MATH 350: Numerical Analysis I

Discuss: We could also call this “Discrete Numerical Analysis”. If we call things Numerical Analysis I/II, then the university requires that the second course has the first as a dependency. But in reality, the second course only uses the solution of linear systems from the first, and students can treat this as a black box algorithm. If we used a different naming scheme, say “Discrete Numerical Analysis” and “Continuous Numerical Analysis”, then we could omit the prerequisite.

Course Topics *Numerical methods* are the foundation for all approaches by which we *solve mathematical or engineering problems on computers.* They are typically formulated in terms of algorithms that can be implemented on computers, and include things like solving linear systems or nonlinear equations, but also how to solve ordinary or partial differential equations such as those in fluid dynamics, elasticity, and many other areas.

This course provides an introduction to the basic techniques in this area, covering specifically “discrete problems” such as the solution of linear and nonlinear systems, matrix decompositions, and eigenvalue problems. These techniques will re-appear as the building blocks of more complicated algorithms, such as those we consider in Numerical Analysis II (MATH 450).

In contrast to a course on numerical *methods*, this class on *numerical analysis* is concerned not only with devising methods, but also analyzing their performance. For example, we will consider how *expensive* particular algorithms are, so that we can understand which ones can be implemented efficiently. We will also consider how *accurate* algorithms are – this is important because in most cases, algorithms for non-trivial problems can only give us *approximate* solutions, and we need to know how close these approximations are to the exact solution.

This course will cover the following, specific topics:

1. **Solving linear systems:** Nearly every problem in computational mathematics at some point or other is reduced to solving linear systems $Ax = b$ where A is a known matrix, b is a known vector, and you are looking for the vector x. We like to reduce all of these other problems to solving linear systems because we know very well how to do this. This part of the class walks you through the common methods, starting with the Gaussian elimination approach that you will have used for small matrices in high school already. But Gaussian elimination is too expensive for large problems, and you will see alternative approaches such as LU, Cholesky, and QR decompositions; fixed point iterations; and Krylov space methods. All of these exploit specific properties of the matrix A in making the solution of $Ax = b$ more efficient than the simple Gaussian elimination method. Some of these have been used to solve linear systems with billions or trillions of equations.

2. **Computing eigenvalues and singular values for square and rectangular matrices:** Whenever you are solving a linear system like $Ax = b$, the matrix A often represents something about the system you are modeling. Properties of A therefore often tell you something about the properties of the underlying system. Specifically, we are often interested in the eigenvalues or singular values of A – for example, when fitting a line to data, singular values indicate how *sensitive* the fit is to the data. Similarly, in the widely used *principal component analysis (PCA)*, the principal directions can be understood as the eigenvectors corresponding to the largest eigenvalues of a matrix $A = X^T X$ or alternatively as the singular vectors corresponding to the largest singular values of a matrix X.

3. **Solving nonlinear systems:** Linear systems appear wherever the response of a system is proportional to the input – for example if you pull on a spring, the lengthening of the spring is proportional to the force you apply (this is “Hooke’s law”). But many effects in real life are not proportional, and consequently models are not linear. We are then left with finding solutions x to equations of the form $F(x) = 0$ where F is a nonlinear function. For problems of one variable, one can use the “bisection method”, but this method does not generalize to problems where x and F are vectors. For such problems, you will see methods that repeatedly solve linear systems $A_k x_k = b_k$, $k = 1, 2, \ldots$ in such a

way that x_k comes closer and closer to the solution of $F(x) = 0$. The methods we will discuss are of "fixed point" nature; specifically, we will discuss "Newton's method for finding roots".

4. **Nonlinear optimization:** We solve problems on computers because we either want to *understand* things (this often leads to the data science applications in part 2) or because we want to *optimize*. In nonlinear optimizing, we are seeking that x that minimizes a function $f(x)$. We will rewrite this problem into one finding a root of the equation $f'(x) = 0$, which can then be solved using Newton's method of the previous section. We will also consider the steepest descent method, as well as how these methods need to be modified ("globalized") to solve practical problems.

In concrete terms, the following is an approximate weekly schedule:

- **Week 1:** Examples of applications of numerical methods. Algorithms; finite vs. iterative algorithms. Floating point mathematics, errors, norms, convergence and convergence rates. Complexity classes. Big-O notation.
- **Week 2:** Linear systems and their solution: Gaussian elimination, the algorithm and its complexity. Computing the inverse of a matrix.
- **Week 3:** Linear systems and their solution: LU decomposition and its use to solve linear systems. Cholesky decomposition. Complexity of computing decompositions.
- **Week 4:** Least-squares problems, and the use of the QR decomposition. Application to fitting low-dimensional models to large data sets.
- **Week 5:** Iterative solution of linear systems: Fixed point methods and their analysis.
- **Week 6:** Iterative solution of linear systems: Krylov subspace methods.
- **Week 7:** Computing eigenvalues of square matrices.
- **Week 8:** The singular value decomposition (SVD) of a matrix and its computation.
- **Week 9:** Application of the SVD to data science applications; principal component analysis. Low-rank approximations.
- **Week 10:** Solving a nonlinear equation in one variable: Applications, the bisection method.
- **Week 11:** Solving systems of nonlinear equations: Fixed point methods and their analysis.
- **Week 12:** Solving systems of nonlinear equations: Newton's method and its analysis.
- **Week 13:** Nonlinear optimization: The steepest descent and its limitations. Introduction to Newton's method.
- **Week 14:** Nonlinear optimization: Globalization of Newton's method via line search.
- **Week 15:** Student project presentations.

Textbook I do not require you to get any particular book, and in particular will not pose homework that references a book. That said, if you want to read up on some of the material we discuss in class, the following two books (in any edition you can find) cover essentially everything we do over the course of this semester:

D. Kincaid and W. Cheney: *Numerical Analysis*, Brooks & Cole Publishing Co.

Prerequisites

The following are the current prereqs for MATH 450. We should re-consider whether these are what we want.

(CS 150A or CS 150B or CS 152 or CS 163 or CS 164 or CS 165 or CS 253 or MATH 151) and (MATH 255 or MATH 261).

Many of the homework assignments will require you to write small programs – say, in the range of 20–100 lines of code. In general, I leave the choice of programming language to you, but if your choice is somewhat exotic or outside the realm of what a typical programmer can be expected to read, you will need to provide sufficient commentary to make the code understandable.

Webpage Homework assignments and other course information will be posted on Canvas.

Exams + Grading Final grades will be determined based on the following components:

- Homework and programming assignments: 50%
- Midterm: 20%
- Final project: 30%

Your minimum grade will be A, B, C, or D, for a score of 90%, 80%, 70%, and 60% over the course of the semester, respectively.

You must make arrangements in advance if you expect to miss an exam or quiz. Exam absences due to recognized University-related activities, religious holidays, verifiable illness, and family/medical emergencies will be dealt with on an individual basis. In all cases of absence from exams a written excuse is required. Ignorance of the time and place of an exam will not be accepted as an excuse for absence.

Learning Outcomes and Course Objectives Numerical methods are the foundation of computer simulations in all fields of the sciences and engineering. But they are also important for solving all sorts of equations you find in mathematical biology, chemistry, physics, or finance: Whenever you end up with an equation that cannot be solved on a piece of paper (because you can't find the right approach to solving it, because there just isn't a way to solve the equation on a piece of paper, or because you find yourself not in the mood to solve the problem because it would take too long), then you need a computer to do it; this class teaches you how computers do that.

The goal of this class is to (i) provide a basic level of literacy in numerical methods, as well as (ii) to learn about their analysis. At the end of the semester, you will be able to identify and understand what methods to use depending on the situation; how they will likely perform; and analyze these methods in terms of properties such as approximation quality or speed of convergence. You will also have practice in implementing these methods on computers.

Policies *Academic integrity:* Academic integrity is integral to the success of the University and to you as a learner. Academic integrity is conceptualized as doing and taking credit for one's own work. Academic dishonesty undermines the educational experience at Colorado State University. Examples of academic dishonesty include (but are not limited to) cheating, plagiarism, and falsification. Plagiarism includes the copying of language, structure, images, ideas or thoughts of others and is related only to work submitted for credit. Cheating or any form of academic dishonesty will not be tolerated. The use of material from improperly cited or credited sources will be considered plagiarism. You are encouraged to collaborate with your classmates, unless otherwise directed, but all work intended for a grade must clearly be your work as an individual. Ignorance of the rules does not exclude any member of the CSU community from the requirements or the processes meant to ensure academic integrity.

Disabilities: Colorado State University, in compliance with state and federal laws and regulations, does not discriminate on the basis of disability in administration of its education related programs and activities. We have an institutional commitment to provide equal educational opportunities for disabled students who are otherwise qualified. Students with documented disabilities must contact The Office of Resources for Disabled Students (RDS; 970-491-6385) to make arrangements for class accommodations. It is the responsibility of the student to obtain accommodation letters from RDS and to make arrangements for the implementation of accommodations with faculty in advance. Students who believe they have been denied access to services or accommodations required by law should contact RDS (970-491-6385). Students who believe they have been subjected to discrimination on the basis of disability should contact the Office of Equal Opportunity (970-491-5836). For more information regarding disability grievance procedures, visit `http://oeo.colostate.edu`.