ENGR 695 Advanced Topics in Engineering Mathematics Fall 2023

Lecture Outline for Wednesday, Sept. 20

1. *LU* factorization

   a. Express $N \times N$ matrix A as $A = LU$, where L is a lower triangular matrix and U is upper triangular:

$$L = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & & \vdots \\ \vdots & \vdots & \ddots & 0 \\ l_{N1} & l_{N,2} & \cdots & l_{NN} \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1N} \\ 0 & u_{22} & \cdots & u_{2N} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & u_{NN} \end{bmatrix}$$

   b. One application is solution of $A\mathbf{x} = \mathbf{b}$:
   $A\mathbf{x} = \mathbf{b} \rightarrow LU\mathbf{x} = \mathbf{b}$. Let $U\mathbf{x} = \mathbf{y} \rightarrow L\mathbf{y} = \mathbf{b}$. Solve $L\mathbf{y} = \mathbf{b}$ for $\mathbf{y}$ and then $U\mathbf{x} = \mathbf{y}$ for $\mathbf{x}$. Both solutions are straightforward using forward and backward substitution.

   c. Example: Solve $A\mathbf{x} = \mathbf{b}$, where

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 3 & 1 & 2 \\ 1 & -1 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 9 \\ 16 \\ 1 \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 1 & 1 & 1 \\ 0 & -2 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

   d. LU factorizations are not typically unique. Results depend on method used.

   e. Matlab command `[L U] = lu(A)`

   f. Determinants: $\det(A) = \det(L)\det(U)$

2. *QR* factorization

   a. *LU* factorization is for square systems; *QR* factorization is for over- and underdetermined systems ($M > N$ or $M < N$)

   b. For overdetermined ($M > N$) systems, express $M \times N$ matrix A in the product form $A = QR$, where Q is an $M \times M$ orthogonal matrix and R is an $M \times N$ upper triangular matrix:

$$Q = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_M \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1N} \\ 0 & r_{22} & \cdots & r_{2N} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & r_{NN} \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & & \\ \vdots & & \ddots & \vdots \\ 0 & & \cdots & 0 \end{bmatrix},$$

(*continued on next page*)

where $\mathbf{q}_1$, $\mathbf{q}_2$, … are orthogonal vectors that make up the columns of Q; i.e., $\mathbf{q}_i^T\mathbf{q}_j = \delta_{ij}$., where $\delta_{ij} = 0$ if $i \neq j$ and $\delta_{ij} = 1$ if $i = j$. The $M - N$ rows below the N^{th} row of R are filled with zeroes.

c. Only the first N columns of Q are actually needed. (Why?)
d. To solve a system $A\mathbf{x} = \mathbf{b}$:
   $A\mathbf{x} = \mathbf{b} \rightarrow QR\mathbf{x} = \mathbf{b}$. Let $R\mathbf{x} = \mathbf{z} \rightarrow Q\mathbf{z} = \mathbf{b}$. Solve $Q\mathbf{z} = \mathbf{b}$ for $\mathbf{z}$ and then $R\mathbf{x} = \mathbf{z}$ for $\mathbf{x}$ using backward substitution. Matrix Q is orthogonal, so $Q^{-1} = Q^T$. Thus, $\mathbf{z} = Q^T\mathbf{b}$.
e. Matlab command `[Q R] = qr(A)`
f. Example: Find linear fit to the following data set (seen before):

i	x_i	y_i
1	1.0	1.1
2	2.0	3.2
3	4.0	5.2

For a linear fit: $y = d_0 + d_1 x$, so the "function" matrix F and "data" vector $\mathbf{y}$ are ($M = 3$ and $N = 2$)

$$F = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 4 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 1.1 \\ 3.2 \\ 5.2 \end{bmatrix}.$$

The resulting Q and R matrices are (to four decimal places of accuracy)

$$Q = \begin{bmatrix} -0.5774 & 0.6172 & 0.5345 \\ -0.5774 & 0.1543 & -0.8018 \\ -0.5774 & -0.7715 & 0.2673 \end{bmatrix} \qquad R = \begin{bmatrix} -1.7321 & -4.0415 \\ 0 & -2.1602 \\ 0 & 0 \end{bmatrix}$$

Verify using *Matlab* that Q is orthogonal, then use the two-step solution $\mathbf{z} = Q^T\mathbf{b}$ and $R\mathbf{x} = \mathbf{z}$.

Solution should be $\mathbf{d} = \begin{bmatrix} 0.1000 \\ 1.3143 \end{bmatrix}$

g. As with the LU factorization, A has to be factored only once to solve $A\mathbf{x} = \mathbf{b}$ for any number of different vectors $\mathbf{b}$.

3. Many other kinds of factorizations are available for special situations, such as Cholesky and LDL^T (both for symmetric matrices). The Cholesky and LDL^T factorizations are both available in *Matlab*.

4. For more information on the various factorizations, see

- G. H. Golub and C. F. Van Loan, *Matrix Computations* (4^{th} edition is latest)
- W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing* (3^{rd} edition is latest)