# Graded Homework 2 Numerical Linear Algebra 2024

**The solutions are due by 11:59PM on Friday December 6, 2024**

Note since the full-rank linear least squares was not assigned I have posted the code and included the description of such and assignment here.

# Programming Exercise

# Problem 2.1

## 2.1.a Overview

Your general task is to implement two basic approaches to solving the linear least squares problem

$$\min_{x \in \mathbb{R}^k} \|b - Ax\|_2$$

given $b \in \mathbb{R}^n$ and the matrix $A \in \mathbb{R}^{n \times k}$ with linearly independent columns.

The two algorithms are:

1. the approach based on the problem transformation

$$H_k H_{k-1} \cdots H_1 A = \begin{pmatrix} R \\ 0 \end{pmatrix}, \quad H_k H_{k-1} \cdots H_1 b = \begin{pmatrix} c \\ d \end{pmatrix},$$

   where $H_i$, $1 \leq i \leq k$ are Householder reflectors, $R \in \mathbb{R}^{k \times k}$ is a nonsingular upper triangular matrix with positive elements on the diagonal, $c \in \mathbb{R}^k$, and $d \in \mathbb{R}^{n-k}$.

2. The incremental algorithm for updating the solution

$$x_{min}^{(n)} = \underset{x \in \mathbb{R}^k}{\operatorname{argmin}} \|b_n - A_n x\|_2$$

   to

$$x_{min}^{(n+1)} = \underset{x \in \mathbb{R}^k}{\operatorname{argmin}} \|b_{n+1} - A_{n+1} x\|_2$$

   where all of the matrices are as defined in the class notes. You may implement the weighted version if you wish.

All routines should be implemented to be as efficient as possible in space and operations. Your solutions must be clear and concise about justifying your design.

### 2.1.b Tasks

Design, execute and summarize tests to demonstrate that your codes are correct. This should use specific problems with particular properties to make a point, sets of problems to show overall statistical behavior of the accuracy for particular classes of problem, problems with a range of dimensions. You should also show that your incremental code produces consistent results with your "full problem" code.

You may use state-of-the-art libraries and problem solving environments, e.g., Matlab, to assist in designing, testing, and organizing your results. For example, you may use Matlab to check the solutions by comparing it to yours or to generate test problems. As noted in the syllabus, you may also use Matlab or related languages to implement your codes. However, your codes must be written in such a way that the data structures and control structures are clear and easily translated into a compiled and typed language.

### 2.1.c Comments on Test Problems

For the linear least squares codes and experiments, you should include for various values of $n$, $k$, and $b$ problems for at least the three situations:

1. $n = k$, i.e., a square nonsingular matrix $A$ where $x_{min} = A^{-1}b$.
2. $n > k$ and $Ax = b$ for $b \in \mathbb{R}^n$ and $b \in \mathcal{R}(A)$ i.e., a rectangular matrix $A$ with full column rank and a vector $b$ that define a consistent set of overdetermined equations.
3. $n > k$ and $b \in \mathbb{R}^n$ and $b \notin \mathcal{R}(A)$ i.e., a rectangular matrix $A$ with full column rank and a vector $b = b_1 + b_2$, $b_1 \in \mathcal{R}(A)$, $b_2 \notin \mathcal{R}(A)$, $b_2 \neq 0$, that define a linear least squares problem with a nonzero residual $r_{min} = b_2 = b - Ax_{min}$

This requires careful construction of the test problems.

When generating test problems, you may use library routines available in whatever language environment you are using. The simplest and most controlled is to generate a problem in the structurally orthogonal coordinate system, i.e.,

$$\min_{x \in \mathbb{R}^k} \| \begin{pmatrix} c \\ d \end{pmatrix} - \begin{pmatrix} R \\ 0 \end{pmatrix} x \|_2^2, \quad R \in \mathbb{R}^{k \times k}, \quad c \in \mathbb{R}^k, \quad d \in \mathbb{R}^{n-k},$$

and then generate random orthogonal reflectors to put the problem into a typical coordinate system:

$$A = H_s H_{s-1} \dots H_2 H_1 \begin{pmatrix} R \\ 0 \end{pmatrix} \in \mathbb{R}^{n \times k} \text{ and } b = H_s H_{s-1} \dots H_2 H_1 \begin{pmatrix} c \\ d \end{pmatrix} \in \mathbb{R}^n.$$

This approach allows you to control the norm of the residual, thereby allowing the easy construction of consistent overdetermined systems and true least squares problems. Similarly, this allows you to control the amount of $b$ that is in the perpendicular space and therefore the conditioning of the problem.

This can also be used to initialize a problem for the incremental method.