MATH 934: Topics in Differential Equations
UNL, Spring 2017, Section: 001
Lecture: M, W, F, 8:30 am-9:20 am, Avery Hall 351

Instructor: Dr. Adam Larios **Email:** `alarios@unl.edu`
Office: Avery Hall 305 **Math Dept. Phone:** (402) 472-7250
Office Hours: M,W,F, 2:30 pm - 3:20 pm, or by appointment
Web: `www.math.unl.edu/~alarios2/courses/2017_spring_M934/content.shtml`

Prerequisites: MATH 314/814 (Linear Algebra), and MATH 325 (Elementary Analysis), or equivalents. You are also expected to know differentiation and integration techniques from calculus, as well as the material from multivariable calculus. You are also expected to be able to understand mathematical proofs. This course will require basic computer skills. We will learn programming, but prior knowledge of programming is *not* a prerequisite.

Textbook: *Partial Differential Equations with Numerical Methods.* Stig Larsson, Vidar Thomee. Springer, 2008. ISBN: 978-3540887058.

Contacting me: **NOTE:** Because of privacy rights, **I cannot discuss grades over email or telephone. Please do not email me asking about your grade. I will not be able to give you any information.** Of course, I am happy to discuss grades in my office.

Description: We will solve differential equations using numerical methods, and this does not mean that we will leave mathematical rigor or beauty behind! The subject of numerical PDEs is full of clever ideas, elegant structures, dazzling schemes, and subtle concepts. The world of PDEs is so vast, that you can spend several lifetimes studying just one PDE, and yet there are thousands of PDEs out there. They are used to model phenomena such as weather, turbulence, blood flow, cancer growth, traffic, financial markets, ecology, acoustics, electricity, magnetism, star formation, and the bending of space-time itself. Solving them not only unlocks new areas of science, but often leads to pretty pictures that can amaze people in your poster sessions and astonish audiences at conferences.

To set the stage, we will begin with numerical solutions of ODEs (ordinary differential equations). We will quickly move on to numerical solutions of PDEs (partial differential equations). We will learn spectral/Fourier methods, finite element methods, and, time permitting, several other methods. We will learn to program in Matlab, and we will also also a finite element library called FEniCS, which uses Python.

No programming background is necessary, and it is not necessary for you to have taken a course in PDEs. An undergraduate ODE course, such as Math 221, advanced knowledge of linear algebra, such as Math 415/815 (or anything beyond Math 314), and some analysis, such as Math 825/826 (could be taken concurrently), should be sufficient.

Projects: The majority of the course grade will be based on projects. These will be mostly coding projects, but they may also involve some mathematical analysis. As codes often either work or do not work, it can be difficult to give a scaled grade. Therefore, often codes will be checked off once they are working.

Exams: There will be no exams or quizzes

Reading: Please read the book frequently. It will help.

Homework: Overall, we will aim to keep the course load light; however, to learn this stuff, there will be times where you must try things on your own. Therefore, there will be a few mild homework assignments, but I will do my best to avoid problem sets which burn huge amounts of time, or keep you up all night. Think of them as "lunch-time exercises." Hopefully, you will be able to take these problems with you, and work on them casually when you have time over a few days. There is no need for a formal write-up; just discuss the problem with me at some point, and I will check it off.

Grading: Points will be associated with the course work. If your total number of points is within one standard deviation of the course mean, you will receive a grade in the "A-/A/A+" range. Grades of A+ will be reserved for exemplary course work, although it would certainly be possible for all student to receive a grade of A+. If the total number of points is between 1 and 2 standard deviations below the mean, will receive a grade in the "B-/B/B+" range, and so on for each interval of unit standard deviation. This grading system assumes that the majority of student complete the majority of the projects. Adjustments may be made in the the unlikely event that this does not occur.

Attendance & Preparation: There is no textbook that exactly fits the course goals; hence, the lecture notes are the primary record of the course. Regular attendance and attention is therefore critical. It will be helpful for you to browse through the material before it is presented in class.

Daily attendance for class lectures is expected and is extremely important. While attendance is not recorded, missing even one class will put you behind. You are responsible for all material and announcements in class regardless of whether or not you attended. You are also responsible for making arrangements with another classmate to find out what you missed.

If you know ahead of time that you will be gone. Please let me know as soon as possible in advance. Reasonable accommodations will be made for university-excused absences.

Computing: We will be writing programs to implement numerical methods. I will prepare directions for writing in Matlab. You may use your own computing equipment, and you may also use the computers in the Math Department computer lab in Avery 9, or in labs around campus. Matlab is free to download for UNL students, and can be accessed here:

`http://procurement.unl.edu/matlab-licenses`

At some point, I will also ask you to install a software library called FEniCS. Both Matlab and FEniCS can take up several GB, so please plan to have space available (say,

Collaboration: Collaboration is encouraged in this course. However, copying someone else's work and submitting it as your own is unacceptable.

Incompletes: A grade of "incomplete" may be considered if all but a small portion of the class has been successfully completed , but the student in question is prevented from completing the course by a severe, unexpected, and documented event. Students who are simply behind in their work should consider dropping the course.

Programming: This course contains a gentle introduction to scientific computing with Matlab and FEniCS via Python. Matlab and Python are two of the most widely-used programming languages in science, mathematics, and engineering, and can be a very strong asset to future scientific work. **No previous programming experience is assumed.** Students are assumed to be able to have basic computer skills, such as using a mouse, keyboard, etc., and be able to download and install programs and navigate websites. Basic programming in these languages will be taught in class on certain days. Programming assignments and/or projects will be announced in class.

ADA Statement: Students with disabilities are encouraged to contact the instructor for a confidential discussion of their individual needs for academic accommodation. It is the policy of the University of Nebraska-Lincoln to provide flexible and individualized accommodation to students with documented disabilities that may affect their ability to fully participate in course activities or to meet course requirements. To receive accommodation services, students must be registered with the **Services for Students with Disabilities (SSD) office**, 132 Canfield Administration, 472-3787 voice or TTY.

Grade Questions: Any questions regarding grading/scoring of homework, exams, or projects must be made within two class days from when they were handed back, or no change in grade will be made.
NOTE: Because of privacy rights, **I cannot discuss grades over email or telephone. Please do not email me asking about your grade. I will not be able to give you any information.** Of course, I am happy to discuss grades in my office.

Important Dates:

Jan. 20, 2016 (Fri):	Last day to withdraw from this course and not have it appear on your transcript.
Mar. 3, 2016 (Fri):	Last day to change your grade option to or from Pass/No Pass.
Mar. 19-26, 2016:	Spring break, no class.
Apr. 7, 2016 (Fri):	Last day to drop this course and receive a grade of W. (No permission required.) After this date, you cannot drop.

Departmental Grading Appeals Policy: Students who believe their academic evaluation has been prejudiced or capricious have recourse for appeals to (in order) the instructor, the departmental chair, the departmental appeals committee, and the college appeals committee.

Disclaimer: While this syllabus was prepared carefully and according to information available at the beginning of the semester, changes may be necessary in the interest of good teaching. Changes to any of the information above will be announced in class and posted on the class web site. This includes in particular possible updates or corrections to the syllabus, and changes of exam dates. Care has been taken to avoid any conflict between this syllabus and official university policy. Any such conflict, if it exists, is purely accidental, and appropriate measures will be taken to rectify any such mistake.

Tentative List Of Topics: The following tentive list of topics is a rough guide to the material covered in the course, but is subject to change. Updates and changes to the content will be announced in class, over email, on Canvas, or on the course website.

- Intro to Matlab, loops in chaos
- Euler's method for ODEs, with resolution error
- RK-4, order of error, stability

- Fourier transforms and the DFT
- FFT, Parseval
- FFT in Matlab to compute derivatives (energy error)

- The heat equation and the CFL
- Coding the heat equation
- Burgers equation and aliasing

- Functional analysis review
- Overview of finite element methods
- Coding the stiffness matrix

- Solving 1D Poisson in Matlab
- Error estimates
- Checking error vs. resolution

- Lagrangian interpolation
- Orthogonal polynomials
- Special polynomials

- Newton-Cotes numerical quadrature
- Gaussian numerical quadrature
- 2D numerical quadrature

- Introduction to FEniCS
- Weak derivatives
- Sobolev spaces

- Evolution equations in FEniCS
- Hilbert spaces
- Boundary conditions

- Duality
- Projections
- Lax-Millgram

- Triangular elements
- Transforms and the reference simplex
- Some notes on meshes

- Coding Finite Element Methods
- Data Structures
- The unexpected difficulty of solving $Ax = b$

- Some comments on finite-precision arithmetic
- Preconditioners
- Using software libraries

- The Stokes equations
- Mixed finite elements
- Saddle Point Problems

- Additional topics if time: Parallel computing basics, Finite volume methods, finite difference methods, Symplectic integrators, IMEX methods, and more.
