Civil, Environmental and Architectural Engineering

UNIVERSITY OF COLORADO **BOULDER**

CVEN 5835, NONLINEAR ANALYSIS OF RC AND MASONRY STRUCTURES

This course focuses on the basics behind state of the art nonlinear static and dynamic analysis modeling of Reinforced Concrete (RC) and Masonry structures using the opensource code OpenSees (Open Software for Earthquake Engineering Simulation) and the programming language python. The course includes a theoretical part and practical application with OpenSees and python.Production Quality Computational Multiphysics

CVEN 5838-200B (for credit), NCEN 5838-570 (for non-credit), PRODUCTION QUALITY COMPUTATIONAL MULTIPHYSICS SOFTWARE DEVELOPMENT

Multiphysics code development is a multidisciplinary endeavor, which by its nature requires contributors who have different strengths. At the same time, each contributor is eventually required to perform beyond their trained area of expertise. In this course, you will be exposed to some of the most essential elements of multiphysics code development, in such a way that builds your knowledge, skills, and confidence both for your current research efforts and your possible future role in the field of computational physics. The key multidisciplinary components include numerical methods for discretizing time-dependent partial differential equations, numerical analysis, linear and nonlinear solvers, computer programming, software verification, version control, production process control, testing, and the underlying physics being simulated. This course provides an in depth survey of examples across all of these areas and provides a level of assurance to the successful student that they have enough knowledge to confidently participate in a multiphysics software endeavor in the future while also accelerating their own research now.

The course uses a combination of technical homework problems, software development assignments, and a project to instill understanding and develop skills. There is considerable flexibility in project choices. You may work alone or on a small team. You may start a project from scratch or use what you learn in this course to advance an existing code, e.g., your research. For every project choice, you will be required to implement multiple components, depending on their complexity and state of the code with which you start, that were not pre-existing in your work. Examples include: (1) A new discretization method, (2) A new solver method, (3) Self-documenting regression tests, (4) A test harness, (5) Verification testing, or (6) Repository management. You will be required to write a project pre-proposal, proposal, update, and final report. The goal here is to learn by doing, and to make the "doing" part as relevant as possible to your current research or production work.