MATH 6313.001 (84377) Syllabus
Numerical Analysis
Fall 2016, Tu/Th 4:00–5:15pm, ATC 2.101

Instructor: Dr. Minkoff

Office: FO 2.402B
Phone: (972) 883-6695
Email: sminkoff@utdallas.edu
Website: *http://www.utdallas.edu/∼sminkoff*

Note that I will maintain a web page for this course linked from my main web page. (I will not be using eLearning.)

Office Hours: Tuesdays 2:30–3:30pm or by appointment.

Prerequisite: Knowledge of a high-level programming language and linear algebra and calculus through multivariable calculus. I expect students to be able to both understand proofs we do in class and to be able to prove results themselves on the homework. Therefore I prefer that students have successfully completed an undergraduate Real Analysis course before taking this course. Note that we will be using Matlab exclusively in this course. It is highly recommended that all students taking this course will have a working knowledge of MATLAB.

Course Description (from the catalog): A study of numerical methods including the numerical solution of non-linear equations, interpolation, approximation by polynomials, numerical integration. Numerical solution of ordinary differential equations including initial value problems and two-point boundary value problems.

Texts — Required: *Numerical Analysis: Mathematics of Scientific Computing*, 3rd Edition, by Kincaid and Cheney. Publisher: Brooks/Cole, 2002.
Additional Reference: *An Introduction to Numerical Analysis* by Atkinson. Publisher: John Wiley & Sons 1989.

Useful MATLAB Reference: *Mastering MATLAB*, by Hanselman and Littlefield. Publisher: Prentice Hall, Inc.

Grading Policy:

Homework	40%
Midterm Exam	30%
Final Exam	30%
Total	100%

Homework: There will be one homework due every 1–2 weeks on Thursdays (the length of time will depend on the difficulty of the assignment). Homework is to be turned in at the START of class on Thursday or can be slipped under my office door *prior* to class on Thursday if you must miss class for some reason. *Late homework will not be accepted.*

Please note that the homework constitutes a substantial portion of your overall grade. In order to learn the concepts and be able to apply them to solving problems on exams, etc., you are strongly encouraged to devote as much time as possible to working the homework problems. It is possible that not all homework problems will be able to be graded, but most of your learning will come from devoting good chunks of time each week to the homework. I encourage you to discuss the homework assignments with other students in the class. However, I expect the homework you submit for grading to be written up by you alone (this includes computer programs which must not be duplicates of programs other students turn in).

Tests: No make-up exams will be given except *possibly* in the case of a serious emergency. In such a case I *must* be notified *in advance*. There will be no exceptions to taking the final exam at the date, time, and place specified by the University (TBD). The final exam will be comprehensive although material covered after the midterm will be emphasized.

Learning Goals and Course Motivation: Numerical Analysis is the study of algorithms for solving mathematical problems on computers. Most real world integrals can't be evaluated exactly (i.e., their antiderivative isn't known). Most real world differential equations have non–constant coefficients or must be solved over irregularly–shaped domains and thus must be solved approximately on a computer. In many situations one is working with data points underlying a function rather than an explicit functional form, and therefore, one must approximate the function of interest by one which is easy to manipulate and which gives the character of the function in question at least at a specific set of points. In all of these cases the approximation used to solve the mathematical problem leads to an error which one would like to understand in order to make decisions about whether the solution is accurate enough for the given task. Does the error lie within a specified tolerance or bound? Does it grow with increasing time? In this course you will explore the world most scientists and engineers work in daily but which is different from what you have seen in previous math courses because the solutions to these problems are by necessity approximate.

Specifically, in this course you will:

1. Review how to represent numbers in different bases and how to convert between different number systems. Emphasis will be placed on base 2 which is the most important for representation of numbers on computers.

2. You will learn why round–off error is so important for numerical algorithms. All numbers represented on a computer must be stored with a finite number of bits. Hence most numbers cannot be stored exactly on a computer.

3. We will discuss finding roots of functions which is especially important in the study of optimization.

4. We will learn ways to represent a table of data values $(x, f(x))$ if the underlying function which generated the data isn't known *a priori*. The most common way to describe such data is by fitting a degree n polynomial to the set of $n+1$ points (e.g., fitting a line to two data points). However, there are many other useful representations of the underlying function.

5. We will learn how to decompose functions into a sum of sines (or exponentials) which is important for signal processing, data compression, image processing, etc. And we will learn an extremely efficient way to use computers for this purpose.

6. Most integrals can't be evaluated analytically and must be approximated on a computer. Moreover, computers can't take limits and hence approximating integrals and derivatives on computers requires dropping the limit idea and taking "small" but not infinitesimal quantities in the approximation. Our goal is to understand how accurate such approximations are – a topic that is especially important in the study of differential equations.

7. Finally we investigate iterative techniques for approximating solutions to ordinary differential equations on computers. The most basic idea is Euler's method which approximates the solution at a point in the domain by a short line segment. Other more sophisticated techniques will be discussed.

Academic Conduct: I take academic dishonesty *very seriously* and will not tolerate it in this class in any form. **Academic misconduct includes willfully cheating on or giving aid during an exam or copying homework assignments (from the web, from each other, or from a solutions manual). Blatant copying on an exam, homework assignment, or computer assignment will result in a grade of zero for that work.** Further information on the academic conduct policy can be found at http://www.utdallas.edu/deanofstudents/dishonesty/

UT Dallas Syllabus Policies and Procedures:

The information at http://go.utdallas.edu/syllabus-policies constitutes the University's policy and procedures segment of the course syllabus.

The descriptions and timelines contained in this syllabus are subject to change at the discretion of the Professor.

Class Attendance: I expect students to attend class and to turn up **on time**. Rarely do students do well in classes which they do not attend, and I will be less likely to give outside assistance to students who regularly miss class. Further, students arriving late for class disrupt the entire class.

Email: I am happy to answer questions about the class via email. However, I will not respond to email which does not include the name of the sender. Also, students should be aware that discussions of class concepts and involved homework questions are best asked in person during office hours. I reserve the right not to answer an email question if I feel the topic would best be discussed in person.

Tips for Succeeding in this Class:

1. Before you attempt the homework you should *read the sections in the textbook if appropriate and study your notes.*

2. You will benefit greatly from working with others in the class so long as you use your peers as a way to hash over concepts and not a way to "get the answers". In other words, *start early* and use your fellow-classmates to discuss the best way to approach the problems. Then go off and try to work out the details yourself.

3. **Begin the new homework assignment the same day you turn in the previous assignment!** Do not wait 3–4 days to start the homework as then you will not have enough time to digest the material or understand the point of the problems. When computer assignments are given, starting early on the homework is essential. Debugging programs takes time and your grade and learning will suffer if you attempt the computer problems at the last minute.

4. Come to office hours and get help if you are stuck. It is much better to get help early than to wait. I may ask you to show me what you've come up with at the board so you should have at least attempted the homework problems before asking for help.

Important Dates:

Date	Notes
8/22/16	First day of class
8/29/16	Last day to register and last day to add/swap
9/7/16	Last day to drop class without a "W"
10/13/16	Midterm Exam
11/7/16	**Absolute Last day to drop class**
12/7/16	Last day of classes
TBD	Final Exam

Math 6313, Fall 2016, Tentative Schedule:

Date	Section/Topic
Tu 8/23/16	First Day Handout; §1.1 – Basic Concepts and Taylor's Theorem
Th 8/25/16	§2.1 – Representation of Numbers in Different Bases
Tu 8/30/16	§2.1 – Floating-Point Numbers and Roundoff Errors
Th 9/1/16	§2.2 – Absolute and Relative Errors: Loss of Significance
Tu 9/6/16	§2.2 – Absolute and Relative Errors: Loss of Significance
Th 9/8/16	§3.1 – Bisection Method
Tu 9/13/16	§3.2 – Newton's Method
Th 9/15/16	§3.2 – Newton's Method for Nonlinear Systems
Tu 9/20/16	§3.4 – Fixed Points and Functional Iteration
Th 9/22/16	§6.1 – Polynomial Interpolation
Tu 9/27/16	§6.2 – Divided Differences
Th 9/29/16	§6.1 – Chebyshev Polynomials
Tu 10/4/16	§6.4 – Spline Interpolation
Th 10/6/16	§6.8 – Best Approximation: Least-Squares Theory
Tu 10/11/16	§6.12 – Trigonometric Interpolation
Th 10/13/16	**Midterm Exam**

Date	Section/Topic
Tu 10/18/16	§6.13 – Fast Fourier Transform
Th 10/20/16	§6.13 – Fast Fourier Transform
Tu 10/25/16	§7.1 –Numerical Differentiation and Richardson Extrapolation
Th 10/27/16	§7.2 – Numerical Integration Based on Interpolation
Tu 11/1/16	§7.3 – Gaussian Quadrature
Th 11/3/16	§7.3 – Gaussian Quadrature
Tu 11/8/16	§7.5 – Adaptive Quadrature
Th 11/10/16	§8.2 – Taylor-Series Methods
Tu 11/15/16	§8.3 – Runge-Kutta Methods
Th 11/17/16	§8.4 – Multistep Methods
Tu 11/22/16	**Fall Break**
Th 11/24/16	**Fall Break**
Tu 11/29/16	§8.4 – Multistep Methods
Th 12/1/16	§8.5 – Local and Global Errors: Stability
Tu 12/6/16	Review for Final Exam
TBD	**FINAL EXAM**