ME 471 / AE 420 / CSE 451: Finite Element Analysis

Spring 2024

Instructor:	Brian Mercer	**Time:**	MW 12:00pm – 1:20pm
Email:	bmercer@illinois.edu	**Place:**	114 Transportation Building

Course Web Pages:

1. Canvas - Syllabus; assignments and due dates; lecture notes; etc.
2. Gradescope - Written homework submission
3. PrairieLearn - Programming assignment submission
4. Campuswire - Course forum for homework and conceptual questions

Teaching Assistant: Pavan Ravi, pavanr2@illinois.edu.

Office Hours: Instructor and TA office hours will be posted to the Canvas course page.

Credit: 3 or 4 undergraduate hours; 4 graduate hours. Students seeking the additional hour will be assigned additional problems on some assignments.

Textbook: There is no required textbook for this class. Recommended references are listed below:

1. R. D. Cook, D. S. Malkus, M. E. Plesha, R. J. Witt, *Concepts and Applications of Finite Element Analysis*, 4th edition, John Wiley & Sons, 2002.
2. K. H. Huebner, D. L. Dewhirst, D. E. Smith, T. G. Byron, *The Finite Element Method for Engineers*, 4th edition, John Wiley & Sons, 2001.
3. T. J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover, 2000.

Prerequisites: CS 101 and ME 371 or TAM 470. Alternatively, AE 370 for AE students..

Course Objectives: After completing the course, students will:

1. Understand the mathematical theory and foundation of the finite element method for linear differential equations.
2. Be able to write finite element computer programs to solve problems involving heat transfer and linear elasticity.
3. Be able to use commercial FEA software to model complex engineering problems.

Course topics:

1. Direct approach to finite element method, applications to spring and truss problems.

2. Weighted residual and variational formulation approaches.
3. Review of common element types for 1D, 2D and 3D analyses; isoparametric element formulations; beam elements.
4. Finite element formulations for solving steady-state and transient heat transfer problems.
5. Finite element formulations for solving static and transient problems in linear elasticity.
6. Modal (frequency) analysis for linear elastic structures.
7. FEM error behavior.
8. Computer programming implementation of finite element analysis: concepts and applications.
9. Use and applications of Abaqus commercial FEA software.

Grading Scheme: Assignments (60%), Midterm (15%), Final (25%). The total score s corresponds to the final letter grade as follows:

$97\% \leq s \leq 100\%$	A+	$93\% \leq s < 97\%$	A	$90\% \leq s < 93\%$	A-
$87\% \leq s < 90\%$	B+	$83\% \leq s < 87\%$	B	$80\% \leq s < 83\%$	B-
$77\% \leq s < 80\%$	C+	$73\% \leq s < 77\%$	C	$70\% \leq s < 73\%$	C-
$67\% \leq s < 70\%$	D+	$63\% \leq s < 67\%$	D	$60\% \leq s < 63\%$	D-
$s < 60\%$	F				

Assignments: Assignments will be given every 1–2 weeks. The schedule of due dates will be maintained on the Canvas course site. Assignment types are broken intwo two categories: Homework sets (HW) and Programming Assignments (PA). Each assignment is weighted equally in the final grade calculation of the Assignments category, regardless of the assignment type.

Homework (HW): Homework assignments will involve written analysis problems or problems using the FEA software Abaqus. General rules:

1. You may work together with other students to solve homework problems, but your submitted work must be your own.
2. You may handwrite your work for analysis problems as long at it is clearly legible. Illegible written work will automatically receive a zero.
3. All homework assignments will be submitted to Gradescope as a PDF file unless otherwise noted. You must use the Gradescope interface to tag each problem within the PDF in order to receive credit for the problem. **Failing to tag pages for each problem may result in a zero grade for that problem**.

Programming Assignments (PA): Programming Assignments are hosted in PrairieLearn and must be completed using the Python programming language. It is recommended, but not required, that you install a Python programming/development environment on your personal computer to help complete these assignments (see notes on **Software** section of the syllabus). General rules:

1. You may work together with other students to solve programming assignments, but your submitted code must be your own.
2. You will submit your code to PrairieLearn for autograding. You can submit your code for grading an unlimited number of times before the due date without penalty.

Class attendance and participation: Class time will consist of a mixture of lectures, FEA computer labs, and in-class activities (solving example problems, group/classroom discussions, etc.). Regular attendance and participation in class activities is expected.

Campuswire online forum: Campuswire can be used to get help on assignments you are working on or conceptual questions from class. Campuswire should be used in lieu of email for these kinds of questions (please use email for issues outside the scope of getting help with course content). Please follow these guidelines:

1. Consider posting your questions to "Everyone" (rather than just "Instructor & TAs"), even if anonymously, so that everyone can benefit from the answer/feedback. Anyone, including students, is welcome to answer a given question.
2. Do not make posts to "Everyone" about homework/programming problems you are working on that contain the entire written or code solution. Such posts will be deleted.
3. Please follow the guidelines on Canvas about seeking help on programming problems and debugging code.

Software: You will need access to the following software in this class:

1. Python: Programming Assignments must be completed using Python. It is possible to write and test your code entirely in-browser using PrairieLearn, but you may wish to use Python outside of PrairieLearn for more convenient testing and debugging. For installation of Python locally on your personal computer, we recommend Anaconda:
   - Download and step-by-step instructions for installation: https://docs.anaconda.com/anaconda/install/
   - Using Python with Anaconda: https://docs.anaconda.com/anaconda/user-guide/getting-started/
   - An optional python IDE if you are used to Matlab: https://www.spyder-ide.org/
2. Abaqus: We will use the FEA software Abaqus for some in-class lab sessions and homework assignments. You can use Abaqus in EWS labs or you can install on your personal computer, see getting an educational license through the UIUC WebStore.

Homework/Programming Assignment late policy: Late homework submissions are penalized 10% per day, up to 2 days late. After the 2nd day (i.e. 48 hours after the initial deadline), the assignment will receive a zero. Please carefully review the rules below regarding extension requests for homework:

1. **All extension/make-up work requests must be received in advance of the due date or they will not be considered.**
2. Every student may use **one** no-questions-asked (NQA) 48 hour extension on an assignment (Homework or Programming Assignment) during the semester. You must email the instructor and TA to indicate you are using your NQA extension request for the given assignment. As the name implies, there is no need to explain your situation or provide documentation.
3. Besides the NQA extension request, you may request an assignment deadline extension for the following situations, and must also provide a letter from the Office of the Dean of students:
   - Illness for 3 or more days
   - Personal crisis (e.g., car accident, required court appearance, death of a close relative).
   - Required attendance at an official UIUC activity (e.g., varsity athletics, band concert).
   - For more extreme situations that involve an extended absence for more than a few days, please contact the instructor as soon as possible so we can discuss how to proceed.
4. Note that regardless of documentation provided, the final decision to grant an extension always lies with the instructor.

Midterm and Final Exam: A Midterm exam will be given in class on Wednesday March 6, 2024 and count as 15% of the final course grade. The Final Exam counts as 25% of the course grade and will be given on Friday May 3, 2024 from 1:30pm to 4:30pm, as indicated by the university registrar final exam schedule for Spring 2024. Details regarding each exam will be discussed in class in advance of the exam dates.

Academic integrity: Every student is expected to review and abide by the university's Academic Integrity Policy as outlined in the Student Code. It is your responsibility to read this policy to avoid any misunderstanding. Ignorance is not an excuse. Do not hesitate to ask the instructor if you are ever in doubt about what constitutes plagiarism, cheating, or any other breach of academic integrity.

Online Section (ONL): The online section of this course is conducted as described in the syllabus sections above, with the following additions and modifications:

1. Lecture sessions will be made available as asynchronous recordings, posted soon (generally same-day) after the live class meeting has concluded. A link to the UIUC Mediaspace channel hosting the video lectures will be maintained on the Canvas course page.
2. All exams will be scheduled on the same date (but not necessarily the same time) as the in-person exams indicated in this syllabus. Exams will be scheduled, administered and proctored in-person by a local proctor contact, arranged by the Office of Online & Professional Engineering Programs from the Grainger College of Engineering.
3. Office hours are offered over Zoom by appointment, with either the instructor or TA. Please send us an email if you would like to meet, and we can work together to schedule a time.
4. You are responsible for owning or having access to a computer that can support the software that we will be using in this course. Abaqus, in particular, is a resource-intensive program and may not run smoothly on older computers, so please plan accordingly and aim to install and test the software early in the semester.