PMAI LAB

Linear Algebra: Theory, Computation, and Modern Applications

Course Outline

Course Duration
10 Lectures

Target Audience
Undergraduate BS Mathematics students

Computational Tools
Python • NumPy • Matplotlib

Instructor
Dr. Hira Benish

Mathematical foundations connected with computation and modern applications.

1. Course Overview

A balanced, application-oriented introduction for mathematics students

Course description

This application-oriented course provides undergraduate mathematics students with a balanced understanding of fundamental linear algebra concepts and their modern applications. It combines essential mathematical theory with problem solving, interpretation, and Python computation.

Applications are drawn from computer science, artificial intelligence, data science, engineering, population modelling, environmental analysis, finance, and other scientific fields. Emphasis is placed on understanding and applying concepts rather than lengthy derivations.

> **Course philosophy**
> Students learn each topic through the sequence: concept, mathematical formulation, worked example, application problem, Python computation, and interpretation.

Course learning outcomes

After completing this course, students should be able to:

1. Model real-world problems using vectors, matrices, and systems of linear equations.
2. Analyze vector spaces using span, linear independence, basis, dimension, and rank.
3. Apply matrices and linear transformations to mathematical and computational problems.
4. Use inner products, orthogonality, projections, and least squares for approximation.
5. Interpret eigenvalues, eigenvectors, diagonalization, SVD, and PCA.
6. Solve application-based linear algebra problems using Python.
7. Evaluate results through residuals, reconstruction errors, stability, and contextual interpretation.

Course emphasis

- Conceptual understanding with appropriate mathematical depth
- Fully worked numerical examples
- Application-based problem solving
- Python implementation for computational topics
- Interpretation, verification, and responsible use of results

> **Expected progression**
> Students move from exact systems and vector-space structure to approximation, dynamics, dimensionality reduction, and integrated data-driven applications.

2. Lecture-Wise Course Outline

Lectures 1–5: mathematical representation, structure, and transformation

Lecture 1: Linear Systems and Mathematical Modelling
Formulating real-world constraints, consistency and solution types, resource-planning problems, and solving systems with Python.

Lecture 2: Matrices, Determinants, and Inverses
Matrix operations, determinants, invertibility, solving systems, and transformation and computational applications.

Lecture 3: Vectors, Linear Combinations, Span, and Subspaces
Vector representation, reachable solution spaces, data representation, computer graphics, and scientific modelling.

Lecture 4: Linear Independence, Basis, Dimension, and Rank
Detecting redundancy, identifying essential information, feature selection, and data-dimension analysis.

Lecture 5: Linear Transformations and Matrix Representations
Transformations of vectors, kernel and image, geometric transformations, computer graphics, and data mappings.

Learning progression in Lectures 1–5

- Translate applied situations into equations, vectors, and matrices.
- Recognize consistency, independence, redundancy, basis, and rank.
- Interpret a matrix as both a computational object and a transformation.
- Use Python to solve, verify, and visualize small and medium problems.

Foundation developed
By the end of Lecture 5, students can construct and analyze the matrix representation of a problem before selecting an advanced approximation or decomposition method.

3. Lecture-Wise Course Outline

Lectures 6–10: approximation, dynamics, reduction, and integration

Lecture 6: Inner Products, Orthogonality, and Projections
Vector similarity, orthogonal bases, closest-point problems, recommendation systems, and signal approximation.

Lecture 7: Least Squares and Linear Regression
Inconsistent systems, normal equations, residuals, regression metrics, and energy-demand prediction using Python.

Lecture 8: Eigenvalues, Eigenvectors, and Diagonalization
Matrix powers, dynamic systems, stability, population-transition models, and long-term behaviour.

Lecture 9: Singular Value Decomposition, PCA, and Data Compression
Low-rank approximation, pseudoinverse, image compression, dimensionality reduction, and recommendation systems.

Lecture 10: Integrated Applications and Course Review
Smart-city environmental monitoring, sensor-data analysis, integrated regression and PCA, and a smart-agriculture capstone.

Teaching approach

- Conceptual explanation followed by mathematical formulation
- Fully worked numerical examples and application-based case studies
- Python implementation for every major computational topic
- Interpretation of results, practice exercises, MCQs, and scenario-based problems

Application areas

Computer science, artificial intelligence, machine learning, data science, computer graphics, network analysis, engineering, finance, environmental modelling, population dynamics, image processing, signal processing, and scientific research.

Course outcome
Students complete the course with a connected toolkit for modelling, solving, approximating, reducing, computing, and interpreting linear problems across modern fields.