**CVEN 5537 NUMERICAL METHODS IN CIVIL ENGINEERING**
MWF, 10:10-11:00am, ECCE 1B41, FALL 2022

**INSTRUCTOR:** Professor Richard Regueiro, richard.regueiro@colorado.edu
**Office Hours:**
*Hybrid In-Person in ECOT 421*: MW 11:10am-12pm, Th 11am-12pm, with zoom option ( https://cuboulder.zoom.us/j/7462640502 )

**GRADER:** TBD
**Office Hours:**
*In-Person in ECCE 157 (Bechtel Lab West)*: TBD
*Zoom office hour* ( https://cuboulder.zoom.us/j/? ): TBD

**DESIRED PREREQUISITES:** Undergraduate Engineering Calculus sequence through Differential Equations and Linear Algebra.

**OBJECTIVES:** To introduce you to scientific computing for solving ordinary and partial differential equations (ODEs and PDEs) with applications to engineering and scientific problems. This course will provide you with a set of tools (numerical methods, proficiency with MATLAB and/or Python, and also COMSOL FEA) that you will use in other graduate courses, and potentially also in your thesis research. You will also gain a general awareness of computational approaches to solving problems in engineering, which will enable you to approach other problems that we will not cover in the course, but may be pertinent to your research.

**SYLLABUS:** Rather than take the classical teaching and learning approach to numerical analysis, which focuses on approximation and interpolation theory, numerical integration, solving systems of linear equations and finally ODEs and PDEs (APPM teaches a 3-course sequence along these lines), we will focus mainly on ODEs and PDEs; with some coverage of solving linear systems of equations. This is because one of the most basic skills needed in subsequent graduate courses (and research) is to model physical systems described by ODEs and PDEs. There are other sub-branches in Numerical Methods, including Optimization and Statistics, which we will not cover in the course; these topics are covered in other courses (e.g., CVEN 5454).

We begin with an overview of MATLAB and Python, which we use in the course because most entering graduate students have limited previous experience with scientific programming beyond EXCEL. The numerical methods for solving PDEs that we will cover are *finite-difference methods*. In terms of theory and programming, we have chosen to stay within the framework of finite-difference methods and cover a broader range of PDEs and boundary value problems, so you can see how a single approach applies to a variety of problems. Several other introductory courses on finite element methods are offered in the college (e.g., CVEN 5511). Without covering the theory or programming, you will also be exposed to a Finite Element Analysis (FEA) software package, such as COMSOL.

Class notes will be provided on canvas.colorado.edu, along with MATLAB and Python coding examples, problem sets, projects, etc.

SUPPLEMENTARY READING: ( You may purchase one of these books if you wish, but the matlabacademy should cover all relevant topics for the course with respect to MATLAB; for Python, there are many online tutorials, such as https://www.freecodecamp.org/news/python-code-examples-simple-python-program-example/ )

Introductory books/resources on MATLAB:

- Getting Started with MATLAB, R. Pratap (Oxford University Press)
- MATLAB - An Introduction with Applications, A. Gilat, (John Wiley and Sons)
- Engineering and Scientific Computations Using MATLAB, S. Lyshevski (Wiley)
- MATLAB's help and documentation are very useful, see www.mathworks.com/help/matlab/getting-started-with-matlab.html
- MATLAB's online academy: matlabacademy.mathworks.com, click "My Courses" and login using your MathWorks account ( obtained through OIT: oit.colorado.edu/software-hardware/software-downloads-and-licensing/matlab , where you can download Matlab for free )

References on Numerical Methods for ODEs and PDEs:

- Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems, R.J. LeVeque (SIAM)
- Numerical Methods for Partial Differential Equations, W. Ames (Academic Press)
- Numerical Solution of Partial Differential Equations, Morton and Mayers (Cambridge)
- Numerical Recipes in (FORTRAN or C or C++), W. H. Press and others (Cambridge)
- Fundamentals of Engineering Numerical Analysis, P. Moin (Cambridge University Press, numerics.stanford.edu)

SEQUENCE OF TOPICS (subject to change by Instructor)

1. **MATLAB and Python programming overview and exercises:**
   Project I - codes for simple problems (e.g., root-finding, series summation).

2. **Numerical methods for systems of ODEs (initial value problems):**
   Project II - Writing your own ODE solver in MATLAB or Python, and using it to solve some ODE systems, comparison to MATLAB's and/or Python's built-in ODE solvers.

3. **Heat/Diffusion PDE and numerical methods:**
   Project III – MATLAB and Python codes for explicit and implicit finite-difference solutions to the one-dimensional heat/diffusion PDE with applications.

4. **Elliptic PDEs – Laplace's and Poisson's equations:**
   Finite-difference discretization for 2D problems (x,y coordinates), matrix assembly and structure, sparse matrices, direct and iterative solution.

5. **Basic Iterative solution of systems of linear equations arising from Laplace's equation:**
   Project IV - MATLAB or Python code for iterative solvers applied to the Laplace/Poisson equation with examples from groundwater flow, membrane deflection, heat conduction.

6. **Nonlinear boundary value problems and time-dependent PDEs** – finite-difference discretization and solution using Newton's method and Picard iteration:
   Project V - MATLAB or Python code for nonlinear PDE in one dimension (nonlinear diffusion-reaction or heat transfer with phase change).

**GRADING: (subject to change by Instructor)**
15%: problem sets during earlier part of course
10%: in-class midterm exam (open book, open notes; date TBD (usually end of October); based on Problem Sets 3 and 4 and Projects I and II (no programming, just concepts and equations))
60%: 4 Matlab or Python programming projects (you pick your preferred programming language)
15%: individual final programming project of your choosing (5 minute presentations during final exam period; final report due by end of final exam week), or Project V

*One assignment (problem set or project, but NOT the final project) may be submitted 1 week late without prior approval* (but please let me know you're taking your one week late assignment).
*Otherwise, no late assignments will be accepted* (unless under extreme circumstances, in which case email me).

**Zoom Meeting Information to connect to lectures in real time:**
https://cuboulder.zoom.us/j/98678472998

For more information, visit the OIT Zoom website:
oit.colorado.edu/services/conferencing-services/web-conferencing-zoom/help/getting-started

There are many required statements to be entered in the syllabus, such that it is more straightforward, and allows for updates, to post directly the url: academicaffairs/policies-customs-guidelines/required-syllabus-statements