Lecture 01: Course Administration, Introduction, and Introduction to Matlab

Learning Objectives:

- Use variables, operators, and control structures to implement simple sequential algorithms
- Use Matlab m-files to create user-defined programs and functions
- Distill numerical results into a readable format that answers specific civil engineering analysis and design questions

Topics/Outline:

1. Syllabus and course administration
2. Introduction to numerical methods
3. Tour of Matlab
Syllabus

The following presents the syllabus for Fall 2015.

**CVEN 302 Computer Application in Engineering and Construction. (2-2). Credit 3.** Application of computers to solution of civil engineering problems using various numerical methods; structured computer programming; mathematical modeling and error analysis; solution of algebraic and differential equations; numerical differentiation and integration; curve-fitting; root-finding. Prerequisites: ENGR 112; MATH 308 or registration therein; admitted to major degree sequence in civil engineering.

**Learning Outcomes**

By the end of this course, students will be able to design and execute computer programming solutions to civil engineering problems using numerical methods.

To achieve this goal, students will learn to:

- Translate numerical methods into simple, reusable program modules
- Choose appropriate numerical methods for solutions to specific mathematical problems
- Analyze the applicability and accuracy of numerical solutions to specific mathematical problems
- Synthesize multiple program modules into larger program packages
- Distill numerical results into a readable format that answers specific civil engineering analysis and design questions

The course content is organized into five learning modules. The module content and related specific learning outcomes are summarized in the CVEN 302 Learning Modules.

**ABET Outcomes**

This course also contributes to the following ABET-identified outcomes of the civil engineering curriculum:

- Ability to apply knowledge of mathematics through differential equations, science (including physics, chemistry, and one additional area of science), and engineering
- Ability to identify, formulate, and solve civil engineering problems
- Ability to use modern tools, techniques, and computation methods necessary for civil engineering practice

**Keywords:**

Structured computer programming, Matlab, numerical methods, error analysis, root finding, arrays and matrices, regression, interpolation, numerical integration, numerical
differentiation, ordinary differential equations, initial value problems, boundary value problems, eigenvalue problems, partial differential equations.

**Textbooks**

There are two required textbooks for this course


**Course Calendar**

Please refer to the Calendar page.

**Grading**

Your final grade for the course will be calculated as follows:

- Problem Sets and Quizzes: 10%
- Programming Assignments: 20%
- Midterm Exam 1: 22%
- Midterm Exam 2: 23%
- Final Exam: 25%

Letter grades will be assigned from your total course score according to 90% to 100%: A, greater than or equal to 80% but <90%: B, greater than or equal to 70% but <80%: C, greater than or equal to 60% but <70%: D, below 60%: F. Please note that programming assignments are 20% of your grade: please do not neglect this work!

**Problem Sets**

Please refer to the Assignments page for a list of all homework assignments.

Problem sets will generally be assigned on a Monday or Wednesday and due within one week. Solutions will be posted the day before assignments are due.

Homework is an important part of the learning process and should be completed individually. You may ask others for help at places where you have made diligent attempts and have become stumped. You may ask others for confirmation of results at significant milestones in the problem. You may not share computer programs, Word documents, or Excel files. Copying another student's solution, even if you slightly change the presentation will be considered as cheating and given a grade of zero (see Plagiarism statement below). Referring to posted homework solutions is acceptable as long as the work you turn in is your own.

Homework problems must be answered clearly, showing all your work, and should be easy to follow. Where applicable, the solution to each problem should contain:

- A brief statement of the problem
- The general form of the equations used to solve the problem
- An equation with the plugged in numbers and the highlighted solution

Failure to include one of these elements will result in lost credit for the problem. *Not all homework problems may be graded.*

Unless you have a university excuse (see Absences below), late assignments will not be
accepted for credit. Please do not ask for exceptions. Your problem sets grade will be calculated as the average of your top ten scores on the twelve assigned problem sets.

**Class Participation and Quizzes**

You are expected to attend all classes, turn in all assignments, and complete all exams at their scheduled times. Exceptions are only permitted for university excused absences (see Absences below).

Classes will start on time, and pointers for the homework assignments and last-minute changes to the schedule may be announced in class. It is your responsibility to be in class to receive this information or to get the information from another student.

Quizzes may be given on Fridays and would feature one of the problems from the past week’s homework assignment. Quizzes are closed book and notes. Your quiz grade is factored with your grade for problem sets and count as half a problem set each. All quizzes will be included in the quiz grade. Because the quizzes test the homework material, timely completion of homework and attendance are essential to success in this class.

**Programming Assignments**

Please refer to the [Assignments](http://ceprofs.civil.tamu.edu/ssocolofsky/CVEN302/syllabus.html) page for a list of all programming assignments.

Programming assignments will generally be assigned each Wednesday and due the following Friday.

Programming assignments are a significant part of your grade: you should complete all assignments, putting forward your best effort, and cheating will not be tolerated. All programming assignments and their related reports will be submitted through [Turnitin.com](http://ceprofs.civil.tamu.edu/ssocolofsky/CVEN302/syllabus.html) and tested for plagiarism or copying.

Students may choose to complete programming assignments individually or in groups of no more than two students. The assignment must clearly state on the first page the names and UINs of each student who participated in the assignment, and each student will be responsible to turn a copy of the assignment in through their account on Turnitin.com (e.g., if two students work together, they must both submit a copy of the assignment through the website).

As with the homework assignments, you may ask others for help at places where you have made diligent attempts and have become stumped. You may ask others for confirmation of results at significant milestones in the problem. You may not share computer programs, Word documents, or Excel files. Copying another student’s solution, even if you slightly change the presentation will be considered as cheating and given a grade of zero (see Plagiarism statement below).

Each programming assignment will specify a format for reporting the results, but will generally consist of a short written memorandum followed by a program listing, code validation calculations, and complete program output.

Similarly to the homework, unless you have a university excuse (see Absences below), late assignments will not be accepted for credit. Please do not ask for exceptions. Your programming grade will be calculated as the average of your top eight scores on the nine assigned programming assignments. In addition, each assignment will feature a challenge problem, worth 30 extra points out of 100, and due at the same time as the assigned program; you may submit three challenge problems for points per semester. It is not necessary to complete any of the challenge problems, and they should not be attempted until the assigned problem is completed satisfactorily.

**Exams**

Two 110-minute midterm exams and one 2-hr final exam are scheduled (see Course
Calendar above for scheduling). The grading of the exams will be based on both the approach and the final answer. Exams will be closed book and closed notes. You may prepare a crypt sheet on the front and back of one page of 8 ½ x 11 paper for each exam. You will need a hand-held calculator for each exam. It is your responsibility to ensure that your calculator is working and will perform in the examinations.

**Plagiarism and Cheating**

"An Aggie does not lie, cheat, or steal, or tolerate those who do." Students are expected to understand and abide by the Aggie Honor Code presented on the web at:

- [http://aggiehonor.tamu.edu/](http://aggiehonor.tamu.edu/)

No form of scholastic misconduct will be tolerated. Academic misconduct includes cheating, fabrication, falsification, multiple submissions, plagiarism, complicity, etc. These are more fully defined in the above web site. As commonly defined, plagiarism consists of passing off as one's own the ideas, words, writings, etc., which belong to another. In accordance with this definition, you are committing plagiarism if you copy the work of another person and turn it in as your own, even if you should have permission of that person. Since the problem sets and programming assignments grade for this course is a high percentage of your total grade, no plagiarism or cheating will be permitted in these assignments. Violations will be handled in accordance with the Aggie Honor System Process described on the web site.

**Absences**

The university views class attendance as an individual student responsibility. Students are expected to attend class and to complete all assignments. Instructors are expected to give adequate notice of the dates on which major tests will be given and assignments will be due. For more details, please read Part I, Rule 7 of the Texas A&M University Student Rules at

- [http://student-rules.tamu.edu/rule07](http://student-rules.tamu.edu/rule07)

Please contact me as soon as you know you will miss a class or an exam so that a reasonable alternative can be accommodated. Unexcused absences will result in a grade of zero for the missed work. The instructor is under no obligation to provide an opportunity for the student to make up work missed because of an unexcused absence.

**ABET**

Students may be asked to allow copies of their assignments and exams to be submitted to the Accreditation Board for Engineering and Technology (ABET) review panel. The purpose of this is to demonstrate to ABET that our stated mission and objectives are being effectively implemented. Your grade will not be affected by participation.

**Americans with Disabilities Act (ADA)**

The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. If you believe you have a disability requiring an accommodation, please contact Disability Services, in Cain Hall, Room B118, or call 845-1637. For additional information visit [http://disability.tamu.edu](http://disability.tamu.edu).

**Export Controls**

All information included in the public areas of the course website is paraphrased from the course textbooks, which are public domain and publicly available. All of the programming and numerical methods covered in this course are listed in the course
catalog and are public domain. The online videos are protected and only available to students enrolled in the course; none of the material in the videos is export controlled, but this precaution prevents the unintended export of instructional material to an embargoed country.
CVEN 302
Computer Applications in Engineering and Construction

Learning Outcomes

Overall course learning outcome
By the end of this course, students will be able to design and execute computer programming solutions to civil engineering problems using numerical methods.

To achieve this goal, students will learn to:

• Translate numerical methods into simple, reusable program modules
• Choose appropriate numerical methods for solutions to specific mathematical problems
• Analyze the applicability and accuracy of numerical solutions to specific mathematical problems
• Synthesize multiple program modules into larger program packages
• Distill numerical results into a readable format that answers specific civil engineering analysis and design questions

Course Modules
This course is organized into five course modules. The following list summarizes these categories and provides the specific content and learning outcomes for each module.

Module 1: Basic Structured Programming (Lectures 1-7)
The first part of this course will introduce the basic building blocks of structured computer programs. These include the following topics:

1. Variables and data types
2. Logical operations
3. The basic control structures IF, FOR, and WHILE
4. Programs and user-defined functions
5. Variable passing and computer memory
6. Debugging techniques

Once the material in this module is mastered, students will be able to:

• Use variables, operators, and control structures to implement simple sequential algorithms
• Use Matlab m-files to create user-defined programs and functions
• Develop simple logical algorithms to solve engineering problems
• Generalize program code to create modules that encapsulate reusable parts of an algorithm
• Test program output for accuracy using hand calculations and debugging techniques
• Synthesize multiple program modules (including built-in Matlab functions) into larger program packages
Module 2: Introduction to Numerical Methods (Lectures 8-10)

Most of the programs created in this course will implement numerical solutions to civil engineering problems. In this section we introduce the basic notation for interpreting numerical methods through illustration by the following topics:

1. Taylor series expansion
2. Error measures
3. Root-finding techniques

Though brief, by the end of this module, students will be able to:

- Translate numerical methods into simple, reusable program modules
- Choose appropriate numerical methods for solutions to specific mathematical problems
- Analyze the accuracy of numerical solutions to specific mathematical problems

Module 3: Arrays and Matrices (Lectures 11-14)

Most numerical methods require holding multiple, related values in a single array for bookkeeping purposes or utilize matrices to find the solution to systems of simultaneous equations. This section introduces arrays and matrices through the following topics:

1. The Matlab array data type
2. Basics of linear algebra
3. Naive Gauss elimination
4. Gauss elimination with pivoting
5. The matrix inverse
6. Issues of numerical stability

By the end of this module, students will be able to:

- Store and manipulate datasets in Matlab arrays
- Write systems of equations in matrix form
- Manipulate equations following the rules of matrix algebra
- Write simple program modules to implement matrix operations
- Analyze the applicability and accuracy of matrix numerical solutions to linear systems of equations
- Calculate solutions to civil engineering problems using matrix algebra

Module 4: Data Modeling (Lectures 15-18)

One reason computers are so vital to civil engineering practice is the vast amount of data engineers are often required to process. This module will introduce several standard methods to describe and model data, including:

1. Regression
2. Interpolation
3. Extrapolation
4. Statistics
5. Measures of goodness-of-fit

Once the material in this module is mastered, students will be able to:
• Apply standard modeling techniques to describe and summarize large datasets
• Calculate results from data analysis using regression, interpolation, and extrapolation
• Analyze the quality of a particular model designed to represent a given data set
• Distill discrete data values into a readable format that answers specific civil engineering analysis and design questions

Module 5: Numerical Modeling and Simulation (Lectures 19-28)

Computers are widely used to simulate the behavior of complex systems using numerical solutions to differential equations. This module will introduce basic numerical methods for integration, differentiation, and solutions to differential equations. The topics covered include:

1. Integration
2. Differentiation
3. Initial value problems
4. Boundary value problems
5. Eigenvalues
6. Partial differential equations

This module somewhat synthesizes each of the prior modules of the course. By the end of this section, students will be able to:

• Implement algorithms to perform numerical differentiation and integration on a discrete numerical grid
• Apply numerical solutions to differential equations to build numerical models of civil engineering systems
• Analyze the applicability and accuracy of numerical solutions to differential equations
• Distill numerical results into a readable format that answers specific civil engineering analysis and design questions
Office Hours

Office hours for Fall 2015: Monday and Wednesday from 9:00 - 10:00 a.m. If you cannot meet during one of those times, you may schedule an appointment by email at socolofs@tamu.edu.
## Calendar

The following table presents a tentative course calendar for **Fall 2015**. In the Reading Assignment column, "M" stands for the Matlab programming book by Chapman and "C" is the numerical methods book by Chapra.

In addition to the weekly class meeting times, three important dates should be noted:

- **Midterm Exam 1**: Friday, October 9 from 8:00-10:00 in CVLB 421
- **Midterm Exam 2**: Friday, November 13 from 8:00-10:00 in CVLB 421
- **Final Exam**: Friday, December 11 from 10:00-12:00 in CVLB 421

You may view examples of past midterms and finals on the [Past Exams](#) page.

<table>
<thead>
<tr>
<th>LEC #</th>
<th>Topic</th>
<th>Reading Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/31</td>
<td>Course Administration, Introduction and Introduction to Matlab [pdf, m-files] Programming: Data Types, Arrays, Input / Output, Built-In Functions, M-Files, and Plotting [notes, m-files]</td>
<td>M: Ch1</td>
</tr>
<tr>
<td>9/23</td>
<td>Error and the Taylor Series Expansion [notes]</td>
<td>C: Ch4</td>
</tr>
<tr>
<td>9/25</td>
<td>Lab 3, continued</td>
<td>C: Ch5</td>
</tr>
<tr>
<td>9/28</td>
<td>Root Finding: Bracketing Methods [notes]</td>
<td>C: Ch6</td>
</tr>
<tr>
<td>9/30</td>
<td>Root Finding: Open Methods [notes]</td>
<td>C: Ch6</td>
</tr>
</tbody>
</table>
10/2  Lab 4: Chemical Equilibrium using Root-Finding Techniques

10/5  Matrices: Introduction to Matrix Algebra and the Matlab array data type
      [notes]
      C: Ch8

10/7  Matrices: Solutions to Systems of Equations, Inverse, and Stability
      [notes]
      C: Ch11

10/9  **Exam 1**: Introduction to programming and numerical methods, root finding

10/12 Matrices: Naive Gauss Elimination [notes]
      C: 9.1-9.2

10/14 Matrices: Gauss Elimination and Pivoting [notes]
      C: 9.3-9.5

10/16 Lab 5: Analysis of a Statically-Determinant Truss

10/19 Regression: Linear Least-Squares Regression [notes]
      C: 14.1-14.2

10/21 Regression: Function Linearization [notes]
      C: 14.3-14.5

10/23 Lab 6: Identifying the Sources of Particles in the Air

10/26 Regression: Goodness-of-Fit Measures [notes]
      C: 15.1-15.2

10/28 Interpolation: Linear and Non-Linear Interpolation Techniques [notes]
      C: 17.1, 17.3

10/30 Lab 7: Data Analysis applied to Toll Road Rate Data

11/2  Calculus: Numerical Integration [notes]
      C: 19.1-19.6

11/4  Calculus: Numerical Differentiation [notes]
      C: 21.1-21.4

11/6  Lab 7, continued

11/9  IVP: Euler's Method [notes]
      C: 22.1-22.3

11/11 IVP: Runge-Kutta Methods [notes]
      C: 22.4

11/13 **Exam 2**: Matrices, regression, interpolation, and numerical calculus

11/16 IVP: Systems of ODEs and Using the Built-in Matlab IVP Solvers [notes]
      C: 22.5 22.6

11/18 BVP: Shooting Method and Finite Difference [notes]
      C: Ch24

11/20 Lab 8: Earthquake Vibrations in a Multi-Story Building

11/23 Eigenvalues: General Properties [notes]

11/25 Reading Day

11/27 Thanksgiving Holiday

11/30 Eigenvalues: As Methods to Solve Boundary Value Problems [notes]

12/2  PDE: Generalization of Finite Difference Methods

12/4  Lab 9: Principle Stresses and Flow Nets in Geotechnical Design

http://ceprofs.civil.tamu.edu/assocolotsky/CVEN302/calendar.html
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/7</td>
<td>Lab 9, continued.</td>
</tr>
<tr>
<td>12/9</td>
<td>PDE: Matrix Solutions and Course Review</td>
</tr>
<tr>
<td>12/11</td>
<td><strong>Final Exam</strong>: Comprehensive</td>
</tr>
</tbody>
</table>
Assignments

Problem Sets

The following presents the written assignments for Fall 2015. You may also view assignments from Past Semesters for additional examples.

<table>
<thead>
<tr>
<th>Date out</th>
<th>Problems</th>
<th>Date due</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/31</td>
<td>HW1: Matlab Basics</td>
<td>9/2</td>
</tr>
<tr>
<td>9/2</td>
<td>HW2: Introduction to Programming</td>
<td>9/9</td>
</tr>
<tr>
<td>9/9</td>
<td>HW3: Loops and Branch Statements</td>
<td>9/16</td>
</tr>
<tr>
<td>9/16</td>
<td>HW4: Matlab User-Defined Functions</td>
<td>9/23</td>
</tr>
<tr>
<td>9/28</td>
<td>HW5: Root Finding</td>
<td>10/5</td>
</tr>
<tr>
<td>10/5</td>
<td>Review: Work unassigned problems from the Chapman textbook</td>
<td>--</td>
</tr>
<tr>
<td>10/12</td>
<td>HW6: Linear Algebra</td>
<td>10/19</td>
</tr>
<tr>
<td>10/19</td>
<td>HW7: Linear Least-Squares Regression</td>
<td>10/26</td>
</tr>
<tr>
<td>10/26</td>
<td>HW8: Interpolation and Data Modeling</td>
<td>11/2</td>
</tr>
<tr>
<td>11/2</td>
<td>HW9: Numerical Calculus</td>
<td>11/9</td>
</tr>
<tr>
<td>11/9</td>
<td>Review: Work unassigned problems from the Chapra textbook.</td>
<td>--</td>
</tr>
<tr>
<td>11/16</td>
<td>HW10: Numerical Solutions to IVPs</td>
<td>11/23</td>
</tr>
<tr>
<td>11/23</td>
<td>HW11: Boundary Value and Eigenvalue Problems</td>
<td>12/2</td>
</tr>
<tr>
<td>12/2</td>
<td>HW12: Initial Value and Eigenvalue Problems</td>
<td>12/5</td>
</tr>
</tbody>
</table>

Programming Assignments

The following presents the programming assignments for Fall 2015. You may also view programming assignments from Past Semesters for additional examples. You should report the results of your work using the CVEN 302 Memorandum Template.

<table>
<thead>
<tr>
<th>Date out</th>
<th>Problems</th>
<th>Date due</th>
</tr>
</thead>
<tbody>
<tr>
<td>9/2</td>
<td>Lab 1: Analysis of Historical Gas Price Data</td>
<td>9/11</td>
</tr>
<tr>
<td>9/9</td>
<td>Lab 2: Random Number Generation and Particle Diffusion</td>
<td>9/18</td>
</tr>
<tr>
<td>Date</td>
<td>Assignment</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>9/16</td>
<td>Lab 3: Bouncing Circles in a Closed Box</td>
<td></td>
</tr>
<tr>
<td>9/30</td>
<td>Lab 4: Chemical Equilibrium using Root-Finding Techniques</td>
<td></td>
</tr>
<tr>
<td>10/14</td>
<td>Lab 5: Analysis of a Statically-Determinant Truss</td>
<td></td>
</tr>
<tr>
<td>10/21</td>
<td>Lab 6: Identifying the Sources of Particles in the Air</td>
<td></td>
</tr>
<tr>
<td>10/28</td>
<td>Lab 7: Data Analysis Applied to Toll Road Rate Data</td>
<td></td>
</tr>
<tr>
<td>11/18</td>
<td>Lab 8: Earthquake Vibrations in a Multi-Story Building</td>
<td></td>
</tr>
<tr>
<td>12/2</td>
<td>Lab 9: Principle Stresses and Flow Nets in Geotechnical Design</td>
<td></td>
</tr>
<tr>
<td>10/2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Course Administration and Introduction**

*Scott A. Socolofsky*

CVEN 302

**INSTRUCTORS**

- **Instructor:** Dr. Scott A. Socolofsky
  - Office: CE/TTI 802B
  - Office Hours: MW 9:00 - 10:00, or by appointment via email
  - Email: socolofsky@tamu.edu
  - Phone: (979) 845-4517

- **TA:** TBA
  - Office:
  - Office Hours:
  - Email:
  - Phone:

**LABS AND OFFICES**

**ADMINISTRATION**

- In-Class Exercises require student computer access
  - Login using your TAMU NetID and password
  - Available to ANY student

- Be sure to check your TAMU email address
  - Only way I can contact students in course
  - Required by TAMU student rules

**COURSE OBJECTIVES**

- **Objective.** Students will learn to design and execute computer programming solutions to Civil Engineering problems using numerical methods

- **Learning Outcomes.**
  - Translate numerical methods into simple, reusable program modules
  - Choose appropriate numerical methods for solutions to specific mathematical problems
  - Analyze the applicability and accuracy of numerical solutions
  - Synthesize multiple program modules into larger program packages
  - Distill numerical results into a readable format that answers analysis and design questions

**COURSE DESCRIPTION**

- **Textbooks**
  - Chapman Matlab Programming for Engineers (5th edition)
  - Chapra Applied Numerical Methods (3rd edition) book

- Pre-requisites: ENGR 112, MATH 308 or co-registration, admission to CE degree

- Course website:
  - http://ceprofs.tamu.edu/ssocolofsky/cven302
  - Get homework, lecture notes, etc.
**ORGANIZATION**

- **Lectures:** MW. Mostly "chalkboard" style
- **Laboratory:** F. Dedicated to solving larger programming assignments
- **Homework:**
  - Problem Sets (10% of grade)
    - One per week
    - Grade best 10 out of 12
    - Solutions posted before due date
  - Programming Assignments (25% of grade)
    - One per week
    - Grade best 8 out of 9

**GRADING**

- Problem sets and quizzes: 10%
- Programming assignments: 20%
- Midterm Exam 1:
  - (October 9 in class)
  - 22%
- Midterm Exam 2:
  - (November 13 in class)
  - 23%
- Final Exam:
  - (December 11 at 10:00 am)
  - 25%

**GROUND RULES**

- **Homework**
  - "An Aggie does not lie, cheat, or steal, or tolerate those who do."
  - No sharing of computer programs or excerpts
  - Programs will be scanned for similarities
  - No late assignments will be accepted except for University excused absences
- **Exams**
  - Bring your own calculator
  - One page of notes
- **Excused Absences**
  - Give advance warning unless an emergency

**COMPUTERS**

**NUMERICAL METHODS**

\[
\frac{d^2}{dx^2} \left( EI \frac{du}{dx} \right) = w
\]

- Relates deflections \( u \) to applied load \( w \)
- For ideal situations, can be integrated analytically
- If:
  - \( I \) changed with distance?
  - \( E \) changed with distance?
  - \( W \) changed with time?
- How do they do it?
  - Use a software package
**NUMERICAL METHODS**

- Approximate equation by numerical derivatives:
  \[ \frac{du}{dx} = \frac{u(x + \Delta x) - u(x)}{\Delta x} \]
- Dissect beam into lots of parts
- Make tons of simple algebraic calculations -> Use a computer!

**NUMERICAL METHODS**

- Definition:
  "A numerical method is a complete and unambiguous set of procedures for the solution of a problem, together with computable error estimates. The study and implementation of such methods is the province of numerical analysis."
  - A Dictionary of Computing

**BUILDING BLOCKS**

- Root Finding
- Matrices
- Data Modeling
  - Regression
  - Interpolation
  - Statistics
- Calculus
  - Differentiation
  - Integration
- Differential Equations
  - Simulate the behavior of complex systems

**PROGRAMMING LANGUAGES**

- A set of instructions read and executed by a computer
- The most fundamental: machine language
- High-level compiled languages:
  - FORTRAN
  - BASIC
  - C
  - C++
- Higher-level scripting language:
  - Javascript
  - Perl, Python, Applescript
  - Matlab

**PROGRAM EXECUTION**

- Interactive environment – does not require formal compilation or linking/loading for execution
- Scripts – interpreter reads and executes m-files that contain Matlab commands

**MATLAB**

- Matrix Laboratory
- Widely used in science and engineering
- Relatively easy to learn
- High level programming language gets programs working quickly
- Powerful plotting, graphics and data manipulation
- GUI builder
MATLAB PROGRAMMING

- Intelligent text editor
- Built-in debugger
- Fully-functional programming language
- Extensive online help

MATLAB GRAPHICS

- Extensive suite of publication-quality plot types
- Easy automation using program scripts
- Full-service graphics package available to programs
- Industry standard for scientific and engineering plotting

Today's Lecture

- Introduction to Matlab programming environment
- Chapman MindTap site.
Tour of Matlab:

Open Matlab
Type in Command Window
Show Command History
  (Also arrow keys)
Show Current Directory
Show Workspace
Change Directory
Open edit window
  Type Comments
  Type Commands
Save and run.

Tour Continued:

Plot \( \sin(x) \) from 0 : 2\( \pi \).
Fundamental operators:
  =, +, -, *, /, ^
Basic Functions:
  \( \sin, \exp, \sqrt{\cdot} \)
Help: lookfor; menus
ck, clf, clear
who, whose, which
variables:
  array
  numbers; character string; logical

Demo:

\[
\begin{align*}
x &= 7 \\
y &= 2 \\
z &= \frac{x^2 y}{x^2 + y^2}
\end{align*}
\]
Tour Continued:

- Execute bouncing circles
- Show script in Editor
- Demo Help.

Question?

Reading:
- Ch. 1 & 2 in Chapman.
This is a Matlab script: This is a program in Matlab.

```matlab
x = 7;
y = 6;
z = x*y;  % supresses output
disp('Calculated values is:')
disp(z);
```

% Plot sin(x) from 0 to 2 pi
```
x = 0:0.01:2*pi;
y = sin(x);
plot(x,y);
axis([0 2*pi -1 1])
xlabel('x')
ylabel('y')
title('Plot of y = sin(x)')
print -dtiff -r300 myplot.tif
```

% Assignment Operator = : sets LHS to value computed on RHS
```
x = 7;
y = 8;
my_var = x*y/(x^2 + y^2) - 2*y;
```

% Variables must start with letter, are case sensitive, and should not be
% the same name as a built in function. To check names, type help and then
% the name you want to know.
```
help my_var
help sin
```

% Help works when you know the name you are looking for.
% Use lookfor if you're not sure of the name in matlab
```
lookfor 'square root'
```

% Hit ctrl-c to stop Matlab if it get's stuck
```
while(1)
disp('Oops...infinite loop')
end
```

% Variables can hold numbers
```
x = 2*pi;  % pi is built-in to Matlab
```

% arrays of numbers
```
x = 0:0.1:10;
x = [1 5 7; 2 5 9];
```

% characters
```
x = 'stop';
```

% or logical values
```
x = true;
```

% Matlab creates the correct variable and memory allocation when the
% assignment statement (=) is executed. There are other data types, but
% these are the most common in this course.