AM 213A, Winter 2022
Homework 3 (100 points)

Posted on Mon, Feb 14, 2022
Due Thu, Feb 24, 2022

Submit your homework to your Git repository by 11:59 pm

---

1. General Guidelines

1.1. Two parts

You have two parts in this homework set:

- **Part 1 (50 pts)**: Numerical coding problems (Make sure to use double precision unless you are told otherwise!!!)
- **Part 2 (50 pts)**: Theory problems

Your final set of answers should consist of

- For Part 1: a pdf file with your written answers. This pdf *must* be created using a word processor. Prepare a "README" file under each subdirectory to give a short description of how to execute your code implementations, e.g., "To run the code, execute make first; run the executable file `hw3_exe`; the name of the driver routine – `hw3_cholesky.f90`". Each README file should be short and concise to provide first-hand guidelines on executing your codes.
- For Part 2: solutions to the theory problems in a pdf file (a cleanly handwritten scanned pdf is also allowed).
- For both Part 1 and Part 2: Please submit all your answers to your git. Any submission to Canvas is not going to be graded. Use subdirectories named as **Part1** and **Part2** under your top directory called **HW3**.

Please organize and present the material in the best possible way, in particular, for your Part 1 solutions:

- For any written material, be informative but concise. Mathematical derivations, if appropriate, should contain enough pertinent steps to show the reader how you proceeded. All included figures should be clearly annotated and have an informative caption. Each problem should be addressed in its own Section (e.g., "Section 1: Problem 1", "Section 2: Problem 2", etc.). References should be included if you are using any material as part of your discussions and/or calculations.

- For the codes: All codes must be written in Fortran 90/C (or more recent Fortran, but not Fortran 77) or C, but *not* in any other language. MATLAB or Python is to be used to (i) generate figures, and/or (ii) perform simple analysis (e.g., calculating errors) of the data produced by your Fortran/C codes. Put all the codes relevant for each problem in separate directories called Prob1, Prob2, etc. Annotate your codes carefully so the readers know what each part of the code does, in addition to the short instruction in README. For each routine and function in your Fortran/C, include a header (i.e., comment sections at the beginning explanations) explaining the inputs and outputs of the code, and what the routine does. In the main program (i.e., driver routine) in each directory, include a header that contains (i) the command required to compile the code, and (ii) the structure of the inputs and outputs of the main program. You can copy and paste this information to README files.
- Template codes are provided in Fortran 90/C; for C routines, see the announcement on Canvas by TA Ian May on Feb 3rd, 2022.

Part 1: Coding Problems

Download the `least_squares_data.dat` data file, which contains a data set whose first column are the values of x and the second column are the corresponding y values. There are 21 lines to this file.

(1) Cholesky solution of the least-squares problem (25 points)

In your `LinAl.f90` module from Homework 2, create a Cholesky decomposition routine that takes the following information as input and output arguments:

- a square matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$ (input)
- a logical flag (i.e., `.TRUE.` or `.FALSE.`) that indicates whether the problem is singular or not, and whether the matrix is SPD or not (output). Execute the Cholesky decomposition first and use its outcome to determine if a matrix is SPD or not.

On exit, $\mathbf{A}$ is replaced by its Cholesky decomposition in the lower triangle, and the original matrix $\mathbf{A}$ in the upper triangle (minus the diagonal). The logical is set to be `.FALSE.` if there were no problems; `.TRUE.` if the matrix is either singular or not SPD. You may add, as an extra argument, the original diagonal of $\mathbf{A}$ so $\mathbf{A}$ is effectively saved.

Then create a backsubstitution routine that solves $\mathbf{L}\mathbf{L}^T\mathbf{x} = \mathbf{b}$. The routine takes as the following information as arguments:

- the matrix $\mathbf{A}$ already Cholesky-decomposed (input)
- its first dimension (input)

- a vector $\mathbf{b}$ (Note : you can generalize this and input a matrix $\mathbf{B}$ containing n rhs vectors if you prefer, i.e., $\mathbf{B} = [\mathbf{b}_1 \sqcup \mathbf{b}_2 \sqcup \cdots \sqcup \mathbf{b}_n]$, $\mathbf{b}_i \in \mathbb{R}^m$) (input)

You may choose to over-write the vector $\mathbf{b}$ (or $\mathbf{b}_i$) on exit with the solution $\mathbf{x}$ (or $\mathbf{x}_i$), or return the solution as a separate vector (in which case it has to be added to the argument list).

Finally, create a program that

- reads the `least_squares_data.dat` data file
- creates the matrices associated with the normal equations (you will use a polynomial fitting to approximate `least_squares_data.dat` in this problem and the next problem. To do this, you will need to use the Vandermonde matrices as discussed in the lecture note.)
- solves the normal equations using Cholesky decomposition and back-substitution
- prints the solution vector $\mathbf{x}$
- verifies that the solution is correct by calculating its 2-norm error
- computes the fitted curve, and print it to a file
- calculates the 2-norm error between the fitted curve and the data
- use Python or Matlab to plot your results (choose carefully what kind of information you want to show in your figure(s) to best present your results)

In your PDF document:

- explain in reasonable detail how the Cholesky decomposition and back-substitution work.
- fit the data with a 3rd-degree polynomial, using single-precision floating-point arithmetics. Report what the polynomial coefficients you find are, what the 2-norm error on the fit is, and provide a figure comparing the fitted curve with the data. Note that the choice of single- vs. double-precision can be easily done in a Makefile by managing corresponding flag options (let me or TA know if you have questions on it).
- fit the data with a 5th-degree polynomial, using single-precision floating-point arithmetics. Report what the polynomial coefficients you find are, what the 2-norm error on the fit is, and provide a figure comparing the fitted curve with the data.
- discuss what should be the maximum degree of polynomials for the given data? Explain briefly what happens when you try to fit the data with a higher-degree polynomial, and why.
- discuss at what point does this algorithm fail (in single-precision floating-point arithmetic)? Or, does the algorithm works always? Discuss.

(2) QR solution of the least-squares problem (25 points)

In your `LinAl.f90` module, create and add a routine that performs a Householder-based QR decomposition of the matrix A. You are free to choose the argument list but explain very clearly which of each argument is an input and/or an output.

If your QR decomposition routine does *not* return $\mathbf{Q}$ and $\mathbf{R}$, create a separate routine that takes the outputs from your QR decomposition routine and computes $\mathbf{Q}$ and $\mathbf{R}$. Check the correctness of your QR decompositions using the Householder QR factorization example in the lecture note.

Finally, create a program that

- reads the `least_squares_data.dat` data file and create the matrices $\mathbf{A}$ and $\mathbf{b}$
- performs a QR decomposition on $\mathbf{A}$
- computes $\mathbf{A} - \mathbf{QR}$ and prints it to the screen
- computes either the 2-norm or the Frobenius norm of $\mathbf{A} - \mathbf{QR}$ and prints it to the screen
- computes $\mathbf{Q}^T\mathbf{Q} - \mathbf{I}$ and prints it to the screen
- computes either the 2-norm or the Frobenius norm of $\mathbf{Q}^T\mathbf{Q} - \mathbf{I}$ and prints it to the screen
- solves the least squares equations projected onto the span of $\mathbf{A}$, namely the resulting reduced square system, $\hat{\mathbf{R}}\mathbf{x} = \hat{\mathbf{Q}}^T\mathbf{b}$ as in the notes
- prints the solution vector $\mathbf{x}$
- verifies that the solution is correct by calculating its 2-norm error

In your PDF document:

- explain in detail how the Householder QR decomposition works. In particular, calculate explicitly the product of $\mathbf{H}_j$ with $\mathbf{A}$ at step j to prove that it indeed casts $\mathbf{A}$ in the right form, and explicitly show that the new diagonal element of $\mathbf{A}$ is $-s_j$.
- fit the data with a 3rd-degree polynomial, using single-precision floating point arithmetic. Report what the polynomial coefficients you find are, what the 2-norm error on the fit is, and provide a figure comparing the fitted curve with the data (again, you will need to use the Vandermonde matrices).
- fit the data with a 5th-degree polynomial, using single-precision floating point arithmetic. Report what the polynomial coefficients you find are, what the 2-norm error on the fit is, and provide a figure comparing the fitted curve with the data.

- discuss at what point does this algorithm fail (in single-precision floating point arithmetic)? Or, does the algorithm works always? Discuss.
- for the 5th order polynomial case, discuss your findings about the 2-norms of $\mathbf{A} - \mathbf{QR}$ and $\mathbf{Q}^T\mathbf{Q} - \mathbf{I}$.

Part 2: Theory Problems (50 points; 10 points each)

1. If P is an orthogonal projector, then $I - 2P$ is unitary.

2. Let $P \in \mathbb{R}^{m\times m}$ be a nonzero projector.

(a) Show that $||P||_2 \geq 1$, with equality if and only if P is an orthogonal projector.

(b) If P is an orthogonal projector, then P is positive semi-definite with its eigenvalues are either zero or 1.

3. Let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$, and let $A = \hat{Q}\hat{R}$ be a reduced QR factorization.

(a) Show that A has rank n if and only if all the diagonal entries of $\hat{R}$ are nonzero.

(b) Suppose $\hat{R}$ has k nonzero diagonal entries for some k with $0 \leq k < n$. What does this imply about the rank of A? Exactly k? At least k? At most k? Give a precise answer and prove it.

4. Determine the (i) eigenvalues, (ii) determinant, and (iii) singular values of a Householder reflector. For the eigenvalues, give a geometric argument as well as an algebraic proof.

5. Let $A \in \mathbb{R}^{m\times n}$. Show that $\text{cond}(A^TA) = \Big(\text{cond}(A)\Big)^2$.