ME 3255 – Computational Mechanics

Prof. Ryan C. Cooper

ME 3255 – Computational Mechanics

- Introduction to scientific programming using Matlab/Octave
  - Best (Good?) practices for programming
  - Github (basic use)
  - Matlab/Octave functions
- Use tools to approximate solutions for:
  - Linear problems
  - Nonlinear problems
  - Differential equations

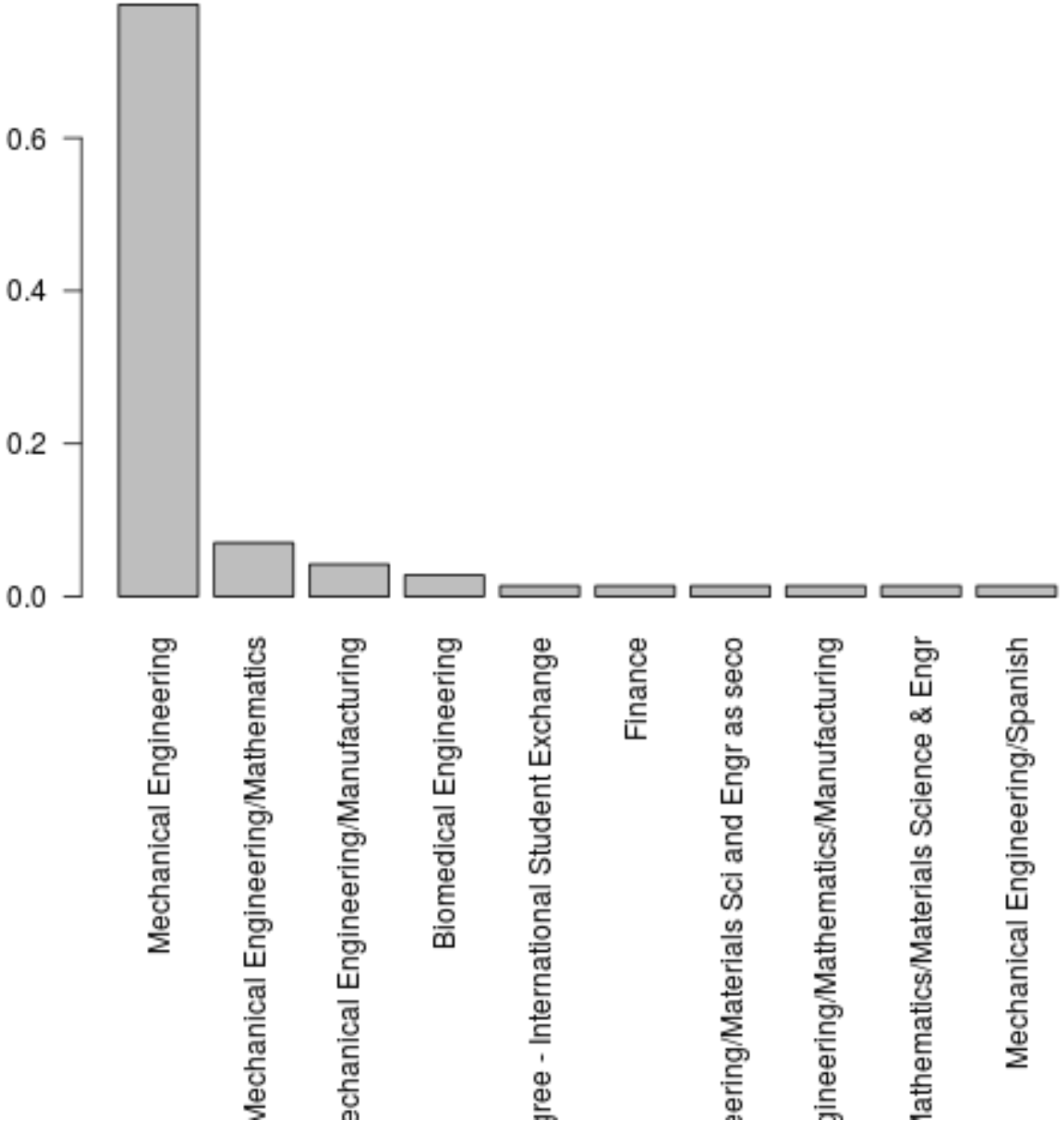
ME 3255 – Computational Mechanics

Grading

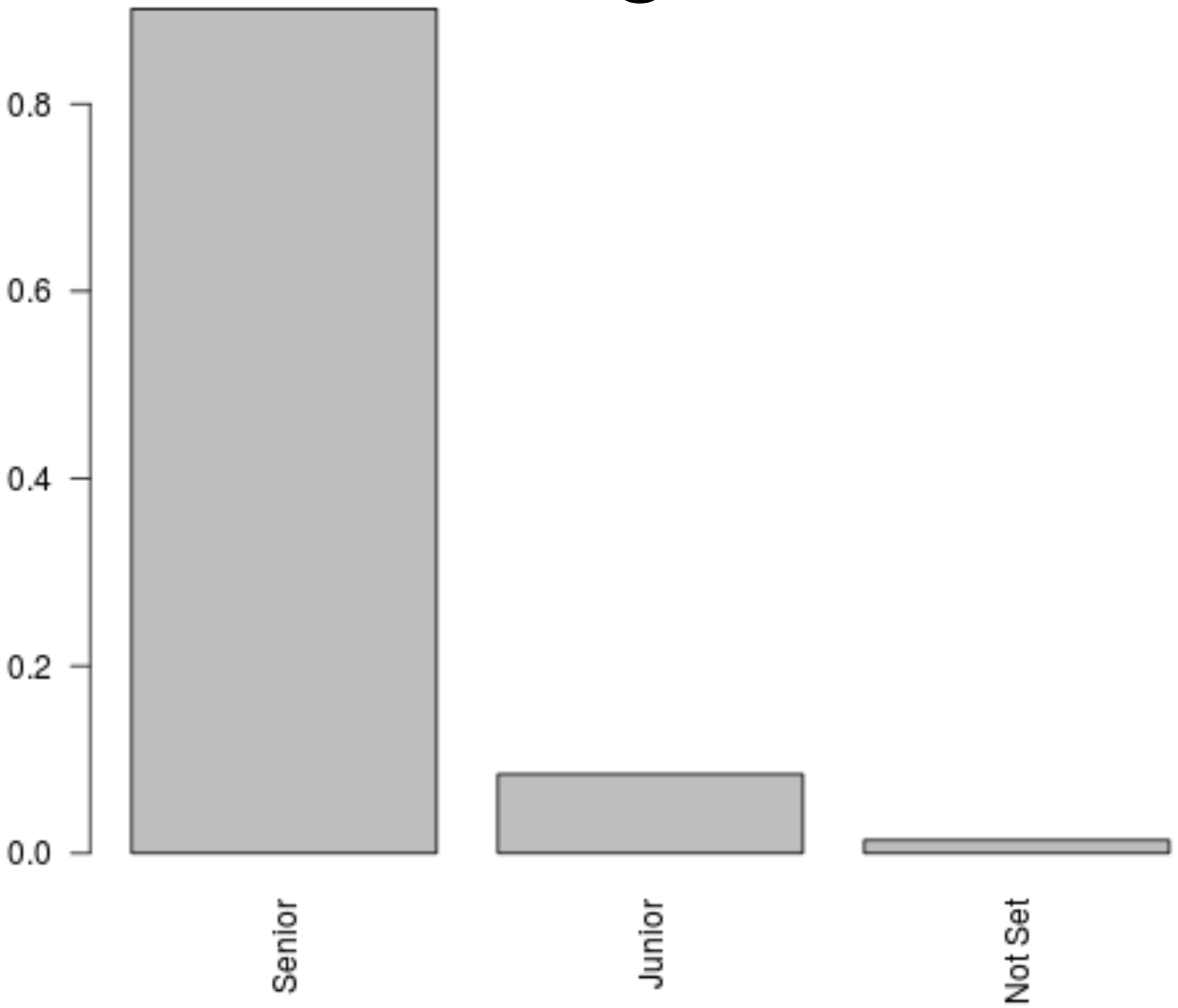
Item	Percent	Requirement
Homework	50 %	Turn in homeworks by assigned due date
Midterm Exam	10 %	One midterm exam
Final Project	30 %	A final project that will consist of code and documentation
Participation	10 %	During class online form will be sent out, you must submit form with your user ID to get credit

Who is taking this class?

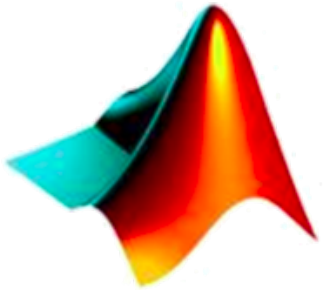
Who is taking this class?



Who is taking this class?



Why use Matlab/Octave?



MATLAB

GNU Octave

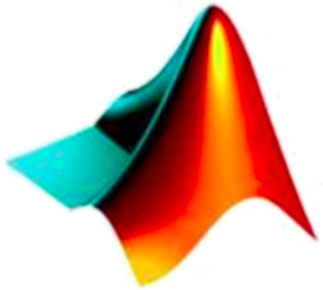
Scientific Programming Language

- Powerful mathematics-oriented syntax with built-in plotting and visualization tools
- Free software, runs on GNU/Linux, macOS, BSD, and Windows
- Drop-in compatible with many Matlab scripts

- Powerful set of tools for matrix and vector operations
- Excellent documentation
- User contributed code on mathworks.com/matlabcentral/fileexchange

- Free, open source software
  - If feature doesn't exist, you can make it yourself
- Most matlab functions available
- Documentation is great (but not as great as matlab)
- Can use most matlab files

Why use Matlab/Octave?



MATLAB

GNU Octave

Scientific Programming Language

- Powerful mathematics-oriented syntax with built-in plotting and visualization tools
- Free software, runs on GNU/Linux, macOS, BSD, and Windows
- Drop-in compatible with many Matlab scripts

- Powerful set of tools for matrix and vector operations
- Excellent documentation
- User contributed code on mathworks.com/matlabcentral/fileexchange

- Free, open source software
  - If feature doesn't exist, do it yourself
- Most matlab functions available
- Documentation is great (but not as great as matlab)
- Can use most matlab files

cost	$2650	$0

Why use Matlab/Octave?

- Matlab written in C/C++ and java
- Octave primarily written in C++
- Both provide quick access to efficient numerical algorithms
- Great for
  - Quick programming
  - Back of the envelope calculations
  - Data processing and visualization
  - Statistics and curve-fitting
- Not meant for
  - Complicated for and while loops
  - Multimillion degree of freedom problems
  - Optimized coding

Why use Matlab/Octave?

- Matlab and Octave are **high-level**, **interpreted** languages
- **High-level**: many details of human-computer interactions taken care of (in contrast, the lowest level language is machine code)

Example of machine code: A function in hexadecimal representation of 32-bit x86 machine code to calculate the nth Fibonacci number:

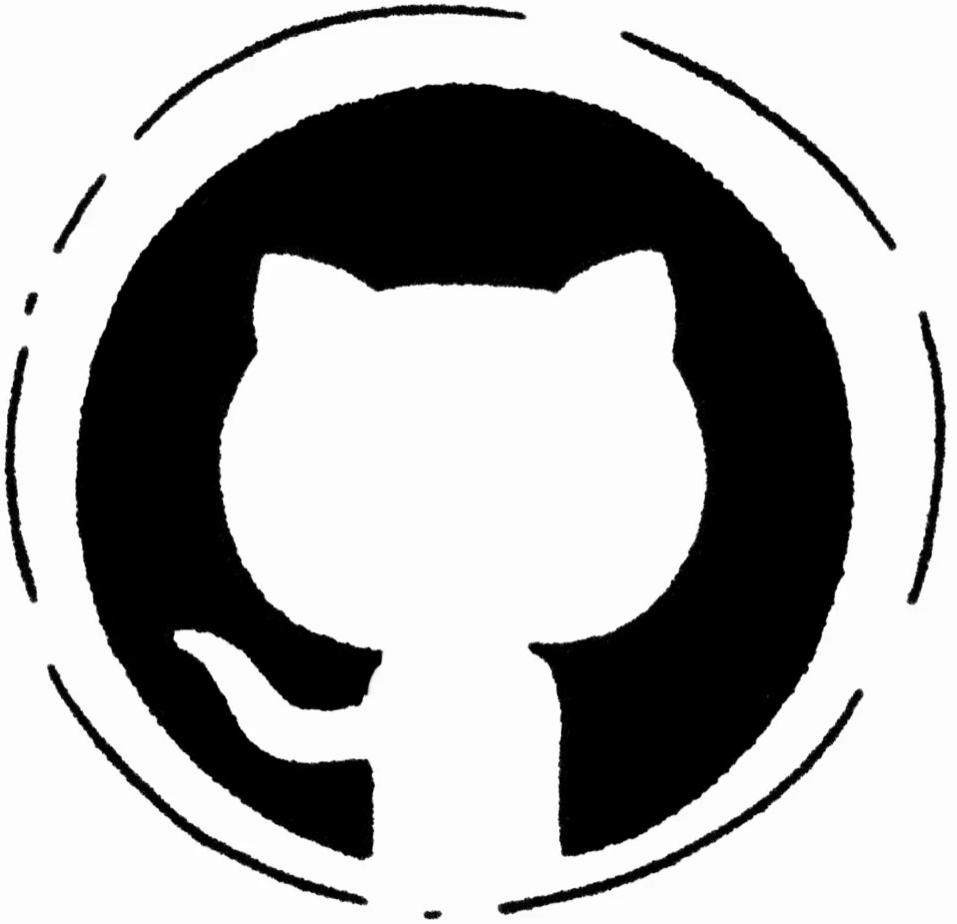
```
8B542408 83FA0077 06B80000 0000C383
FA027706 B8010000 00C353BB 01000000
B9010000 008D0419 83FA0376 078BD989
C14AEBF1 5BC3
```

- **Interpreted**: interpreted languages have precompiled functions that are called by the main program. This means, to run a matlab file `example.m` you need to first load matlab/octave then execute the program
  - Advantage: troubleshooting is much easier because you can call functions one at a time and pull up help files
  - Disadvantage: Lots of overhead (memory taken up by unused functions)

Why use Github?

- This class briefly introduces a number of numerical methods
  - These have been done and optimized by others
- Github allows access to thousands of open-source projects
  - Can use code in your own projects
  - Can contribute back to community
- Organizes your own code with good looking README.md files
- Creates an interactive website for your code

Why use Github?



What can we do with these tools?

Mathematical model relates dependent variables to independent variables, parameters, and forcing functions.

$$\text{dependent vars} = \text{f}\left(\text{independent vars},\ \text{parameters},\ \text{forcing}\right)$$

consider the velocity of a free-falling object:

$$\sum \bar{\text{F}} = \bar{\text{F}}_{\text{drag}} + \bar{\text{F}}_{\text{gravity}}$$

$$\bar{\text{a}} = \frac{\sum \bar{\text{F}}}{\text{m}}$$