ENGR 4/695 Advanced Topics in Engineering Mathematics Fall 2026

Lecture Outline for Friday, Sept. 25, 2026

1. Wrapping up *QR* and other factorizations

    a. In *QR* factorization, if A is complex, then Q is unitary.
    b. 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*.
    c. 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)

2. The greatest factorization of them all, singular value decomposition (SVD)

    a. Very good at handling difficult systems in which the matrix is very close to singular (ill conditioned), which can happen with large data sets, measurement errors, and/or noisy data, to name just a few issues often encountered in real problems.
    b. Recommended over the normal equation for solving difficult overdetermined systems. Main disadvantages are more memory storage (an extra matrix) and sometimes it is slower.
    c. Can also be used for data compression (demo soon).
    d. Using the so-called "economy-sized" or "thin" decomposition, an $M \times N$ matrix A can be expressed in the product form (assuming $M > N$ or $M = N$ for now)

    $$A = U\Sigma V^H,$$

    where U is an $M \times N$ column-orthogonal matrix, Σ (sometimes labeled S) is an $N \times N$ diagonal matrix, V is an $N \times N$ orthogonal (or unitary) matrix, and H indicates complex conjugate transpose (V can be complex if A is complex; otherwise, V is simply transposed):

    $$U = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_N \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}_{M\times N} \qquad \Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_N \end{bmatrix}_{N\times N} \qquad V = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_N \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}_{N\times N}$$

    where $\mathbf{u}_i^H\mathbf{u}_j = \delta_{ij}$ (orthogonal/unitary) and $\mathbf{v}_i^H\mathbf{v}_j = \delta_{ij}$ (orthogonal/unitary)

    e. In the full SVD, U is $M \times M$ and Σ is $M \times N$, but parts of U and Σ are not necessary for non-square ($M > N$) systems, hence the "economy-sized" decomposition. The full $M \times M$ matrix U is orthogonal.

(*continued on next page*)

f. The diagonal elements of Σ are called *singular values*. They are always real and either positive or zero, even if A has complex entries. They can repeat. Thus,

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \ldots \geq \sigma_N .$$

Zero singular values, if any, occupy the highest index numbers (i.e., up to and including σ_N)

g. The *condition number* of a matrix A is equal to the ratio of the largest to smallest singular values:

$$\text{cond}\left(A\right) = \frac{\sigma_1}{\sigma_N} \quad \text{(infinite for singular matrices)}$$

h. Matlab command (full SVD unless option is added): `[U,S,V] = svd(A)`

3. Implications and properties of SVD and some ways to find U, Σ, and V

a. $A = U\Sigma V^T = \sum_{j=1}^{N} \sigma_j \mathbf{u}_j \mathbf{v}_j^T$, where $\mathbf{u}_j \mathbf{v}_j^T$ is an outer product, each of which is $M \times N$.

   i. The outer products have rank = 1. (See similar HW #2 problem.)
   ii. A is a weighted sum of rank-1 $M \times N$ matrices.
   iii. The weights (σ_j) grow progressively smaller.
   iv. If any of the singular values are zero or too small to matter (lost in the noise, for example), then A can be represented by a subsets of the orthogonal vectors $\{\mathbf{u}_i\}_{i = 1 \text{ to } r}$ and $\{\mathbf{v}_i\}_{i = 1 \text{ to } r}$, where $r < N$.
   v. The summation terms for zero or nearly zero singular values can be ignored. (Example given in image processing demo later.)

b. Use complex conjugate transpose for complex matrix A.

c. U and V are both orthogonal ($U^{-1} = U^T$ and $V^{-1} = V^T$); thus,
$A = U\Sigma V^T \quad \rightarrow \quad AV = U\Sigma \quad \rightarrow \quad A\mathbf{v}_i = \sigma_i \mathbf{u}_i$

d. $A = U\Sigma V^T \quad \rightarrow \quad A^T = V\Sigma^T U^T \quad \rightarrow \quad A^T U = V\Sigma \quad \rightarrow \quad A^T \mathbf{u}_i = \sigma_i \mathbf{v}_i$

e. $A^T A = \left(U\Sigma V^T\right)^T \left(U\Sigma V^T\right) = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T \Sigma V^T$ (an $N \times N$ system)
$\rightarrow \quad \left(A^T A\right)V = V\Sigma^T\Sigma \quad \rightarrow \quad \left(A^T A\right)\mathbf{v}_i = \sigma_i^2 \mathbf{v}_i$ for $i = 1$ to N
σ_i^2 are the eigenvalues of A^TA, and $\{\mathbf{v}_i\}_{i = 1 \text{ to } N}$ are the eigenvectors.
If $N > M$, then at most only M of the N eigenvalues will be nonzero.

f. $AA^T = \left(U\Sigma V^T\right)\left(U\Sigma V^T\right)^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T$ (an $M \times M$ system)
$\rightarrow \quad \left(AA^T\right)U = U\Sigma\Sigma^T \quad \rightarrow \quad \left(AA^T\right)\mathbf{u}_i = \sigma_i^2 \mathbf{u}_i$ for $i = 1$ to M
σ_i^2 are also the eigenvalues of AA^T, and $\{\mathbf{u}_i\}_{i = 1 \text{ to } M}$ are the eigenvectors
If $M > N$, then at most only N of the M eigenvalues will be nonzero.

g. If A is symmetric, then $A^TA = AA^T = A^2$, so $\lambda_i^2 = \sigma_i^2 \quad \rightarrow \quad \left|\lambda_i\right| = \left|\sigma_i\right|$ (sign ambiguity);
$\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i\}$ are the same and are the eigenvectors of A

h. If A is not symmetric, then neither $\{\mathbf{u}_i\}$ nor $\{\mathbf{v}_i\}$ are the eigenvectors of A