# 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 and choose either 'fys3150' or 'fys4150'. 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 this and all later projects, you should include 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 2, Schrödinger's equation for two electrons in a three-dimensional harmonic oscillator well, deadline October 5 (noon)

The aim of this project is to solve Schrödinger's equation for two electrons in a three-dimensional harmonic oscillator well with and without a repulsive Coulomb interaction. Your task is to solve this equation by reformulating it in a discretized form as an eigenvalue equation to be solved with Jacobi's method. To achieve this you will have to write your own code which implements Jacobi's method.

Electrons confined in small areas in semiconductors, so-called quantum dots, form a hot research area in modern solid-state physics, with applications spanning from such diverse fields as quantum nano-medicine to the contruction of quantum gates. You can read about quantum dots at `http://en.wikipedia.org/wiki/Quantum_dot`, which also contains links to several scientific articles. A recent article of interest is the review by Semonin *et al* in Materials Today, volume 15, page 508 (2012).

Here we will assume that these electrons move in a three-dimensional harmonic oscillator potential (they are confined by for example quadrupole fields) and repel each other via the static Colulomb interaction. We assume spherical symmetry.

We are first interested in the solution of the radial part of Schrödinger's equation for one electron. This equation reads

$$-\frac{\hbar^2}{2m}\left(\frac{1}{r^2}\frac{d}{dr}r^2\frac{d}{dr}-\frac{l(l+1)}{r^2}\right)R(r)+V(r)R(r)=ER(r).$$

In our case $V(r)$ is the harmonic oscillator potential $(1/2)kr^2$ with $k=m\omega^2$ and $E$ is the energy of the harmonic oscillator in three dimensions. The oscillator frequency is $\omega$ and the energies are

$$E_{nl}=\hbar\omega\left(2n+l+\frac{3}{2}\right),$$

with $n=0,1,2,\ldots$ and $l=0,1,2,\ldots$.

Since we have made a transformation to spherical coordinates it means that $r\in[0,\infty)$. The quantum number $l$ is the orbital momentum of the electron. Then we substitute $R(r)=(1/r)u(r)$ and obtain

$$-\frac{\hbar^2}{2m}\frac{d^2}{dr^2}u(r)+\left(V(r)+\frac{l(l+1)}{r^2}\frac{\hbar^2}{2m}\right)u(r)=Eu(r).$$

The boundary conditions are $u(0) = 0$ and $u(\infty) = 0$.

We introduce a dimensionless variable $\rho = (1/\alpha)r$ where $\alpha$ is a constant with dimension length and get

$$-\frac{\hbar^2}{2m\alpha^2}\frac{d^2}{d\rho^2}u(\rho) + \left(V(\rho) + \frac{l(l+1)}{\rho^2}\frac{\hbar^2}{2m\alpha^2}\right)u(\rho) = Eu(\rho).$$

We will set in this project $l = 0$. Inserting $V(\rho) = (1/2)k\alpha^2\rho^2$ we end up with

$$-\frac{\hbar^2}{2m\alpha^2}\frac{d^2}{d\rho^2}u(\rho) + \frac{k}{2}\alpha^2\rho^2u(\rho) = Eu(\rho).$$

We multiply thereafter with $2m\alpha^2/\hbar^2$ on both sides and obtain

$$-\frac{d^2}{d\rho^2}u(\rho) + \frac{mk}{\hbar^2}\alpha^4\rho^2u(\rho) = \frac{2m\alpha^2}{\hbar^2}Eu(\rho).$$

The constant $\alpha$ can now be fixed so that

$$\frac{mk}{\hbar^2}\alpha^4 = 1,$$

or

$$\alpha = \left(\frac{\hbar^2}{mk}\right)^{1/4}.$$

Defining

$$\lambda = \frac{2m\alpha^2}{\hbar^2}E,$$

we can rewrite Schrödinger's equation as

$$-\frac{d^2}{d\rho^2}u(\rho) + \rho^2u(\rho) = \lambda u(\rho).$$

This is the first equation to solve numerically. In three dimensions the eigenvalues for $l = 0$ are $\lambda_0 = 3, \lambda_1 = 7, \lambda_2 = 11, \ldots$.

We use the by now standard expression for the second derivative of a function $u$

$$u'' = \frac{u(\rho + h) - 2u(\rho) + u(\rho - h)}{h^2} + O(h^2), \tag{1}$$

where $h$ is our step. Next we define minimum and maximum values for the variable $\rho$, $\rho_{\min} = 0$ and $\rho_{\max}$, respectively. You need to check your results for the energies against different values $\rho_{\max}$, since we cannot set $\rho_{\max} = \infty$.

With a given number of steps, $n_{\text{step}}$, we then define the step $h$ as

$$h = \frac{\rho_{\max} - \rho_{\min}}{n_{\text{step}}}.$$

Define an arbitrary value of $\rho$ as

$$\rho_i = \rho_{\min} + ih \qquad i = 0, 1, 2, \ldots, n_{\text{step}}$$

we can rewrite the Schrödinger equation for $\rho_i$ as

$$-\frac{u(\rho_i + h) - 2u(\rho_i) + u(\rho_i - h)}{h^2} + \rho_i^2 u(\rho_i) = \lambda u(\rho_i),$$

or in a more compact way

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + \rho_i^2 u_i = -\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + V_i u_i = \lambda u_i,$$

where $V_i = \rho_i^2$ is the harmonic oscillator potential. Define first the diagonal matrix element

$$d_i = \frac{2}{h^2} + V_i,$$

and the non-diagonal matrix element

$$e_i = -\frac{1}{h^2}.$$

In this case the non-diagonal matrix elements are given by a mere constant. *All non-diagonal matrix elements are equal.* With these definitions the Schrödinger equation takes the following form

$$d_i u_i + e_{i-1} u_{i-1} + e_{i+1} u_{i+1} = \lambda u_i,$$

where $u_i$ is unknown. We can write the latter equation as a matrix eigenvalue problem

$$\begin{pmatrix} d_1 & e_1 & 0 & 0 & \dots & 0 & 0 \\ e_1 & d_2 & e_2 & 0 & \dots & 0 & 0 \\ 0 & e_2 & d_3 & e_3 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & d_{n_{\text{step}}-2} & e_{n_{\text{step}}-1} \\ 0 & \dots & \dots & \dots & \dots & e_{n_{\text{step}}-1} & d_{n_{\text{step}}-1} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ \dots \\ \dots \\ u_{n_{\text{step}}-1} \end{pmatrix} = \lambda \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ \dots \\ \dots \\ u_{n_{\text{step}}-1} \end{pmatrix} \tag{2}$$

or if we wish to be more detailed, we can write the tridiagonal matrix as

$$\begin{pmatrix} \frac{2}{h^2} + V_1 & -\frac{1}{h^2} & 0 & 0 & \dots & 0 & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2} + V_2 & -\frac{1}{h^2} & 0 & \dots & 0 & 0 \\ 0 & -\frac{1}{h^2} & \frac{2}{h^2} + V_3 & -\frac{1}{h^2} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \frac{2}{h^2} + V_{n_{\text{step}}-2} & -\frac{1}{h^2} \\ 0 & \dots & \dots & \dots & \dots & -\frac{1}{h^2} & \frac{2}{h^2} + V_{n_{\text{step}}-1} \end{pmatrix} \tag{3}$$

Recall that the solutions are known via the boundary conditions at $i = n_{\text{step}}$ and at the other end point, that is for $\rho_0$. The solution is zero in both cases.

a) Your task here is to write a function which implements Jacobi's rotation algorithm (see Lecture notes chapter 7) in order to solve Eq. (2).

We Define the quantities $\tan\theta = t = s/c$, with $s = \sin\theta$ and $c = \cos\theta$ and

$$\cot 2\theta = \tau = \frac{a_{ll} - a_{kk}}{2a_{kl}}.$$

We can then define the angle $\theta$ so that the non-diagonal matrix elements of the transformed matrix $a_{kl}$ become non-zero and we obtain the quadratic equation (using $\cot 2\theta = 1/2(\cot\theta - \tan\theta)$)

$$t^2 + 2\tau t - 1 = 0,$$

resulting in

$$t = -\tau \pm \sqrt{1 + \tau^2},$$

and $c$ and $s$ are easily obtained via

$$c = \frac{1}{\sqrt{1 + t^2}},$$

and $s = tc$. Explain why we should choose $t$ to be the smaller of the roots. Convince yourself that this choice ensures that $|\theta| \leq \pi/4$) and has the effect of minimizing the difference between the matrices $\mathbf{B}$ and $\mathbf{A}$ since

$$||\mathbf{B} - \mathbf{A}||_F^2 = 4(1 - c) \sum_{i=1, i\neq k,l}^{n} (a_{ik}^2 + a_{il}^2) + \frac{2a_{kl}^2}{c^2}.$$

(use the last equation to convince yourself that it ensures that $|\theta| \leq \pi/4$ is indeed satisfied).

b) How many points $n_{\text{step}}$ do you need in order to get the lowest three eigenvalues with four leading digits? Remember to check the eigenvalues for the dependency on the choice of $\rho_{\text{max}}$.

How many similarity transformations are needed before you reach a result where all non-diagonal matrix elements are essentially zero? Try to estimate the number of transformations and extract a behavior as function of the dimensionality of the matrix.

You can check your results against the code based on Householder's algorithm, *tqli* in the file lib.cpp. Alternatively, you can use the Armadillo function for solving eigenvalue problems.

Comment your results (here you could for example compute the time needed for both algorithms for a given dimensionality of the matrix).

c) We will now study two electrons in a harmonic oscillator well which also interact via a repulsive Coulomb interaction. Let us start with the single-electron equation written as

$$-\frac{\hbar^2}{2m}\frac{d^2}{dr^2}u(r) + \frac{1}{2}kr^2u(r) = E^{(1)}u(r),$$

where $E^{(1)}$ stands for the energy with one electron only. For two electrons with no repulsive Coulomb interaction, we have the following Schrödinger equation

$$\left(-\frac{\hbar^2}{2m}\frac{d^2}{dr_1^2} - \frac{\hbar^2}{2m}\frac{d^2}{dr_2^2} + \frac{1}{2}kr_1^2 + \frac{1}{2}kr_2^2\right)u(r_1, r_2) = E^{(2)}u(r_1, r_2).$$

Note that we deal with a two-electron wave function $u(r_1, r_2)$ and two-electron energy $E^{(2)}$.

With no interaction this can be written out as the product of two single-electron wave functions, that is we have a solution on closed form.

We introduce the relative coordinate $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$ and the center-of-mass coordinate $\mathbf{R} = 1/2(\mathbf{r}_1 + \mathbf{r}_2)$. With these new coordinates, the radial Schrödinger equation reads

$$\left( -\frac{\hbar^2}{m}\frac{d^2}{dr^2} - \frac{\hbar^2}{4m}\frac{d^2}{dR^2} + \frac{1}{4}kr^2 + kR^2 \right) u(r, R) = E^{(2)}u(r, R).$$

The equations for $r$ and $R$ can be separated via the ansatz for the wave function $u(r, R) = \psi(r)\phi(R)$ and the energy is given by the sum of the relative energy $E_r$ and the center-of-mass energy $E_R$, that is

$$E^{(2)} = E_r + E_R.$$

We add then the repulsive Coulomb interaction between two electrons, namely a term

$$V(r_1, r_2) = \frac{\beta e^2}{|\mathbf{r}_1 - \mathbf{r}_2|} = \frac{\beta e^2}{r},$$

with $\beta e^2 = 1.44$ eVnm.

Adding this term, the $r$-dependent Schrödinger equation becomes

$$\left( -\frac{\hbar^2}{m}\frac{d^2}{dr^2} + \frac{1}{4}kr^2 + \frac{\beta e^2}{r} \right) \psi(r) = E_r\psi(r).$$

This equation is similar to the one we had previously in (a) and we introduce again a dimensionless variable $\rho = r/\alpha$. Repeating the same steps as in (a), we arrive at

$$-\frac{d^2}{d\rho^2}\psi(\rho) + \frac{1}{4}\frac{mk}{\hbar^2}\alpha^4\rho^2\psi(\rho) + \frac{m\alpha\beta e^2}{\rho\hbar^2}\psi(\rho) = \frac{m\alpha^2}{\hbar^2}E_r\psi(\rho).$$

We want to manipulate this equation further to make it as similar to that in (a) as possible. We define a 'frequency'

$$\omega_r^2 = \frac{1}{4}\frac{mk}{\hbar^2}\alpha^4,$$

and fix the constant $\alpha$ by requiring

$$\frac{m\alpha\beta e^2}{\hbar^2} = 1$$

or

$$\alpha = \frac{\hbar^2}{m\beta e^2}.$$

Defining

$$\lambda = \frac{m\alpha^2}{\hbar^2}E,$$

we can rewrite Schrödinger's equation as

$$-\frac{d^2}{d\rho^2}\psi(\rho) + \omega_r^2\rho^2\psi(\rho) + \frac{1}{\rho} = \lambda\psi(\rho).$$

We treat $\omega_r$ as a parameter which reflects the strength of the oscillator potential.

Here we will study the cases $\omega_r = 0.01$, $\omega_r = 0.5$, $\omega_r = 1$, and $\omega_r = 5$ for the ground state only, that is the lowest-lying state.

With no repulsive Coulomb interaction you should get a result which corresponds to the relative energy of a non-interacting system. Make sure your results are stable as functions of $\rho_{\max}$ and the number of steps.

We are only interested in the ground state with $l = 0$. We omit the center-of-mass energy.

You can reuse the code you wrote for (a), but you need to change the potential from $\rho^2$ to $\omega_r^2 \rho^2 + 1/\rho$.

Comment the results for the lowest state (ground state) as function of varying strengths of $\omega_r$.

For specific oscillator frequencies, the above equation has answers in an analytical form, see the article by M. Taut, Phys. Rev. A 48, 3561 - 3566 (1993). The article can be retrieved from the following web address `http://prola.aps.org/abstract/PRA/v48/i5/p3561_1`.

d) In this exercise we want to plot the wave function for two electrons as functions of the relative coordinate $r$ and different values of $\omega_r$. With no Coulomb interaction you should have a result which corresponds to the non-interacting case. Plot either the function itself or the probability distribution (the function squared) with and without the repulsion between the two electrons. Varying $\omega_r$, the shape of the wave function will change.

We are only interested in the wave function for the ground state with $l = 0$ and the two first excited states with the same symmetry and omit again the center-of-mass motion.

You can choose between three approaches; the first is to use the existing *tqli* function. Here the eigenvectors are obtained from the matrix $z[i][j]$, where the index $j$ refers to eigenvalue $j$. The index $i$ points to the value of the wave function in position $\rho_j$. That is, $u^{(\lambda_j)}(\rho_i) = z[i][j]$.

The eigenvectors are normalized. Plot then the normalized wave functions for different values of $\omega_r$ and comment the results.

Another alternative is to add a piece to your Jacobi routine which also returns the eigenvectors. This is the more difficult part. You will also need to normalize the eigenvectors.

Finally, the armadillo function *eig_sys* can be used to find eigenvalues and eigenvectors.

e) Here you are asked to implement unit tests in either your C++ program or your Fortran program. For C++ users, please follow the guidelines on how to install unit tests with c++ on the webpage. For Fortran user, the software package Fortran Unit Test Framework (FRUIT), `sourceforge.net/projects/fortranxunit`. These issues will be discussed in more detail at the lab and lectures during week 38.

f) This exercise is optional and is meant more as a challenge. Implement the iterative Lanczos' algorithm discussed in the lecture notes and compute the lowest eigenvalues as done in exercise c) above. Compare your results and discuss faults and merits of the iterative method versus direct methods like Jacobi's method.