Hamiltonian have no spin operators. This means that the spin of a single-particle state or a many-particle state is a conserved quantity.

If only the harmonic oscillator part of the Hamiltonian, the so-called unperturbed part is given by

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

The wave function for one electron in an oscillator potential in two dimensions is

$$\phi_{n_x,n_y}(x,y) = AH_{n_x}(\sqrt{\omega}x)H_{n_y}(\sqrt{\omega}y)\exp\left(-\omega(x^2+y^2)/2\right).$$

The functions $H_{n_x}(\sqrt{\omega}x)$ are so-called Hermite polynomials, discussed in the appendix while A is a normalization constant. For the lowest-lying state we have $n_x = n_y = 0$ and an energy $\epsilon_{n_x,n_y} = \omega(n_x + n_y + 1) = \omega$.

In setting up both a Hartree-Fock code and a code for VMC calculations we need to set up a given single-particle basis. In our case the single-particle basis is provided by the harmonic oscillator functions. The first part of this project deals thus with the setup of the basis functions and the computation of matrix elements for the Coulomb interaction.

Project 1 a): getting started. Make a program that sets up all the quantum numbers for the single-particle basis using a harmonic oscillator in two dimensions. The list should contain the quantum numbers n_x and n_y, spin and its projection and the single particle energies in atomic units. Define a cutoff in the list according to the harmonic oscillator excitation energy. The table here lists the

first four harmonic oscillator shells, with its pertinent degeneracies, and the total number of electrons which can be accomodated up to a given energy. These numbers are the so-called *magic numbers* for a two-dimensional quantum dot. The degeneracies take into account the two spin values an electron can take. Convince yourself about the correctness of this table and use it to check that your code is running correctly.

Shell number	(n_x, n_y)	Energy	Degeneracy	N
1	$(0,0)$	$\hbar\omega$	2	2
2	$(1,0), (0,1)$	$2\hbar\omega$	4	6
3	$(2,0), (0,2), (1,1)$	$3\hbar\omega$	6	12
3	$(3,0), (0,3), (2,1), (1,2)$	$4\hbar\omega$	8	20

Convince yourself that the unperturbed lowest-lying energy for the two-electron system is simply 2ω. Similarly, find the corresponding energies for $N = 6$, $N = 12$, $N = 20$ and $N = 30$ electrons. These results will turn out to be very useful when you need to check your programs.

Project 1 b): Setting up the single-particle basis. With the single-particle basis from the previous step, you should now write a class which encodes the information about the harmonic oscillator single-particle basis. This should include the single-particle energies, spin and its projections as well as the harmonic oscillator functions that depend on n_x and n_y. This class can then used for both the Monte Carlo and the Hartree-Fock calculations.

Project 1 c): Computing the two-body matrix elements. In the Hartree-Fock project we need to be able to compute the two-electron matrix elements of the two-body Coulomb interaction. In order to set up the two-body matrix elements, we need to define the so-called direct and exchange matrix elements. This leads to what we call an anti-symmetrized matrix element.

We introduce the following shorthands for the integrals

$$\langle pq|\hat{v}|rs\rangle = \int \psi_p^*(x_i)\psi_q^*(x_j)V(r_{ij})\psi_r(x_i)\psi_s(x_j)dx_i dx_j,$$

which defines the so-called direct matrix element and

$$\langle pq|\hat{v}|sr\rangle = \int \psi_p^*(x_i)\psi_q^*(x_j)V(r_{ij})\psi_s(x_i)\psi_r(x_j)dx_i dx_j,$$

which defines the exchange element. The variables *pqrs* define all the single-particle quantum numbers. In our case these are n_x, n_y and spin and its projections.

These matrix elements are defined in terms of two-body quantum numbers. Which quantum numbers are conserved? Which are the possible values the total spin of a two-body state can have?

The direct and exchange matrix elements can be brought together if we define the anti-symmetrized matrix element

$$\langle pq|\hat{v}|rs\rangle_{\mathrm{AS}} = \langle pq|\hat{v}|rs\rangle - \langle pq|\hat{v}|sr\rangle.$$

It has the symmetry property

$$\langle pq|\hat{v}|rs\rangle_{\mathrm{AS}} = -\langle pq|\hat{v}|sr\rangle_{\mathrm{AS}} = -\langle qp|\hat{v}|rs\rangle_{\mathrm{AS}},$$

and

$$\langle pq|\hat{v}|rs\rangle_{\mathrm{AS}} = \langle qp|\hat{v}|sr\rangle_{\mathrm{AS}}.$$

Your task is to write a function which calculates the integral

$$\langle pq|\hat{v}|rs\rangle = \int \psi_p^*(x_i)\psi_q^*(x_j)V(r_{ij})\psi_r(x_i)\psi_s(x_j)dx_idx_j,$$

and then assemble the direct and exchange terms in order to construct the anti-symmetrized matrix elements to be used in the Hartree-Fock calculations. The integral is four-dimensional but can eventually be simplified. Your task is to figure out how. You should also consider whether it is appropriate to store the matrix elements for the Hartree-Fock calculations. Chapter five of the lecture notes discusses Gaussian quadrature. You may find it convenient to use Hermite polynomials in order to set up the integration points ad weights as well parallelizing the calculations of the integrals.

Project 1 d): Hartree-Fock equations. The Hartree-Fock functional is written as

$$E[\Phi] = \sum_{i \leq F}\langle i|\hat{h}_0|i\rangle + \frac{1}{2}\sum_{ij \leq F}\left[\langle ij|\hat{v}|ij\rangle - \langle ij|\hat{v}|ji\rangle\right],$$

where F indicates the Fermi level, or the last occupied single-particle state.

With the given functional, we can perform at least two types of variational strategies.

- Vary the Slater determinant by changing the spatial part of the single-particle wave functions themselves.

- Expand the single-particle functions in a known basis and vary the coefficients, that is, the new function single-particle wave function $|p\rangle$ is written as a linear expansion in terms of a fixed basis ϕ (harmonic oscillator, Laguerre polynomials etc)

$$\psi_p = \sum_{\lambda} C_{p\lambda}\phi_{\lambda},$$

Both cases lead to a new Slater determinant which is related to the previous via a unitary transformation. The second one is the one we will use in this project.

Consider a Slater determinant built up of single-particle orbitals ψ_λ, with $\lambda = 1, 2, \ldots, N$.

The unitary transformation

$$\psi_p = \sum_\lambda C_{p\lambda} \phi_\lambda,$$

brings us into the new basis. Show that the new basis is orthonormal. Show that the new Slater determinant constructed from the new single-particle wave functions can be written as the determinant based on the previous basis and the determinant of the matrix C. Show that the old and the new Slater determinants are equal up to a complex constant with absolute value unity. (Hint, C is a unitary matrix).

Minimizing with respect to $C_{p\alpha}^*$, remembering that $C_{p\alpha}^*$ and $C_{p\alpha}$ are independent and defining

$$h_{\alpha\gamma}^{HF} = \langle\alpha|\hat{h}_0|\gamma\rangle + \sum_{j \leq F} \sum_{\beta\delta} C_{j\beta}^* C_{j\delta} \langle\alpha\beta|\hat{v}|\gamma\delta\rangle_{AS},$$

show that you can write the Hartree-Fock equations as

$$\sum_\gamma h_{\alpha\gamma}^{HF} C_{p\gamma} = \epsilon_p C_{p\alpha}.$$

Explain the meaning of the different terms.

Project 1 e): Writing a Hartree-Fock program. The Hartree-Fock algorithm can be broken down as follows. We recall that our Hartree-Fock matrix is

$$\hat{h}_{\alpha\beta}^{HF} = \langle\alpha|\hat{h}_0|\beta\rangle + \sum_{j \leq F} \sum_{\gamma\delta} C_{j\gamma}^* C_{j\delta} \langle\alpha\gamma|\hat{v}|\beta\delta\rangle_{AS}.$$

Normally we assume that the single-particle basis $|\beta\rangle$ forms an eigenbasis for the operator $\hat{h}_0$ (this is our case, the harmonic oscillator energies), meaning that the Hartree-Fock matrix becomes

$$\hat{h}_{\alpha\beta}^{HF} = \epsilon_\alpha \delta_{\alpha,\beta} + \sum_{j \leq F} \sum_{\gamma\delta} C_{j\gamma}^* C_{j\delta} \langle\alpha\gamma|V|\beta\delta\rangle_{AS}.$$

The Hartree-Fock eigenvalue problem

$$\sum_\beta \hat{h}_{\alpha\beta}^{HF} C_{p\beta} = \epsilon_p^{HF} C_{p\alpha},$$

can be written out in a more compact form as

$$\hat{h}^{HF} \hat{C} = \epsilon^{HF} \hat{C}.$$

The equations are often rewritten in terms of a so-called density matrix, which is defined as

$$\rho_{\gamma\delta} = \sum_{i \leq F} \langle \gamma | i \rangle \langle i | \delta \rangle = \sum_{i \leq F} C_{i\gamma} C_{i\delta}^*. \tag{2}$$

It means that we can rewrite the Hartree-Fock Hamiltonian as

$$\hat{h}_{\alpha\beta}^{HF} = \epsilon_\alpha \delta_{\alpha,\beta} + \sum_{\gamma\delta} \rho_{\gamma\delta} \langle \alpha\gamma | V | \beta\delta \rangle_{AS}.$$

It is convenient to use the density matrix since we can precalculate in every iteration the product of two eigenvector components C.

Develop a Hartree-Fock program and perform Hartree-Fock calculations for $N = 2$, $N = 6$, $N = 12$ and $N = 20$ electrons. How many single-particle shells do you need before the total Hartree-Fock energy stabilizes as function of the number of shells? This is normally called the Hartree-Fock limit in the literature. Compare your Hartree-Fock energy with the unperturbed energy. Make sure that in case of no Coulomb interaction that your Hartree-Fock code reproduces the unperturbed energy.

How much do the single-particle energies change compared to the harmonic oscillator energies? Are the degeneracies seen in the harmonic oscillator calculations preserved?

Additional material on Hermite polynomials

The Hermite polynomials are the solutions of the following differential equation

$$\frac{d^2 H(x)}{dx^2} - 2x \frac{dH(x)}{dx} + (\lambda - 1)H(x) = 0. \tag{3}$$

The first few polynomials are

$$H_0(x) = 1,$$

$$H_1(x) = 2x,$$

$$H_2(x) = 4x^2 - 2,$$

$$H_3(x) = 8x^3 - 12x,$$

and

$$H_4(x) = 16x^4 - 48x^2 + 12.$$

They fulfil the orthogonality relation

$$\int_{-\infty}^{\infty} e^{-x^2} H_n(x)^2 dx = 2^n n! \sqrt{\pi},$$

and the recursion relation

$$H_{n+1}(x) = 2x H_n(x) - 2n H_{n-1}(x).$$

Literature

- S. Riemann and M. Manninen, Reviews of Modern Physics **74**, 1283 (2002).

- M. Taut, Physical Review A **48**, 3561 (1993).

- M. Pedersen Lohne, G. Hagen, M. Hjorth-Jensen, S. Kvaal, and F. Pederiva, Physical Review B **84**, 032501 (2011).

Introduction to numerical projects

Here follows a brief recipe and recommendation on how to write a report for each project.

- Give a short description of the nature of the problem and the eventual numerical methods you have used.

- Describe the algorithm you have used and/or developed. Here you may find it convenient to use pseudocoding. In many cases you can describe the algorithm in the program itself.

- Include the source code of your program. Comment your program properly.

- If possible, try to find analytic solutions, or known limits in order to test your program when developing the code.

- Include your results either in figure form or in a table. Remember to label your results. All tables and figures should have relevant captions and labels on the axes.

- Try to evaluate the reliabilty and numerical stability/precision of your results. If possible, include a qualitative and/or quantitative discussion of the numerical stability, eventual loss of precision etc.

- Try to give an interpretation of you results in your answers to the problems.

- Critique: if possible include your comments and reflections about the exercise, whether you felt you learnt something, ideas for improvements and other thoughts you've made when solving the exercise. We wish to keep this course at the interactive level and your comments can help us improve it.

- Try to establish a practice where you log your work at the computerlab. You may find such a logbook very handy at later stages in your work, especially when you don't properly remember what a previous test version of your program did. Here you could also record the time spent on solving the exercise, various algorithms you may have tested or other topics which you feel worthy of mentioning.

Format for electronic delivery of report and programs

The preferred format for the report is a PDF file. You can also use DOC or postscript formats or as an ipython notebook file. As programming language we prefer that you choose between C/C++, Fortran2008 or Python. The following prescription should be followed when preparing the report:

- Use Devilry to hand in your projects, log in at http://devilry.ifi.uio.no with your normal UiO username and password.

- Upload **only** the report file! For the source code file(s) you have developed please provide us with your link to your github domain. The report file should include all of your discussions and a list of the codes you have developed. The full version of the codes should be in your github repository.

- In your github repository, please include a folder which contains selected results. These can be in the form of output from your code for a selected set of runs and input parameters.

- Still in your github make a folder where you place your codes.

- In this and all later projects, you should include tests (for example unit tests) of your code(s).

- Comments from us on your projects, approval or not, corrections to be made etc can be found under your Devilry domain and are only visible to you and the teachers of the course.

Finally, we encourage you to work two and two together. Optimal working groups consist of 2-3 students. You can then hand in a common report.