Syllabus

Math 4363/5383: **Numerical Analysis** Fall 2022

Prerequisite: MATH 2584 (Differential Equations) or my consent

Instructor: Mark Arnold
Office: SCEN 418 **Phone**: 575-7701
e-mail: arnold@uark.edu **web**: https://arnold.hosted.uark.edu

Office Hours:

MWF 10:00am - 10:50pm and by appointment (or drop by)– in my office or zoom

Text: Burden & Faires, *Numerical Analysis*, 10^{th} edition*
Material covered: Chapters 1-5 of text and my notes

Grading on a 90/80/70/60 scale based on the following

Exam over chapters 1 and 2	100
Exam over chapters 3 and 4	100
Take-Home Final exam	100
In-Class Final exam (chapter 5 & comprehensive)	100
Homework & Programs	200
Total (after dropping lowest exam)	500

Notes

→ * The 9^{th} edition of our text *is* acceptable, but you are responsible for problems assigned from the 10^{th} edition.

→ The final is December 12, 12:45 – 2:45.

→ No exams may be made-up without my prior approval.

→ You are responsible for material covered and announcements made at any lecture, *even if you miss that day.* The lectures will be recorded.

→ Programs will be written in the Matlab programming language; it is licensed to all UofA students (https://its.uark.edu/help/ta/matlab-student-setup.php)

→ MATH 5383 students may have larger assignments, but same exams.

→ **Inclement Weather Policy**: A Test will be rescheduled to the next class meeting if the Fayetteville public schools are closed due to weather on a test day.

→ *As a core part of its mission, the University of Arkansas provides students with the opportunity to further their educational goals through programs of study and research in an environment that promotes freedom of inquiry and academic responsibility. Accomplishing this mission is only possible when intellectual honesty and individual integrity prevail.*

Each University of Arkansas student is required to be familiar with and abide by the University's 'Academic Integrity Policy' which may be found at http://provost.uark.edu/ Students with questions about how these policies apply to a particular course or assignment should immediately contact their instructor.

Goals

We will be surveying some of the more mathematical aspects of the discipline of *scientific computing*. Scientific computing deals with mathematically formulated problems from mathematics, engineering, physics, chemistry, medicine, biology, AI/ML, etc. and many other disciplines through statistical/data-science computations.

The tools in the discipline consist of *hardware* (calculators, computers, super-computers, graphics-cards, networks, etc.), *software* (c, fortran, java, python, julia, matlab, R, Hadoop, etc.) and, of course, the *methods* for solving the mathematical problems.

Numerical methods are the mathematical algorithms which are coded in software and run on hardware. We will approach some of the most fundamental problems in applied mathematics and investigate methods for their approximate solution on computers.

We will *not* cover numerical linear algebra (see, e.g. MATH 4353/5393), discrete algorithms (see, e.g. CSCE 4133), or symbolic computation (like Mathematica, Maple, Derive, etc.). Nor will we cover the use of software packages to solve our problems (this course is more about *creating* such packages).

While we are interested in the methods, we are most interested in the *development* and *analysis* of the methods. This is the main difference between a numerical analysis course vs. a numerical methods course. In particular, I hope that you accomplish the following:

- → Develop an understanding of how we represent the real numbers on a computer (floating point) and the resulting mathematical and computational implications.
- → Develop an understanding of the difference between easy and hard problems and good and bad methods.
- → Develop methods for solving equations in one variable (solve $f(x) = g(x)$ for x).
- → Develop interpolation (and approximation) methods; that is, given a set of points in the xy-plane, find a simple function that passes through (or near) the points.
- → Develop methods for approximating the derivative and the definite integral of a function (i.e., given a function f, compute $\nabla f(x_0)$ or $\int_\Omega f(x)\, d\Omega$).
- → Develop methods to approximate the solutions to initial value problems (i.e., approximate the function y that satisfies $y' = f(t, y)$ with $y(a) = \alpha$).
- → Gain programming experience by implementing many of these methods in floating point arithmetic (Matlab).