

# 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. As programming language we prefer that you choose between C/C++, Fortran90/95, Java or Python. Matlab/Maple/Mathematica/IDL are not allowed as programming languages for the handins, but you can use them to check your results where possible. The following prescription should be followed when preparing the report:

- Use Classfronter to hand in your projects, log in at blyant.uio.no and choose 'fellesrom fys3150 og fys4150'. Thereafter you will see an icon to the left with 'hand in' or 'innlevering'. Click on that icon and go to the given project. There you can load up the files within the deadline.
- Upload **only** the report file and the source code file(s) you have developed. 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.
- Comments from us on your projects, approval or not, corrections to be made etc can be found under your Classfronter domain and are only visible to you and the teachers of the course.

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

## Project 2, deadline 1 october 12am (midnight)

The task of this project is to integrate in a brute force manner a six-dimensional integral which is used to determine the ground state correlation energy between two electrons in a helium atom. We will employ both Gauss-Legendre quadrature and Monte-Carlo integration. Furthermore, you will need to parallelize your code for the Monte-Carlo integration.

The results you obtain here will be used in project four as well. We assume that the wave function of each electron can be modelled like the single-particle wave function of an electron in the hydrogen atom. The single-particle wave function for an electron  $i$  in the  $1s$  state is given in terms of a dimensionless variable (the wave function is not properly normalized)

$$\mathbf{r}_i = x_i \mathbf{e}_x + y_i \mathbf{e}_y + z_i \mathbf{e}_z,$$

as

$$\psi_{1s}(\mathbf{r}_i) = e^{-\alpha r_i},$$

where  $\alpha$  is a parameter and

$$r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}.$$

We will fix  $\alpha = 2$ , which should correspond to the charge of the helium atom  $Z = 2$ .

The ansatz for the wave function for two electrons is then given by the product of two  $1s$  wave functions as

$$\Psi(\mathbf{r}_1, \mathbf{r}_2) = e^{-\alpha(r_1+r_2)}.$$

Note that it is not possible to find an analytic solution to Schrödinger's equation for two interacting electrons in the helium atom.

The integral we need to solve is the quantum mechanical expectation value of the correlation energy between two electrons, namely

$$\left\langle \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \right\rangle = \int_{-\infty}^{\infty} d\mathbf{r}_1 d\mathbf{r}_2 e^{-2\alpha(r_1+r_2)} \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}. \quad (1)$$

Note that our wave function is not normalized. There is a normalization factor missing, but for this project we don't need to worry about that.

- a) Use Gauss-Legendre quadrature and compute the integral by integrating for each variable  $x_1, y_1, z_1, x_2, y_2, z_2$  from  $-\infty$  to  $\infty$ . How many mesh points do you need before the results converges at the level of the fourth leading digit? Hint: the single-particle wave function  $e^{-\alpha r_i}$  is more or less zero at  $r_i \approx 10 - 15$ . You can therefore replace the integration limits  $-\infty$  and  $\infty$  with  $-10$  and  $10$ , respectively. You need to check that this approximation is satisfactory.
- b) Compute the same integral but now with brute force Monte Carlo and compare your results with those from the previous point. Discuss the differences. With brute force we mean that you should use the uniform distribution.
- c) Improve your brute force Monte Carlo calculation by using importance sampling. Hint: use the exponential distribution. Does the variance decrease? Does the CPU time used compared with the brute force Monte Carlo decrease in order to achieve the same accuracy? Comment your results.
- d) Parallelize your code from the previous point and compare the CPU time needed with that from point [c)]. Do you achieve a good speedup?
- e) **Optional:** The integral of Eq. (1) has an analytical expression. Can you find it?