Math 156, Spring 2025
Numerical Analysis for Data Science and Statistics

Instructor: Prof. Jon Wilkening
Office: 1051 Evans Hall
Office Hours: Mon 3:30-4:30 PM, Wed 9:45-11:00 AM
e-mail: wilkening@berkeley.edu (emergencies & administration only. No questions about HW, please)
online discussion forum for our class: edstem.org , GSI: Tom Schang, tom_schang@berkeley.edu
Course Announcements, Homework Solutions, etc.: https://bcourses.berkeley.edu/

Lectures: MWF 2:10-3:00 PM, Cory 241

Required textbook: "Linear Algebra and Learning from Data," by Gilbert Strang.
Recommended reading: "Accuracy and Stability of Numerical Algorithms," by Nicholas J. Higham
"Applied Numerical Linear Algebra," James W. Demmel
"Numerical Optimization," Jorge Nocedal and Stephen J. Wright

Prerequisites: Multivariable Calculus (Math 53) and Linear Algebra (Math 54 or 56)

Syllabus: Introduction to applied linear algebra and optimization with applications in data science. We will cover Parts I, II, V, VI (and a few subtopics of III, VII) of Strang's book, as well as Higham (chap. 2-3), Demmel (chap. 2), Nocedal/Wright (chap. 2-6). In more detail:

- Floating-point arithmetic, relative and absolute error, running error analysis
- Vector spaces (real/complex), dimension, subspaces, linear operators, rank-nullity theorem
- Gaussian elimination, LU factorization, positive definite matrices, Cholesky factorization
- Inner products, adjoint operator, orthogonal projections, closest point property
- Orthogonal and unitary matrices, QR via Gram-Schmidt or Householder, LAPACK implementation
- Eigenvalues and eigenvectors, singular value decomposition (SVD)
- Least squares via QR or the SVD, pseudo inverse, polar decomposition
- Perturbation theory, condition number, backward stability analysis
- Low rank approximation, Eckart-Young theorem (2-norm, Frobenius norm)
- Discrete probability, sample mean and covariance, principal component analysis
- Advanced calculus review (Jacobian, gradient, Hessian), multivariable Newton's method
- Nonlinear least squares, trust-region framework, Levenberg-Marquardt method
- Line-search methods, Wolfe conditions, conjugate gradients, GMRES, BFGS
- Rank-revealing QR, sparse solutions, matrix completion, intro to deep neural networks

Grades: Programming assignments: 20% (tentatively due Feb 15, Mar 8, Apr 5, Apr 26)
Homework: 20%. 12 assignments, 2 lowest scores dropped
Midterms: 15% each (in class, Feb 26 and April 9)
Final: 30% or 45% (Tuesday, May 13, 11:30-2:30. Can replace midterm score with the score on the final if helpful.) *No make-up exams for any reason... don't miss the final exam!*

Grade cutoffs: 98 A+, 90 A, 86 A–, 82 B+, 78 B, 74 B–, 70 C+, 66 C, 62 C–, 58 D+, 50 D (no D- given)
(raw scores on exams will be mapped to scaled scores, keeping these cutoffs in mind. The scaled score will never be lower than your raw score expressed as a percentage)

Incomplete grades: (University policy) The grade "I" will only be given if "your work in a course has been of passing quality but is incomplete for reasons beyond your control"

More details: 12 homework assignments, 4 programming assignments. You may discuss the homework and programming assignments with your classmates, but **you must write up your own solutions.** The two lowest homework scores will be dropped. Python will be the official programming language for the course (submitted to gradescope via a Jupyter notebook.)