# DATA 374-A: Computational Linear Algebra

**Course Syllabus**

**Description:** *The course develops the idea of linear algebra in a computational setting. Students will gain experience with methods performing QR or Singular Value Decomposition and other matrix algorithms on large datasets. The idea of sparse vs. dense matrices and approximation will be explored. Other advanced topics related to data mining may be covered. This course will be taught primarily in Python or MATLAB.*

**Credit Hours:** 3 credits

**Audience:** Advanced science or mathematics students will benefit. Primarily for students interested in field with advanced computational applications such as bioinformatics, data science, meteorology, computational physics, etc.

**Prerequisite:** (MATH 260 or MATH 264) and (CS 157 or DATA 151)

**Format:** 2 lectures (50 min) and 1 computer lab (75 min) per week

**Textbook:** *Linear Algebra and Learning from Data* by Gilbert Strang (ISBN 978-0-692-19638-0) (Required)

**Technology:** Use of MATLAB and Python is required (available on campus computers)

**Internet:** Course material and grades are often maintained in Blackboard, at the discretion of the instructor.

**Course Goals:**

A. Students will reinforce and apply knowledge from linear algebra, such as matrix operations, matrix decomposition, and finding eigenvalues.
B. Students will understand how fundamental concepts in linear algebra are implemented and executed computationally.
C. Students can apply and perform both basic and advanced algorithmic/computational tasks within linear algebra.
D. Students can develop and improve analytics thinking for problem formation and solution validation, especially using technology.
E. Students prepare for success in fields with advanced computation, such as bioinformatics, computational physics, astrophysics, meteorology, data science, etc.

**Topical Objectives** (with goals addressed).

Preface: *Students will be able to…*

1. Understand and be able to perform singular value decomposition (SVD) and other common matrix factorizations (e.g. LU, QR, Cholesky, etc). (A,C)

2. Find eigenvalues/eigenvectors computationally using exact and heuristic algorithms. (A,B,C)
3. Computationally use the Gram-Schmidt, and other, orthonormalization processes (A,B,C)
4. Understand the difference between dense and sparse matrices, including appropriate algorithms for key computational on each. (A,B,C,D)
5. Understand how MATLAB, and other programming languages, implement and store matrices along with corresponding performance implications. (B,E)
6. Distinguish problems based on computability, including the impact of sparse vs. dense matrix operations (C,D,E)
7. Use the Linear Algebra Package (LAPACK) library of subroutines (and its derivatives) (B,C,E)
8. Apply skills from objectives 1, 2, and 3 to large matrices (D,E)
9. Apply decomposition and eigen-based methods to image processing, data mining, advanced simulations, or other application areas (D,E)

**General Objectives** (with goals addressed).

Preface: *Students will be able to …*

10. **CIS/AMOD:** have a working knowledge of the theoretical foundations of the discipline (A,B)
11. **CIS/AMOD:** demonstrate expertise in the development and design of software (B,C)
12. **CIS/AMOD:** demonstrate the ability to communicate computer science and related topics in written and oral formats (D,E)
13. **MATH/AMOD:** demonstrate the ability to solve advanced mathematical problems and communicate solutions in writing (A,D,E)
14. **MATH/AMOD:** demonstrate the ability to comprehend advanced mathematics (A,C,D)
15. **MATH/AMOD:** be aware of mathematics outside the classroom (E)