Applied Mathematics 205
Advanced Scientific Computing: Numerical Methods

Lecturer: David Knezevic

Time: Tuesday, Thursday 10am

Course Description

Scientific computing has become an indispensable tool in many branches of research, and is vitally important for studying a wide range of physical and social phenomena. In this course we will examine the mathematical foundations of well-established numerical algorithms and explore their use through practical examples drawn from a range of scientific and engineering disciplines.

There will be an emphasis on mathematical theory and numerical analysis to ensure that students understand the concepts that underpin each algorithm that we consider. There will also be a significant programming component in the course. Students will be expected to implement a range of numerical methods in homework assignments to get hands-on experience with modern scientific computing. In-class demos will be performed with Matlab, and the homework assignments can be completed with either Matlab, Python or Julia.

Learning Objectives

After taking this course, students should be able to:

- Apply standard techniques to analyze key properties of numerical algorithms such as stability and convergence
- Understand and analyze common pitfalls in numerical computing such as ill-conditioning and instability
- Perform data analysis efficiently and accurately using data fitting methods
- Derive and analyze numerical methods for ODEs and PDEs
- Perform optimization using well-established algorithms
- Implement a range of numerical algorithms efficiently in a modern scientific computing programming language

Prerequisites

Course material will assume familiarity with linear algebra and calculus, and students are expected to have basic programming experience (e.g., CS50).

Intended Audience

This course is aimed at students who will employ numerical algorithms in their research. This generally includes students from a wide range of disciplines including life sciences, physical sciences, the humanities, engineering and applied mathematics.

Course Content

The course content will be divided into five main Units, each consisting of a set of Chapters. There will be an assessed homework assignment associated with each Unit.

Unit 0: Overview of Scientific Computing

Unit I: Data Fitting

Chapter I.1: Motivation
Chapter I.2: Polynomial interpolation
Chapter I.3: Linear least squares fitting
Chapter I.4: Nonlinear least squares

Unit II: Numerical Linear Algebra

Chapter II.1: Motivation
Chapter II.2: LU and Cholesky factorizations
Chapter II.3: QR factorization, SVD

Unit III: Numerical Calculus and Differential Equations

Chapter III.1: Motivation
Chapter III.2: Numerical differentiation, numerical integration
Chapter III.3: ODEs, forward/backward Euler, Runge-Kutta schemes
Chapter III.4: Lax equivalence theorem, stability regions for ODE solvers
Chapter III.5: Boundary value problems, PDEs, finite difference method

Unit IV: Nonlinear Equations and Optimization

Chapter IV.1: Motivation
Chapter IV.2: Root finding, univariate and multivariate cases
Chapter IV.3: Necessary conditions for optimality
Chapter IV.4: Survey of optimization algorithms

Unit V: Eigenvalue Problems

Chapter V.1: Motivation
Chapter V.2: QR algorithm
Chapter V.3: Power method, inverse iteration
Chapter V.4: Lanczos algorithm, Arnoldi algorithm

Final Project

- Projects will be completed in groups of 2 or 3 students
- Each group will propose a project topic drawn from an application area of of interest. The project should make use of concepts covered in the course.
- The project should be roughly equivalent in scope to a section of a published research article.

- You will be required to write software to solve your problem, and to submit a report that includes a mathematical discussion of your methodology in relation to the theory covered in the course.
- Projects will be assessed based on a written report, and the quality and correctness of software. Code should be well-documented and should be organized so that figures submitted in the report can be easily reproduced by the graders.
- Further guidance on project topics can be obtained by scheduling a meeting with one of the TFs during the latter part of the semester.

Assessment

- 5 homework assignments: 60%
- Mid-term exam (take home exam): 10%
- Final project: 30%

Required Textbooks

None

Recommended Textbooks

M. Heath. Scientific Computing: An Introductory Survey. McGraw Hill, 2002.

A. Greenbaum and T. P. Chartier. Numerical Methods: Design, Analysis and Computer Implementation of Algorithms. Princeton University Press, 2012.

D. J. Higham and N. J. Higham. MATLAB guide. SIAM, 2005.

C. Moler. Numerical Computing with MATLAB. SIAM, 2004.

L. N. Trefethen, D. Bau. Numerical Linear Algebra. SIAM, 1997.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. Numerical Recipes: The Art of Scientific Computing. Cambridge University Press, 2007.

L. R. Scott. Numerical Analysis. Princeton University Press, 2011.

E. Suli, D. F. Mayers. An Introduction to Numerical Analysis. Cambridge University Press, 2003.