# Project 1 (Midterm project)

The completed project will consist of a series of programs and routines, datafiles, and a PDF document containing your written answers and figures. Put everything in a single directory, archive it, and send it to me by email by the deadline.

*General Guidelines:* You may inspire yourself from existing publicly available codes, but your programs and routines *must* be written by you. Any obvious "copy-and-paste" similarities with existing codes will be treated as plagiarism. If you inspire yourself from an existing code, write a comment in the header stating your source. Make your programs and routines as versatile as possible (i.e. don't "hard-wire" anything if possible). You can include the routines in your `LinAl.f90` module, or leave them out of it, it's up to you. Think carefully about the input and output variables for the routines. Comment your code thoroughly (i.e. all the steps taken). Make the code easily readable following the suggestions discussed in Section 4.

Note that if you are unable to write the code for Questions 1 and 2, you *may* use canned routines (e.g. LAPACK) in Questions 3 and 4, but you will not get credit for these questions.

**Question 1:** QR decomposition.

*Code to be produced:* Write a routine that returns the QR decomposition of a non-square $n \times m$ matrix. Note that you do *not* necessarily have to return the full matrices $\mathbf{Q}$ and $\mathbf{R}$, but you can if you prefer to. Your routine does not have to be ideally optimized, it just needs to work!

*PDF:* In the first Section of your PDF document, create a brief documentation of how someone should use your routine, including what the required inputs are, what the outputs are, and what limitations this routine may have (i.e. how NOT to use the routine, if applicable).

**Question 2:** QR backsubstitution.

*Code to be produced:* Write a routine that solves $\mathbf{R}_m\mathbf{x} = \mathbf{Q}_m^T\mathbf{b}$, where $\mathbf{b}$ is a RHS vector of size $m \times 1$, and where $\mathbf{R_m}$ and $\mathbf{Q}_m^T$ where defined in the notes to contain only the first $m$ lines of the $\mathbf{Q}^T$ and $\mathbf{R}$ matrices obtained as the result of the call to the QR decomposition routine written in Section 1.

*PDF:* In the second Section of your PDF document, create a brief documentation of how someone should use your routine, including what the required inputs are, what the outputs are, and what limitations this routine may have (i.e. how NOT to use the routine).

**Question 3:** Comparison of the QR and Cholesky algorithm on the Atkinson data.

*Code to be produced:* Create a program that reads in the `atkinson.dat` file of Homework 3, and performs a Least Square fit of that data for a cubic function using the QR method instead of the Cholesky method. The program should:

- read in the name of the file to be analyzed (in this particular question, `atkinson.dat`)
- return a data file called `parameters.dat` that returns the parameters of the fit and the rms error of the fit.
- return a data file called `fitted.dat` that contains the fitted data in the form of two columns readable by gnuplot (i.e. formatted as in the `atkinson.dat` file)

*PDF*: In the third Section of your PDF document, compare the output of this new code with the output of the Cholesky method applied to the same data (fitting the same cubic). Compare the parameters obtained in both cases, and the rms error. Create a plot that contains the original data, and the curves obtained by fitting the data with the two different methods. Briefly discuss your findings. Hint: the answers obtained in both codes should be quite similar for this dataset, if they are not, there is a problem with your new code.

**Question 4:** The failure of the Cholesky method

We now continue to fit the same `atkinson.dat` data file, but this time we no longer assume that the polynomial to be fitted is a cubic. We shall try different order polynomials and see how sensitive the solution is to the selected order.

*Code to be produced/modified:* If needed at this point, modify both your Cholesky least-square program from Homework 3, and your QR least-square program to be able to fit any order polynomial, where the order is input by the user (either as an argument to the program, or by a prompt – whichever you prefer). Return (as part of your answers) the

- Modified Cholesky Least-Square program and associated routines
- Modified QR Least-Square program and associated routines.

*PDF*: In the fourth Section of your PDF document, create a Table or a graph (whichever you prefer) that shows how the rms error (obtained by the two methods different methods applied to the `atkinson.dat` data file) changes with the order of the polynomial we fit. Vary the order from 3 to 10. What do you notice? Hint: read the title of the question before you go crazy debugging your Cholesky program.

**Question 5:**

To understand what is happening you will need to do a bit of research and reading.

- Find some references about the *condition number* of a matrix.
- Find some references about *Vandermonde* matrices.

*PDF*: In the fifth section of your PDF,

- Briefly define the condition number, and discuss the importance of the condition number of a matrix $\mathbf{A}$ on the stability of the numerical solution of the general linear system $\mathbf{Ax} = \mathbf{b}$. In what follows you may assume without proving it that if the condition number of $\mathbf{A}$ is large, then that of $\mathbf{A}^T\mathbf{A}$ is even larger.
- What could this imply for the solutions of some "square" Least-Square problem (i.e. when the number of data points is equal to the order of the polynomial minus 1) via the Cholesky method vs. via the QR method?
- Briefly discuss why Vandermonde matrices are relevant to Least-Square problems, and what is known about the condition number of Vandermonde matrices as a function of their size.
- Putting all these pieces of information together, how does this explain the results of Question 4? Conclude by discussing (briefly) the often-found statement: "Because of the poor stability properties of the Cholesky method, the latter is rarely used for Least-Square problems".

Provide references for all your sources of information.