

Review Sheet for Exam #1

Exam #1 is scheduled to take place **3:00–5:00 pm on Friday, October 2 in Academic West 204**. The exam will be designed to be completed in about 1 hour even though you will have 2 hours available.

The following policies will be in effect for the exam:

1. You will be allowed to use a non-wireless enabled calculator, such as a TI-99. You must put away smart watches, cell phones, tablets, laptops, and other devices that can connect to the internet. I will record your calculator make and model during the exam.
2. You will be allowed to use up to 10 two-sided paper help sheets. They may include the notes that you recorded in class (original or copied), homework solutions, supplemental notes from the course Moodle site, copied or printed pages from the textbook, or any other printed or handwritten material. Hard copies of the textbook will not be allowed since not every student has access to one. A copy of the linear system solvability flowchart (Fig. 8.3.1 of Zill, 6th ed.) will be provided and does not count toward the 10-sheet limit. If there is anything else that you feel would be helpful, please notify me.
3. If you begin the exam after the start time, you must complete it in the remaining allotted time. However, you will not be allowed to take the exam if you arrive after the first student has completed it and left the room. The latter case is equivalent to missing the exam.
4. **You may not leave the exam room without prior permission except in an emergency or for an urgent medical condition. Please use the restroom before the exam.**

The following is a list of topics that could appear in one form or another on the exam. Not all of these topics will be covered, and it is possible that an exam problem could cover a detail not specifically listed here. However, this list has been made as comprehensive as possible.

Although significant effort has been made to ensure that there are no errors in this review sheet, some might nevertheless appear. The textbook and the course lecture notes and supplemental readings are the final authority in all factual matters, unless errors have been specifically identified there. You are ultimately responsible for obtaining accurate and reliable information when preparing for the exam.

Solution of systems of linear algebraic equations in matrix form

- rank of matrix = max. no. of linearly independent row or column vectors
- max. rank = $\min\{M, N\}$, M = no. rows and N = no. of columns
- rules of linear algebra; interactions between matrices, vectors, and scalars; dimensions
- matrix transpose and its properties
- solution of $A\mathbf{x} = \mathbf{b}$; for an $N \times N$ (square) system, the following statements are equivalent:
 - o $A\mathbf{x} = \mathbf{b}$ has a unique solution
 - o A has a unique inverse (A^{-1})
 - o A is non-singular
 - o A has full rank (i.e., $\text{rank}(A) = N$)
 - o $\det(A) = |A| \neq 0$
- solution using element row operations (EROs) to obtain row echelon form or reduced row echelon form

- consistency
 - o consistent = one solution or infinitely many solutions (2-D analogy: solution space is a point or a line)
 - o inconsistent = no solution (2-D analogy: parallel lines – no intersecting point)
- solvability of $M \times N$ systems (see Fig. 8.3.1 of Zill, 6th ed.)
 - o M = no. of equations; N = no. of unknowns
 - o if $r = \text{rank}(A) = \text{rank}(A|\mathbf{b})$, system is consistent & a solution is possible
 - $r = N$: unique solution; typically, a square system
 - $r < N$: infinitely many solutions; typically, an underdetermined system
 - o if $r = \text{rank}(A) < \text{rank}(A|\mathbf{b})$, system is inconsistent & no solution is possible (but a “best” fit is possible); typically, an overdetermined system

Curve-fitting and the method of least squares

- data set: (x_i, y_i) , $i = 1$ to $M \rightarrow$ data vectors $\mathbf{x}$ and $\mathbf{y}$
- model: a set of functions $\{f_j(x)\}_{j=1}^N$ and coefficients $\{c_j\}_{j=1}^N$ that yield the best approximation to $\{y_i\}_{i=1}^M$ (data set) or $y(x)$ (continuous function):

$$y(x_i) \approx \hat{y}(x_i) = \sum_{j=1}^N c_j f_j(x_i) \rightarrow \hat{\mathbf{y}} = F\mathbf{c}, \text{ where } F_{ij} = f_j(x_i); \hat{\mathbf{y}} \text{ contains best fit;}$$

vector $\mathbf{c}$ contains coefficients
- basis functions $\{f_j(x)\}$ are usually elementary functions like polynomials (including constant, linear, quadratic, and cubic), sin/cos, exponentials, logarithms
- seemingly nonlinear models can sometimes be transformed into linear models via changes of variables and/or algebraic manipulation
- least squares approach
 - o minimize residual vector: $\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}} \rightarrow$ minimize $E = \|\mathbf{F}\mathbf{c} - \mathbf{y}\|^2$ by setting $\partial E / \partial c_k = 0$ for all $k = 1$ to N
 - o make residual vector $\mathbf{r}$ orthogonal to approximation ($\mathbf{r}^T \hat{\mathbf{y}} = 0$)
- basic normal equation

$$F^T F \mathbf{c} = F^T \mathbf{y} \rightarrow \mathbf{c} = (F^T F)^{-1} F^T \mathbf{y},$$
- practical considerations:
 - o In *Matlab*, can write `c = F \backslash y`; automatically forms solution using normal equation (or its equivalent) if $M > N$
 - o $F^T F$ is symmetric and nonsingular if there are no repeated data points
 - o F is $M \times N$, so $F^T F$ is $N \times N$
- constrained LS optimization
 - o many constraints are possible; need to express them in linear algebraic form
 - o one approach: suppress magnitudes of coefficients
 - cost function: $E = \|\mathbf{F}\mathbf{c} - \mathbf{y}\|^2 + \gamma \mathbf{c}^T \mathbf{c}$
 - modified normal equation: $\mathbf{c} = (F^T F + \gamma I_N)^{-1} F^T \mathbf{y}$
 - o another approach: suppress magnitudes of second finite differences of coefficients
 - cost function: $E = \|\mathbf{F}\mathbf{c} - \mathbf{y}\|^2 + \gamma \mathbf{c}^T H \mathbf{c}$

- $H = K^T K$, where $K = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & \\ 0 & 1 & -2 & 1 & & \vdots \\ \vdots & & & \ddots & & \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & 0 & 0 & 0 \end{bmatrix}$
- modified normal equation: $\mathbf{c} = (F^T F + \gamma H)^{-1} F^T \mathbf{y}$

- adjust value of Lagrange multiplier γ until coefficients vary smoothly and have reasonable magnitudes

Eigenvalues and eigenvectors for an $N \times N$ matrix A

- basic form: $A\mathbf{x} = \lambda\mathbf{x}$
- geometric interpretation: For $A\mathbf{x} = \mathbf{b}$, if $\mathbf{x}$ is an eigenvector, then $\mathbf{b}$ is a stretched or shrunk and possibly flipped but not rotated version of $\mathbf{x}$
- computing eigenvalues: $\det(A - \lambda I) = 0$
 - eigenvalues are roots of N th-order polynomial
 - special case: If $N = 2$, then $\lambda^2 - \text{tr}(A)\lambda + \det(A) = 0$, where $\text{tr}(A)$ = trace of A , the sum of the entries on the main diagonal
 - eigenvalues can be repeated, and they can be complex
 - eigenvalues of upper and lower-triangular matrices are the diagonal values
 - eigenvalues of diagonal matrices are the diagonal values
- eigensystem properties
 - if A is square with real entries, then any complex eigenvalues/eigenvectors come in conjugate pairs
 - if A is square, then 0 is an eigenvalue iff (if and only if) A is singular
 - if A is nonsingular and λ is an eigenvalue, then $1/\lambda$ is an eigenvalue of A^{-1} ; both eigenvalues have the same corresponding eigenvectors
 - simple eigensystem has N distinct eigenvalues, N linearly independent eigenvectors
 - $A\mathbf{x}_i = \lambda_i \mathbf{x}_i \rightarrow A\mathbf{X} = \mathbf{X}\Lambda$ (horizontal augmentation of column vectors)
 - symmetric matrices behave well
 - eigenvalues of nonsingular symmetric matrices are nonzero and real
 - eigenvectors for distinct eigenvalues are orthogonal
 - repeated eigenvalues also have linearly independent (LI) eigenvectors (LI $\neq$ orthogonal); however, the eigenvectors *could* be orthogonal; it is always possible to find orthogonal eigenvectors with enough effort (see end of Sec. 8.10)
 - symmetric matrices can be singular and therefore have at least one zero eigenvalue; even so, the properties above still apply
 - orthogonal matrices ($A^{-1} = A^T$, which implies that $A^T A = I$)
 - A is orthogonal iff its columns form an orthonormal set (orthonormal is orthogonal with each vector having a length of 1; i.e., $|\mathbf{x}| = \mathbf{x}^T \mathbf{x} = 1$)
 - orthogonal matrices are not usually symmetric (The only orthogonal *and* symmetric matrix is I because a matrix that is both satisfies $A^{-1} = A^T = A$.)
 - if eigenvectors are orthogonal, a matrix formed with eigenvectors as columns is orthogonal and has the property $X^{-1} = X^T$
 - linearly independent $\neq$ orthogonal

- diagonalization
 - express $N \times N$ matrix A as $A = PDP^{-1}$, where D is a diagonal matrix
 - A does not have to be symmetric or orthogonal to be diagonalizable
 - Theorem: An $N \times N$ matrix A is diagonalizable iff A has N LI eigenvectors
 - Theorem: If $N \times N$ matrix A has N distinct eigenvalues then it is diagonalizable (but it might be diagonalizable even if the eigenvalues are not distinct, that is, if some are repeated)
 - Theorem: An $N \times N$ matrix A can be orthogonally diagonalized iff A is symmetric (*orthogonal diagonalization* means that P is orthogonal)
 - many ways to diagonalize a matrix
 - one important example: Given an $N \times N$ matrix A with N LI eigenvectors (i.e., A is full rank), since $A\mathbf{x}_1 = \lambda_1\mathbf{x}_1$, $A\mathbf{x}_2 = \lambda_2\mathbf{x}_2$, etc., then, using the definitions

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \lambda_N \end{bmatrix} \quad \text{and} \quad X = \begin{bmatrix} \uparrow & \uparrow & \dots & \uparrow \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_N \\ \downarrow & \downarrow & \dots & \downarrow \end{bmatrix},$$

where X is formed by arranging the eigenvectors in adjacent columns, we obtain $AX = X\Lambda \rightarrow A = X\Lambda X^{-1}$. Thus, an $N \times N$ matrix A with LI eigenvectors can be diagonalized into its eigenvalues and eigenvectors.

Some basic factorizations

- LU factorization
 - express $N \times N$ matrix A as $A = LU$, where L is a lower triangular matrix and U is upper triangular
 - 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.
 - LU factorizations are not typically unique; results depend on method used
 - *Matlab* command `[L, U] = lu(A)`
 - determinants: $\det(A) = \det(L) \det(U)$
- QR factorization
 - LU factorization is for square systems; QR factorization is for over- and underdetermined systems ($M > N$ or $M < N$). (It also works for square systems, but it is computational overkill.)
 - 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
 - only the first N columns of Q are actually needed
 - 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, $\mathbf{z} = Q^T\mathbf{b}$.
 - *Matlab* command `[Q, R] = qr(A)`

Singular value decomposition (SVD)

- using “economy-sized” or “thin” decomposition, an $M \times N$ matrix A can be expressed in the product form $A = U\Sigma V^T$, where U is an $M \times N$ column-orthogonal matrix, Σ (sometimes labeled S) is an $N \times N$ diagonal matrix, and V is an $N \times N$ orthogonal matrix:

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

where $\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$ (orthogonal column vectors) and $\mathbf{v}_i^T \mathbf{v}_j = \delta_{ij}$ (orthogonal column vectors), and where $\delta_{ij} = 1$ if $i = j$ and 0 if $i \neq j$

- the above assumes $M > N$ or $M = N$; the $M < N$ case requires a little more care
- in 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 systems, hence the “economy-sized” decomposition
- product form $A = U\Sigma V^T$ is still valid in full SVD
- *Matlab* command (full SVD unless ‘econ’ option is added): `[U, S, V] = svd(A)`
- diagonal elements of Σ are called *singular values*; always real and either positive or zero, even if A has complex entries, and can repeat

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \geq \sigma_N$$

- zero singular values, if any, occupy the highest index numbers (i.e., through σ_N)
- the number of zero singular values = $N - r$, where r = rank of matrix (if $M \geq N$)
- implications and properties:
 - o condition number of a matrix A : $\text{cond}(A) = \sigma_1/\sigma_N$; figure of merit that indicates how close the matrix is to being singular; it could be “numerically singular” because of finite precision of stored numbers in computer
 - o $A = U\Sigma V^T = \sum_{j=1}^N \sigma_j \mathbf{u}_j \mathbf{v}_j^T$, where $\mathbf{u}_j \mathbf{v}_j^T$ is an $M \times N$ outer product; weights are singular values (σ_j) and grow smaller with increasing j . Also, $\text{rank}(\mathbf{u}_j \mathbf{v}_j^T) = 1$.
 - o U and V are both orthogonal ($U^{-1} = U^T$ and $V^{-1} = V^T$); thus,
 - $A = U\Sigma V^T \rightarrow AV = U\Sigma \rightarrow A\mathbf{v}_i = \sigma_i \mathbf{u}_i$
 - $A = U\Sigma V^T \rightarrow A^T = V\Sigma^T U^T \rightarrow A^T U = V\Sigma \rightarrow A^T \mathbf{u}_i = \sigma_i \mathbf{v}_i$
 - $A^T A = (U\Sigma V^T)^T (U\Sigma V^T) = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T \Sigma V^T$
 $\rightarrow (A^T A)V = V\Sigma^T \Sigma \rightarrow (A^T A)\mathbf{v}_i = \sigma_i^2 \mathbf{v}_i$ for $i = 1$ to N
 σ_i^2 are the eigenvalues of $A^T A$, and $\{\mathbf{v}_i\}_{i=1 \text{ to } N}$ are the eigenvectors;
 if $N > M$, the at most only M of the N eigenvalues will be nonzero;
 one way to find singular values and $\mathbf{v}$ eigenvectors
 - $AA^T = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma \Sigma^T U^T$
 $\rightarrow (AA^T)U = U\Sigma \Sigma^T \rightarrow (AA^T)\mathbf{u}_i = \sigma_i^2 \mathbf{u}_i$ 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 $M > N$, the at most only N of the M eigenvalues will be nonzero;
 one way to find singular values and $\mathbf{u}$ eigenvectors
- if A is symmetric, then $A^T A = AA^T = A^2$, so $\lambda_i^2 = \sigma_i^2 \rightarrow |\lambda_i| = |\sigma_i|$ (sign ambiguity)

- to solve a system $A\mathbf{x} = \mathbf{b}$ (for overdetermined and square systems; underdetermined requires interpretation):

$$U\Sigma V^T \mathbf{x} = \mathbf{b} \rightarrow \Sigma V^T \mathbf{x} = U^T \mathbf{b} \rightarrow V^T \mathbf{x} = \Sigma^{-1} U^T \mathbf{b} \rightarrow \mathbf{x} = V \Sigma^{-1} U^T \mathbf{b}$$

equivalent to

$$\mathbf{x} = \sum_{j=1}^N \left(\frac{\mathbf{u}_j^T \mathbf{b}}{\sigma_j} \right) \mathbf{v}_j$$

- for overdetermined curve-fitting
 - o $\mathbf{c} = V \Sigma^{-1} U^T \mathbf{y}$ yields same coefficients as unconstrained LS if all singular values are used
 - o equivalent to

$$\mathbf{c} = \sum_{j=1}^N \left(\frac{\mathbf{u}_j^T \mathbf{y}}{\sigma_j} \right) \mathbf{v}_j$$
 - o to smooth oscillating coefficient values, set entries in Σ^{-1} matrix corresponding to small singular values (i.e., below a certain threshold) to zero; reduces impact of eigenvectors that do not model data vector $\mathbf{y}$ very well

Relevant course material:

Homework: #1, #2, and #3

Labs: #1 and #2

Reading: Assignments from Aug. 24 through Sept. 25, including the supplemental readings:
 “Constrained Least-Squares Optimization Using Minimized Coefficient Magnitudes”
 Mathworks Help Center documentation on factorizations
 SVD readings from *Numerical Recipes in C*, 2nd ed.
 “The Fundamental Theorem of Linear Algebra” (optional)

This exam will focus primarily on the course outcomes listed below, although only the parts of the third outcome relevant to linear algebra will be covered.

- Review basic concepts of linear algebra
- Apply linear-algebraic methods to the solution of applied problems
- Develop the concept of an eigenvalue problem in both linear algebra and differential equations

The course outcomes are listed on the course syllabus, which was distributed at the beginning of the semester and is available on the Syllabus and Policies page at the course web site. The outcomes are also listed on the Course Description page at the web site. Note, however, that some topics not directly related to the course outcomes could be covered on the exam as well.