Matlab for First-Year College Engineers

Alexander Haubold
School of Engineering and Applied Science
Columbia University, New York
ah297@columbia.edu
Overview

• Motivation
• Course Background
• Approach
  – Lectures 1-6
• Challenges
• Conclusion
Overview

- Motivation
- Course Background
- Approach
  - Lectures 1-6
- Challenges
- Conclusion
Motivation

• Matlab used as analysis tool in many engineering majors
• Popularity in upper-level courses
• Toolboxes with specialized functionality for various engineering fields
• Advantage over C, C++, Java:
  – Higher-level programming; scripting
  – Matrix acts as universal data structure
  – Wide availability of functions
Motivation

• Upper-level courses assume basic knowledge of Matlab

• But:
  – No introductory course for engineers
  – Computer science equivalent too narrow

• Therefore:
  – Include in first-year engineering design
  – Cover topics common to all majors
Overview

- Motivation
- Course Background
  - Approach
    - Lectures 1-6
- Challenges
- Conclusion
Course Background

• First-year engineering design course (E1102 at SEAS, Columbia University)
  – Emphasis on:
    • Basic engineering design skills
    • Professional development (ABET)
      (team interaction, communication skills within and outside of team, formulate problem, perform research, find solutions, produce drawings and/or prototypes, etc.)
    • Technical instruction on Matlab (computational analysis tool), Maya (3D design tool)
Course Background

• E1102 (cont.):
  – One-term project-based service-learning
  – >160 students/semester, 4 course sections
  – Mandatory for engineering students
  – 3 hour class/week (lecture + laboratory)

<table>
<thead>
<tr>
<th>Week</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lecture</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Professional Development and Project Management Lectures</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Laboratory</strong></td>
<td></td>
<td>MATLAB: Computational Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Maya: 3D Image Modeling</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Overview

- Motivation
- Course Background
  - Approach
    - Lectures 1-6
- Challenges
- Conclusion
Approach

• Overview of Matlab in 6 lectures
  – 2 lectures: matrices
  – 2 lectures: basic programming constructs
  – 2 lectures: data sets, plotting, statistics

• Regular homeworks

• Written (non-computerized) exam
### 6 Lectures

- Distribution of topics aimed at all majors

<table>
<thead>
<tr>
<th></th>
<th>Theory</th>
<th>Programming</th>
<th>Data analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Matlab User</td>
<td>Programming</td>
<td>Audio – theory</td>
</tr>
<tr>
<td></td>
<td>Interface</td>
<td>Programming</td>
<td>Programming</td>
</tr>
<tr>
<td></td>
<td>Vectors, Matrices</td>
<td>Conditionals</td>
<td>Programming</td>
</tr>
<tr>
<td></td>
<td>Theory</td>
<td>Functions</td>
<td>Functions</td>
</tr>
<tr>
<td></td>
<td>Images – Theory</td>
<td>Image analysis, synthesis</td>
<td>Image analysis, synthesis</td>
</tr>
<tr>
<td></td>
<td>Audio – Theory</td>
<td>Audio analysis, synthesis</td>
<td>Audio analysis, synthesis</td>
</tr>
<tr>
<td></td>
<td>Matlab User</td>
<td>Dataset, Meshing, Plotting,</td>
<td>Statistics</td>
</tr>
<tr>
<td></td>
<td>Interface</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applied Physics &amp; Mathematics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Biomedical Engineering</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chemical Engineering</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Civil Engineering</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Engineering</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Science</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electrical Engineering</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Earth and Environmental Engineering</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Industrial Engineering</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mechanical Engineering</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• Matlab user interface
  – Emphasis on Workspace
    • Our experience: underutilized by students to fix problems related to code evaluation
• Matrix data structure, notation
  – Most data in Matlab represented by matrices
  – Understanding it well is half the battle
• Common operations
  – Scalar arithmetic operations extend to vectors, matrices
Demonstrate Matlab’s built-in capabilities
- Availability of many high-level functions
- Toolboxes replace custom implementation

<table>
<thead>
<tr>
<th>C, C++, Java, etc.</th>
<th>Matlab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical Engineer / Computer Scientist</td>
<td>Engineer / First-year student</td>
</tr>
<tr>
<td>Goal: provide functionality and efficient implementation</td>
<td>Goal: make use of functionality</td>
</tr>
<tr>
<td>Focus on: JPEG compression theory, signal processing, discrete cosine transforms, bit/byte manipulation, data representation (RGB), etc.</td>
<td>Focus on: Load image, understand its representation, perform some analysis, etc.</td>
</tr>
</tbody>
</table>
Lecture 2

• Practical examples of matrix manipulation
  – Matrices: image processing
    • Matrix indexing:
      – Crop image = extract sub-matrices
    • Matrix arithmetic:
      – $\Delta$ brightness = scalar addition
      – $\Delta$ contrast = scalar multiplication
  – Vectors: audio processing
    • Vector indexing:
      – Cut = extract sub-vectors
    • Vector arithmetic:
      – $\Delta$ volume = scalar multiplication
Lecture 2

• Some quantitative estimates using popular media
• Image / video size vs. bandwidth

<table>
<thead>
<tr>
<th></th>
<th>Pixels</th>
<th>Size</th>
<th>Size in Mb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example matrix (lecture 1)</td>
<td>4 x 5</td>
<td>20</td>
<td>0.0</td>
</tr>
<tr>
<td>Small image (1990)</td>
<td>320 x 240 (x 3)</td>
<td>230,400</td>
<td>0.2</td>
</tr>
<tr>
<td>Computer display (2006)</td>
<td>1280 x 1024 (x 3)</td>
<td>3,932,160</td>
<td>3.7</td>
</tr>
<tr>
<td>DVD video (2006)</td>
<td>720 x 480 (x 3) x 30</td>
<td>31,104,000</td>
<td>29.7</td>
</tr>
<tr>
<td>Computer display refreshing at 40 times/sec</td>
<td>1280 x 1024 (x 3) x 40</td>
<td>157,286,400</td>
<td>150.0</td>
</tr>
</tbody>
</table>
• Introduction of programming constructs

• Loops
  – Loop variables
  – 1D – simulate vector-vector arithmetic
  – 2D – extend to matrix-matrix arithmetic

• Examples from sound domain
  – 1D – generate sound effect (raindrop)
    • Through loops: vary some property (frequency)
    • Through concatenation: collect sound samples
  – Opens discussion on basic signal processing
Lecture 3

• Examples from image domain
  – 1D – generate one-color gradient
    • Through loops: vary color component
  – 2D – extend 1D gradient
    • Through nested loop: vary 2nd color component

Raindrop 1D gradient 2D gradient
• Conditionals
  – Basic decision-making construct
  – if-else, if-then-else, if-else if-…-else

• Example from image domain
  – Reduce a high-color image to 8-color space
  – Combines loops and conditional constructs
Lecture 4

• Functions
  – Brief introduction
  – Emphasis:
    • Error checking (use of conditionals)
    • Proper help
    • Variable space
Lecture 5

- Datasets and plotting
  - Preparing, importing datasets
    - (unfortunately, not straightforward in Matlab)
  - Example dataset: gasoline prices
  - 2D and 3D plotting
    - Plots, bar charts, meshes, etc.
    - Proper labeling
Lecture 6

- Basic statistical measures
  - Mean, std, var, min, max, etc.
- Regression, function fitting
  - Example of future gas prices with n-degree polynomials
Overview

• Motivation
• Course Background
• Approach
  – Lectures 1-6
• Challenges
• Conclusion
Curriculum Challenges

• Lecture sequence changed significantly
• Originally: Physics
  – Lectures tied closely to course material
  – Theme: “Personal Transportation” (Segway)
  – Simulation of Segway physical properties
    • Observed forces, translation to battery power, etc.
  – But: Failed at Physics and Matlab levels
Curriculum Challenges

• Modified: Image and Statistics
  – Examples draw on familiar data
  – Somewhat “fun”, results visible and audible
Instructional Challenges

• Pace of instruction never right
  – For some to slow, for others too fast
  – Provide detailed lecture notes on-line

• Course focus on team-based service-learning projects
  – Application of Matlab not straightforward
  – Best match thus far: statistical analysis of project’s problem
  – However: students’ background limited
  – Data acquisition oftentimes difficult
Overview

• Motivation
• Course Background
• Approach
  – Lectures 1-6
• Challenges
• Conclusion
Conclusion

- Matlab for first-year engineers necessary
- Matlab required in regular coursework
- 6 lecture overview good introduction
- Student feedback mixed
  - Most acknowledge necessity
  - Most find material engaging and helpful

“Two great things about learning intro Matlab: 1) South Park and 2) Learning Matlab on an actual computer, and not paper and pencil. The South Park stuff was totally funny and appreciated by all. I actually retained some of this stuff. I wish that all Comp Sci classes were actually taught in a computer lab.”
Thank you!

Questions / Answers?