

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 reliability 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 your 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 send the report 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 and choose FYS4411. There you can load up the files within the deadline.
- 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. Do not include library files which are available at the course homepage, unless you have made specific changes to them.
- In your git 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.
- In all projects, you should include a discussion of tests and possibly unit testing 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.

Project 1, Variational Monte Carlo studies of bosonic systems, The deadline for project 1 is March 4

The spectacular demonstration of Bose-Einstein condensation (BEC) in gases of alkali atoms ^{87}Rb , ^{23}Na , ^7Li confined in magnetic traps has led to an explosion of interest in confined Bose systems. Of interest is the fraction of condensed atoms, the nature of the condensate, the excitations above the condensate, the atomic density in the trap as a function of Temperature and the critical temperature of BEC, T_c .

A key feature of the trapped alkali and atomic hydrogen systems is that they are dilute. The characteristic dimensions of a typical trap for ^{87}Rb is $a_{ho} = (\hbar/m\omega_{\perp})^{\frac{1}{2}} = 1 - 2 \times 10^4 \text{ \AA}$. The interaction between ^{87}Rb atoms can be well represented by its s-wave scattering length, a_{Rb} . This scattering length lies in the range $85 < a_{Rb} < 140a_0$ where $a_0 = 0.5292 \text{ \AA}$ is the Bohr radius. The definite value $a_{Rb} = 100a_0$ is usually selected and for calculations the definite ratio of atom size to trap size $a_{Rb}/a_{ho} = 4.33 \times 10^{-3}$ is usually chosen. A typical ^{87}Rb atom density in the trap is $n \simeq 10^{12} - 10^{14} \text{ atoms/cm}^3$ giving an inter-atom spacing $\ell \simeq 10^4 \text{ \AA}$. Thus the effective atom size is small compared to both the trap size and the inter-atom spacing, the condition for diluteness ($na_{Rb}^3 \simeq 10^{-6}$ where $n = N/V$ is the number density).

Many theoretical studies of Bose-Einstein condensates (BEC) in gases of alkali atoms confined in magnetic or optical traps have been conducted in the framework of the Gross-Pitaevskii (GP) equation. The key point for the validity of this description is the dilute condition of these systems, that is, the average distance between the atoms is much larger than the range of the inter-atomic interaction. In this situation the physics is dominated by two-body collisions, well described in terms of the s-wave scattering length a . The crucial parameter defining the condition for diluteness is the gas parameter $x(\mathbf{r}) = n(\mathbf{r})a^3$, where $n(\mathbf{r})$ is the local density of the system. For low values of the average gas parameter $x_{av} \leq 10^{-3}$, the mean field Gross-Pitaevskii equation does an excellent job. However, in recent experiments, the local gas parameter may well exceed this value due to the possibility of tuning the scattering length in the presence of a so-called Feshbach resonance.

Thus, improved many-body methods like Monte Carlo calculations may be needed. The aim of this project is to use the Variational Monte Carlo (VMC) method and evaluate the ground state energy of a trapped, hard sphere Bose gas for different numbers of particles with a specific trial wave function.

This trial wave function is used to study the sensitivity of condensate and non-condensate properties to the hard sphere radius and the number of particles. The trap we will use is a spherical (S) or an elliptical (E) harmonic trap in one, two and finally three dimensions, with the latter given by

$$V_{ext}(\mathbf{r}) = \begin{cases} \frac{1}{2}m\omega_{ho}^2 r^2 & (S) \\ \frac{1}{2}m[\omega_{ho}^2(x^2 + y^2) + \omega_z^2 z^2] & (E) \end{cases} \quad (1)$$

where (S) stands for symmetric and

$$H = \sum_i^N \left(\frac{-\hbar^2}{2m} \nabla_i^2 + V_{ext}(\mathbf{r}_i) \right) + \sum_{i < j}^N V_{int}(\mathbf{r}_i, \mathbf{r}_j), \quad (2)$$

as the two-body Hamiltonian of the system. Here ω_{ho}^2 defines the trap potential strength. In the case of the elliptical trap, $V_{ext}(x, y, z)$, $\omega_{ho} = \omega_{\perp}$ is the trap frequency in the perpendicular or xy plane and ω_z the frequency in the z direction. The mean square vibrational amplitude of a single boson at $T = 0\text{K}$ in the trap (1) is $\langle x^2 \rangle = (\hbar/2m\omega_{ho})$ so that $a_{ho} \equiv (\hbar/m\omega_{ho})^{\frac{1}{2}}$ defines the characteristic length of the trap. The ratio of the frequencies is denoted $\lambda = \omega_z/\omega_{\perp}$ leading to a ratio of the trap lengths $(a_{\perp}/a_z) = (\omega_z/\omega_{\perp})^{\frac{1}{2}} = \sqrt{\lambda}$.

We will represent the inter-boson interaction by a pairwise, repulsive potential

$$V_{int}(|\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} \infty & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ 0 & |\mathbf{r}_i - \mathbf{r}_j| > a \end{cases} \quad (3)$$

where a is the so-called hard-core diameter of the bosons. Clearly, $V_{int}(|\mathbf{r}_i - \mathbf{r}_j|)$ is zero if the bosons are separated by a distance $|\mathbf{r}_i - \mathbf{r}_j|$ greater than a but infinite if they attempt to come within a distance $|\mathbf{r}_i - \mathbf{r}_j| \leq a$.

Our trial wave function for the ground state with N atoms is given by

$$\Psi_T(\mathbf{R}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N, \alpha, \beta) = \prod_i g(\alpha, \beta, \mathbf{r}_i) \prod_{i < j} f(a, |\mathbf{r}_i - \mathbf{r}_j|), \quad (4)$$

where α and β are variational parameters. The single-particle wave function is proportional to the harmonic oscillator function for the ground state, i.e.,

$$g(\alpha, \beta, \mathbf{r}_i) = \exp[-\alpha(x_i^2 + y_i^2 + \beta z_i^2)]. \quad (5)$$

For spherical traps we have $\beta = 1$ and for non-interacting bosons ($a = 0$) we have $\alpha = 1/2a_{ho}^2$. The correlation wave function is

$$f(a, |\mathbf{r}_i - \mathbf{r}_j|) = \begin{cases} 0 & |\mathbf{r}_i - \mathbf{r}_j| \leq a \\ (1 - \frac{a}{|\mathbf{r}_i - \mathbf{r}_j|}) & |\mathbf{r}_i - \mathbf{r}_j| > a. \end{cases} \quad (6)$$

a) Find analytic expressions for the local energy

$$E_L(\mathbf{R}) = \frac{1}{\Psi_T(\mathbf{R})} H \Psi_T(\mathbf{R}), \quad (7)$$

for the above trial wave function of Eq. (4). Find first the local energy the case with only the harmonic oscillator potential, that is we set $a = 0$. Use first that $\beta = 1$ and find the relevant local energies in one, two and three dimensions for one and N particles with the same mass.

Compute also the analytic expression for the drift force to be used in importance sampling

$$F = \frac{2\nabla \Psi_T}{\Psi_T}. \quad (8)$$

Find first the equivalent expressions for the just the harmonic oscillator part in one, two and three dimensions with $\beta = 1$.

Next, we will find the local energy for the full problem in three dimensions. The tricky part is to find an analytic expressions for the derivative of the trial wave function

$$\frac{1}{\Psi_T(\mathbf{R})} \sum_i^N \nabla_i^2 \Psi_T(\mathbf{R}),$$

with the above trial wave function of Eq. (4). We rewrite (and we can use the same general expressions for projects 2 and 3)

$$\Psi_T(\mathbf{R}) = \Psi_T(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N, \alpha, \beta) = \prod_i g(\alpha, \beta, \mathbf{r}_i) \prod_{i < j} f(a, |\mathbf{r}_i - \mathbf{r}_j|),$$

as

$$\Psi_T(\mathbf{R}) = \prod_i g(\alpha, \beta, \mathbf{r}_i) \exp\left(\sum_{i < j} u(r_{ij})\right)$$

where we have defined $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ and

$$f(r_{ij}) = \exp \left(\sum_{i < j} u(r_{ij}) \right),$$

with $u(r_{ij}) = \ln f(r_{ij})$. We have also

$$g(\alpha, \beta, \mathbf{r}_i) = \exp -\alpha(x_i^2 + y_i^2 + \beta z_i^2) = \phi(\mathbf{r}_i).$$

Show that the first derivative for particle k is

$$\nabla_k \Psi_T(\mathbf{R}) = \nabla_k \phi(\mathbf{r}_k) \left[\prod_{i \neq k} \phi(\mathbf{r}_i) \right] \exp \left(\sum_{i < j} u(r_{ij}) \right) + \prod_i \phi(\mathbf{r}_i) \exp \left(\sum_{i < j} u(r_{ij}) \right) \sum_{j \neq k} \nabla_k u(r_{ij}),$$

and find the final expression for our specific trial function. The expression for the second derivative is (show this)

$$\begin{aligned} \frac{1}{\Psi_T(\mathbf{R})} \nabla_k^2 \Psi_T(\mathbf{R}) &= \frac{\nabla_k^2 \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} + \frac{\nabla_k \phi(\mathbf{r}_k)}{\phi(\mathbf{r}_k)} \left(\sum_{j \neq k} \frac{\mathbf{r}_k}{r_k} u'(r_{kj}) \right) + \\ &\sum_{ij \neq k} \frac{(\mathbf{r}_k - \mathbf{r}_i)(\mathbf{r}_k - \mathbf{r}_j)}{r_{ki} r_{kj}} u'(r_{ki}) u'(r_{kj}) + \sum_{j \neq k} \left(u''(r_{kj}) + \frac{2}{r_{kj}} u'(r_{kj}) \right) \end{aligned}$$

Use this expression to find the final second derivative entering the definition of the local energy. You need to get the analytic expression for this expression using the harmonic oscillator wave functions and the correlation term defined in the project.

- b) Write a Variational Monte Carlo program which uses standard Metropolis sampling and compute the ground state energy of a spherical harmonic oscillator ($\beta = 1$) with no interaction and one dimension. Use natural units and make an analysis of your calculations using both the analytic expression for the local energy and a numerical calculation of the kinetic energy using numerical derivation. Compare the CPU time difference. The only variational parameter is α . Perform these calculations for $N = 1$, $N = 10$, 100 and 500 atoms. Compare your results with the exact answer. Extend then your results to two and three dimensions and compare with the analytical results.
- c) We repeat exercise b), but now we replace the brute force Metropolis algorithm with importance sampling based on the Fokker-Planck and the Langevin equations. Discuss your results and comment on eventual differences between importance sampling and brute force sampling. Run the calculations for the one, two and three-dimensional systems only and without the repulsive potential. Study the dependence of the results as a function of the time step δt . Compare the results with those obtained under b) and comment eventual differences.
- d) In performing the Monte Carlo analysis you should use blocking as a technique to make the statistical analysis of the numerical data. Present your results with a proper evaluation of the statistical errors. Repeat the calculations from exercise c) and include a proper error analysis. Limit yourself to the three-dimensional case only.
- e) We turn now to the elliptic trap with a repulsive interaction. We fix, as in Refs. [1,2] below, $a/a_{ho} = 0.0043$. Introduce lengths in units of a_{ho} , $r \rightarrow r/a_{ho}$ and energy in units of $\hbar\omega_{ho}$. Show then that the original Hamiltonian can be rewritten as

$$H = \sum_{i=1}^N \frac{1}{2} (-\nabla_i^2 + x_i^2 + y_i^2 + \gamma^2 z_i^2) + \sum_{i < j} V_{int}(|\mathbf{r}_i - \mathbf{r}_j|).$$

What is the expression for γ ? Choose the initial value for $\beta = \gamma = 2.82843$ and set up a VMC program which computes the ground state energy using the trial wave function of Eq. (4) using only α as variational parameter. Use standard Metropolis sampling and vary the parameter α in order to find a minimum. Perform the calculations for $N = 10, 50$ and $N = 100$ and compare your results to those from the ideal case in the previous exercises. In actual calculations employing the Metropolis algorithm, all moves are recast into the chosen simulation cell with periodic boundary conditions. To carry out consistently the Metropolis moves, it has to be assumed that the correlation function has a range shorter than $L/2$. Then, to decide if a move of a single particle is accepted or not, only the set of particles contained in a sphere of radius $L/2$ centered at the referred particle have to be considered. Benchmark your results with those of Refs. [1,2].

- f) Repeat the previous calculations by varying the energy using the conjugate gradient method or similar methods to obtain the best possible parameter α .
- g) With the optimal parameters for the ground state wave function, compute again the onebody density with and without the Jastrow factor. How important are the correlations induced by the Jastrow factor?

Literature

1. J. L. DuBois and H. R. Glyde, H. R., *Bose-Einstein condensation in trapped bosons: A variational Monte Carlo analysis*, Phys. Rev. A **63**, 023602 (2001).
2. J. K. Nilsen, J. Mur-Petit, M. Guilleumas, M. Hjorth-Jensen, and A. Polls, *Vortices in atomic Bose-Einstein condensates in the large-gas-parameter region*, Phys. Rev. A **71**, 053610 (2005).

How to write the report

What should the report contain and how can I structure it? A typical structure follows here.

- An abstract with the main findings.
- An introduction where you explain the aims and rationale for the physics case and what you have done. At the end of the introduction you should give a brief summary of the structure of the report
- Theoretical models and technicalities. This sections ends often being the methods section.
- Results and discussion
- Conclusions and perspectives
- Appendix with extra material
- Bibliography

Keep always a good log of what you do.

What should I focus on? Introduction.

You don't need to answer all questions in a chronological order. When you write the introduction you could focus on the following aspects

- Motivate the reader, the first part of the introduction gives always a motivation and tries to give the overarching ideas
- What I have done
- The structure of the report, how it is organized etc

What should I focus on? Methods sections.

- Describe the methods and algorithms
- You need to explain how you implemented the methods and also say something about the structure of your algorithm and present some parts of your code
- You should plug in some calculations to demonstrate your code, such as selected runs used to validate and verify your results. The latter is extremely important!! A reader needs to understand that your code reproduces selected benchmarks and reproduces previous results, either numerical and/or well-known closed form expressions.

What should I focus on? Results sections.

- Present your results
- Give a critical discussion of your work and place it in the correct context.
- Relate your work to other calculations/studies
- An eventual reader should be able to reproduce your calculations if she/he wants to do so. All input variables should be properly explained.
- Make sure that figures and tables contain enough information in their captions, axis labels etc so that an eventual reader can gain a first impression of your work by studying figures and tables only.

What should I focus on? Conclusions sections.

- State your main findings and interpretations
- Try as far as possible to present perspectives for future work
- Try to discuss the pros and cons of the methods and possible improvements

What should I focus on? Additional material, appendices.

- Additional calculations used to validate the codes
- Selected calculations, these can be listed with few comments
- Listing of the code if you feel this is necessary
- You can consider moving parts of the material from the methods section to the appendix. You can also place additional material on your webpage.

What should I focus on? References.

- Give always references to material you base your work on, either scientific articles/reports or books.
- Refer to articles as: name(s) of author(s), journal, volume (boldfaced), page and year in parenthesis.
- Refer to books as: name(s) of author(s), title of book, publisher, place and year, eventual page numbers

Format for electronic delivery of report and programs

You are free to choose your format for handing in. The simplest way is that you send us your github link that contains the report in your chosen format(pdf, ps, docx, ipython notebook etc) and the programs. As programming language you have to choose either C++ or Fortran or Python. We recommend C++ or Fortran. Finally, we recommend that you work together. Optimal working groups consist of 2-3 students, but more people can collaborate. You can then hand in a common report.

Unit tests, how and why?

Unit Testing is the practice of testing the smallest testable parts, called units, of an application individually and independently to determine if they behave exactly as expected. Unit tests (short code fragments) are usually written such that they can be preformed at any time during the development to continually verify the behavior of the code. In this way, possible bugs will be identified early in the development cycle, making the debugging at later stage much easier. There are many benefits associated with Unit Testing, such as

- It increases confidence in changing and maintaining code. Big changes can be made to the code quickly, since the tests will ensure that everything still is working properly.
- Since the code needs to be modular to make Unit Testing possible, the code will be easier to reuse. This improves the code design.
- Debugging is easier, since when a test fails, only the latest changes need to be debugged.
- Different parts of a project can be tested without the need to wait for the other parts to be available.
- A unit test can serve as a documentation on the functionality of a unit of the code.

Here follows a simple example, see the website of the course for more information on how to install unit test libraries.

```
#include <unittest++/UnitTest++.h>

class MyMultiplyClass{
public:
    double multiply(double x, double y) {
        return x * y;
    }
};

TEST(MyMath) {
    MyMultiplyClass my; CHECK_EQUAL(56, my.multiply(7,8));
}

int main()
{
    return UnitTest::RunAllTests();
}
```

For Fortran users, the link at <http://sourceforge.net/projects/fortranxunit/> contains a similar software for unit testing.