Problem Set 1

Due Date: March 1

Write a code that uses the double Gram-Schmidt procedure to invert a given square matrix A. Use the fact that

$$A \cdot G = U \quad \Rightarrow \quad A^{-1} = G \cdot U^*,$$

where U is an orthogonal matrix and U^* is the adjoint (transpose) of U. One may obtain the matrix G by applying the same set of transformations to the identity matrix as is applied to A in the Gram-Schmidt procedure.

In Fortran, your calling sequence should be

$$inv\_double\_gs(a, n, u, b) \tag{1}$$

where

$a(n, n)$ is a (real*8) $n \times n$-matrix to be inverted (input parameter)

n is the (integer) size of the matrix (input parameter)

$u(n, n)$ is a (real*8) $n \times n$-matrix (output parameter). The matrix u is orthogonal, and its columns are the result of the double Gram-Schmidt process applied to the columns of a

$b(n, n)$ is a (real*8) $n \times n$-matrix (output parameter). The matrix b is the inverse of a

In C, your calling sequence should be

$$void\ inv\_double\_gs(double *\ a, int\ n, double *\ u, double *\ b), \tag{2}$$

where

a points to an array of doubles of size n^2, containing $a(1,1), a(1,2), \ldots, a(1,n), a(2,1), \ldots, a(n,n)$, a being the $n \times n$ matrix to be inverted (input parameter)

n is the (integer) size of the matrix (input parameter)

u points to an array of doubles of size n^2, containing $u(1,1), u(1,2), \ldots, u(1,n), u(2,1), \ldots, u(n,n)$, u being the $n \times n$ orthogonal matrix. The columns of u are the result of the double Gram-Schmidt process applied to the columns of a (output parameter, memory allocated by the user)

b points to an array of doubles of size n^2, containing $b(1,1), b(1,2), \ldots, b(1,n), b(2,1), \ldots, b(n,n)$, b being the $n \times n$ matrix. The matrix b is the inverse of a (output parameter, memory allocated by the user)