MATH 240: Introduction to Linear Algebra

Summer Session I – 2026
Instructor: TBD
Credits: 3
Contact Hours: 45 (9 hours/week, e.g., 3 days/week, 3 hours/day)
Prerequisites: Calculus I or equivalent
Class Meeting Days & Time: TBD (e.g., M/W/F, 9:00 AM–12:00 PM)
Office Hours: By appointment after class or via Zoom (see Moodle)
Email: TBD
Course Type: Standard Course
Course Fee: TBD with Institute staff

Course Description

This course introduces the fundamental concepts of linear algebra, emphasizing both theoretical foundations and practical applications. Topics include systems of linear equations, matrix algebra, vector spaces, linear transformations, determinants, eigenvalues, eigenvectors, inner products, orthogonality, and least squares solutions. Applications in data science, engineering, and modeling (e.g., Markov chains, data fitting) are integrated to connect theory to real-world problems. The course is designed for students in mathematics, engineering, computer science, and related fields.

Learning Outcomes and Assessment Measures

By the end of this course, students will be able to:

- Solve systems of linear equations using Gaussian elimination and matrix methods (Quizzes, Problem Sets, Midterm, Final).
- Perform matrix operations and compute determinants and inverses (Quizzes, Problem Sets, Midterm, Final).
- Understand and apply concepts of vector spaces, linear transformations, and bases (Problem Sets, Midterm, Final).
- Compute eigenvalues and eigenvectors and apply them to problems like diagonalization (Problem Sets, Midterm, Final).
- Use inner products, orthogonality, and least squares to solve applied problems (Computational Labs, Final).
- Connect linear algebra concepts to applications in data science and engineering (Computational Labs, Problem Sets).

Course Materials

Textbook

Lay, D. C., Lay, S. R., & McDonald, J. J. (2022). *Linear Algebra and Its Applications* (6th ed.). Pearson. (Available via bookstore or online).

Additional Resources

- A course reader with supplementary notes and practice problems will be provided on Moodle.
- Software: MATLAB or Python (NumPy) for computational labs, accessible via institutional licenses or free platforms.
- Moodle: Primary location for readings, assignments, and announcements.

Grading

Students are reminded that it is their responsibility to note the dates of exams and other assignments. No alternative exam dates will be offered and professors are not required to give partial credit for any late work (they do so at their discretion: the Institute's default policy is no extensions and a zero for any work turned in late). Students who book travel when they have an exam or other assessment will have to change their plans or accept a zero. Letter grades for student work are based on the following percentage scale:

Letter Grade Range	Numerical Score Equivalent	Student Performance
A A-	93% - 100% 90% - 92%	Exceptional Excellent
B+ B B-	87% -89% 83% - 86% 80% - 82%	Superior
C+ C C-	77% - 79% 73% - 76% 70% - 72%	Satisfactory
D+ D D-	67% - 69% 63% - 66% 60% - 62%	Low Pass
F	59% or less	Fail (no credit)

Please note: decimal numerals between 1-4 are rounded down while 5-9 are rounded up: e.g., expect 89.4 to be 89.0 while 89.5 to round up to 90.

Course Requirements

Attendance (10%): One excused "sick day" allowed. Each unexcused absence reduces the final grade by 4% (up to 10% total). No make-ups for attendance.

Quizzes (10%): Weekly quizzes test foundational skills (e.g., row reduction, matrix inverses). Administered in class.

Problem Sets (25%): Weekly assignments include theoretical proofs and applied problems (e.g., Markov chains). Submitted via Moodle.

Computational Labs (15%): Weekly labs involve coding (MATLAB or Python) for applications like least squares or eigenvalue computations. Submitted via Moodle.

Midterm Quiz (20%): Covers Weeks 1–2 (systems, matrices, vector spaces). In-class, Week 3.

Final Exam (20%): Comprehensive, covering all topics. In-class, Week 5.

Late Work: Work submitted after deadlines receives a zero, except for one 24-hour extension (email instructor before deadline). No extensions for the final exam.

Extension & Submitting Late Work
Umbra Academic Policies apply

Attendance Policy
Umbra Academic Policies apply

Tardiness Policy
Umbra Academic Policies apply

Academic Integrity
Umbra Academic Policies apply

Classroom Policy
Umbra Academic Policies apply

Moodle
Please note that Moodle, not this syllabus, is the ultimate reference for due dates, assignment prompts, and course announcements. It is *the student's responsibility* to check the site regularly to be aware of announcements as well as to see and record all due dates for assignments.

Schedule of Topics, Readings, and Assignments

(Each meeting is two hours with break)

Week 1: Systems of Linear Equations and Matrices

- Meeting 1: Lecture and examples on introduction to linear systems, Gaussian elimination.
- Meeting 2: Lecture, examples, and MATLAB/Python assistance on row operations, reduced row echelon form (assistance: setup, matrix input).
- Meeting 3: Lecture and examples on matrix operations (addition, scalar multiplication, multiplication).
- Meeting 4: Lecture, examples, MATLAB/Python assistance on matrix inverses, solving systems with matrices (assistance: matrix operations), Quiz 1.
- Readings: Lay et al., *Linear Algebra and Its Applications* (6th ed.), Sections 1.1–1.5, 2.1–2.2.
- Assignments:
  - Problem Set 1 (due end of Week 1): Solve linear systems using Gaussian elimination (e.g., 3x3 systems), compute matrix products, find inverses by row reduction.
  - Computational Lab 1 (due end of Week 1): Use MATLAB/Python to perform matrix operations (e.g., multiplication, inverses with np.linalg.inv). Start in assistance sessions (Meetings 2, 4), complete outside. Submit .m or .py file via Moodle.

Week 2: Vector Spaces and Linear Transformations

- Meeting 1: Lecture and examples on vectors in $\mathbb{R}^n$, linear combinations, span.

- Meeting 2: Lecture, examples, and MATLAB/Python on linear independence, bases (assistance: vector operations).
- Meeting 3: Lecture and examples on vector spaces, subspaces.
- Meeting 4: Lecture, examples, MATLAB/Python on linear transformations, matrix representations (assistance: transformation matrices), Quiz 2.
- Readings: Lay et al., Sections 4.1–4.4, 5.1–5.2.
- Assignments:
  - Problem Set 2 (due end of Week 2): Determine linear independence of vectors, find bases for subspaces, compute matrix representations of transformations.
  - Computational Lab 2 (due end of Week 2): Use MATLAB/Python to check linear independence (e.g., rank with np.linalg.matrix_rank), visualize transformations. Start in assistance sessions (Meetings 2, 4), complete outside. Submit .m or .py file via Moodle.

Week 3: Determinants and Inverses

- Meeting 1: Lecture and examples on determinants: definition, properties.
- Meeting 2: Lecture, examples, and MATLAB/Python on computing determinants, Cramer's rule (assistance: determinant computation).
- Meeting 3: Lecture and examples on matrix inverses: methods and applications.
- Meeting 4: Midterm Quiz (covers Weeks 1–2), MATLAB/Python (review determinants), review.
- Readings: Lay et al., Sections 3.1–3.3, 2.3.
- Assignments:
  - Problem Set 3 (due end of Week 3): Compute determinants (2x2, 3x3), solve systems using Cramer's rule, derive inverse formulas.
  - Computational Lab 3 (due end of Week 3): Use MATLAB/Python to compute determinants (e.g., np.linalg.det), verify inverses. Start in assistance sessions (Meetings 2, 4), complete outside. Submit .m or .py file via Moodle.

Week 4: Eigenvalues and Eigenvectors

- Meeting 1: Lecture and examples on eigenvalues, eigenvectors: definitions.
- Meeting 2: Lecture, examples, and MATLAB/Python on characteristic polynomial, computing eigenvalues (assistance: eigenvalue computation).
- Meeting 3: Lecture and examples on diagonalization of matrices.
- Meeting 4: Lecture, examples, MATLAB/Python on applications (e.g., Markov chains) (assistance: Markov chain simulation), Quiz 3.
- Readings: Lay et al., Sections 5.3, 5.5, 7.1.
- Assignments:
  - Problem Set 4 (due end of Week 4): Find eigenvalues/eigenvectors, diagonalize matrices, analyze Markov chain transitions.
  - Computational Lab 4 (due end of Week 4): Use MATLAB/Python to compute eigenvalues (e.g., np.linalg.eig), simulate Markov chains. Start in assistance sessions (Meetings 2, 4), complete outside. Submit .m or .py file via Moodle.

Week 5: Inner Products and Orthogonality

- Meeting 1: Lecture and examples on inner products, norms.
- Meeting 2: Lecture, examples, and MATLAB/Python on orthogonality, Gram-Schmidt process (assistance: Gram-Schmidt code).
- Meeting 3: Lecture, examples, and MATLAB/Python on least squares, applications (e.g., data fitting) (assistance: least squares problems).
- Meeting 4: Final Exam (comprehensive, covering all topics), MATLAB/Python assistance

(review least squares code), review (finalize lab submission).

- Readings: Lay et al., Sections 6.1–6.3, 6.7.
- Assignments:
  - Problem Set 5 (due end of Week 5): Compute inner products, apply Gram-Schmidt process, solve least-squares problems.
  - Computational Lab 5 (due end of Week 5): Use MATLAB/Python for least squares (e.g., np.linalg.lstsq), plot data fits. Start in assistance sessions (Meetings 2, 3), complete outside. Submit .m or .py file via Moodle.