A0B17MTB – Matlab
Introduction

Miloslav Čapek
miloslav.capek@fel.cvut.cz
Filip Kozák, Viktor Adler, Pavel Valtr
Department of Electromagnetic Field
B2-626, Prague
You will learn ...

What is MATLAB?

Why to learn MATLAB?

Details of the AE0B17MTB course

Recommended literature, further resources

First steps in MATLAB
What is MATLAB?
MATLAB is...

- High-level programming language (*4th gener. language*)
- Interpreted language
  - intended mainly for numerical computing (nevertheless includes MuPAD symbolic kernel)
- philosophy: kernel + tool boxes + user-defined functions → wide application
  - wide possibilities of linking with other tools (Java, C++, .NET, Excel, physical-/multi-physical softwares)
- speed (of well written) algorithm comes near to that of C++
- excellent for „fast prototyping“
  - Matlab does not require variables declaration
- multi-license for CTU
  - Available for students as well!
  - *download.cvut.cz* + main access password
  - *fel.cvut.cz* → computer network → Multi-license software at CTU
Why to learn MATLAB?

- Matlab is a worldwide standard
- used by more than 5000 universities worldwide
- licenses used by thousands of corporations in aviation, biotechnology, electronics, cybernetics, mechanical engineering, finance, ...
- knowledge of Matlab can be used in other courses at the University as well as in professional life
Where to make use of Matlab?

- Data processing and visualization during laboratory exercises
- When elaborating diploma works
- Seminar exercises (signals, algorithm development, ...)
- Theory verification (mathematics and physics classes, electromagnetic field, electronic circuits, ...)

- Studying abroad (Erasmus, Sokrates)

⇒ “everywhere“ :)

Introduction
Historical development of MATLAB

- 70’s
  - Cleve Moler, Matlab used instead of Fortran
  - MATrix LABoratory → matice is the basic data structure
  - Fortran-based syntax
- 1983
  - Jack Little rewriting Matlab in C
  - new functionality and new mathematical libraries added
- 1984 (Matlab is so far for free!)
  - MathWorks founded in 1984
  - http://www.mathworks.com/
- 2004
  - Matlab used by more than 1 million of active users
- now...
  - ... R2015b is the newest version of Matlab
  - local distribution: Humusoft

see: http://www.mathworks.com/company/aboutus/founders/cleemoler.html
Alternatives to MATLAB

- **Fortran** – most of the libraries still in Fortran, used mostly by physisists
- **Python** – for free, fast and intuitive; Spyder provides MATLAB-like features
- **Mathematica** – symbolic and numerical calculations, excellent symbolic kernel, extensive applicability, mostly for mathematicians and physisists
- **Maple** – symbolic and numerical calculations
- **MathCad** – used for symbolic and numerical calculations, slightly out-of-date
- **Octave** – for free, syntax and functionality similar to Matlab, not so extensive, smartphone executable
- **R** – for free, designed particularly for statistical applications
- **Scilab** – Matlab-like, open documentation
- **Derive** – small, fast, Casio calculator executable
Alternatives to MATLAB

- Matlab vs. C/C++
  - optimal language strongly depends on the application
  - C/C++ faster in general, Matlab, on the other hand, provides implicit parallelism
  - general principle: Matlab more than suitable for everything except commercial compiled code (especially Matlab 6.5 and above: JIT + Real-Time Type Analysis)

- Matlab vs. Fortran
  - Matlab has wider support, more intuitive syntax
  - speed of a well written code is (usually, at least) comparable
  - utilization of Fortran is on the decline

- Matlab vs. Python
  - Matlab offers significant support thanks to MathWorks, Matlab File Exchange
  - Python entirely for free, it's becoming more and more popular
AE0B17MTB

- 13 weeks (14th week is a ‘reserve’)
  - 10 blocks with new theory, 1 block of bonuses, 1 block of examples
- conditions of credit award:
  - to hand in a project (next-to-last week of the semester)
    - competition assignment (see next slide)
  - to pass a test (min. 50%, next-to-last week)
    - on top of that two short tests during semester (min. 5 points)
    - 3 bonus examples during the semester
  - max. 2 missed classes (more absences only after prior arrangement)
    - any lecture can be substituted
- could happen that not all of the stuff of the course will be presented, because of time constraint – understanding the basics is a priority
  - bonus stuff (slides) available for advanced students

<table>
<thead>
<tr>
<th>Data types</th>
<th>Code execution</th>
<th>Visualization</th>
<th>Relation and logical operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix operations</td>
<td>User scripts and functions</td>
<td>Numerical methods</td>
<td>Symbolic math</td>
</tr>
</tbody>
</table>
Competition assignment

Effective display of the parametric surface

- see https://cw.fel.cvut.cz/wiki/courses/a0b17mtb/start > projects > seznam_projektu

- project can be selected by any number of students
- conditions:
  - project is completed according the assignment → credit award
  - project is the best one → winning the competition
    - prizes for the first three winners
## AE0B17MTB – Course syllabus

<table>
<thead>
<tr>
<th></th>
<th>Introduction, information on the course, MATLAB workspace, basic arithmetic operators, basic functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Complex numbers, complex matrix design, matrix operations, element-by-element operations, introduction to vectorization, matrix dimension</td>
</tr>
<tr>
<td>3</td>
<td>Indexation, data type and size, output format, MATLAB Editor, script design</td>
</tr>
<tr>
<td>4</td>
<td>Cycles, relation and logical operators, cycles vs. vectorization, control flow #1</td>
</tr>
<tr>
<td>5</td>
<td>Control flow #2, visualization in MATLAB #1, debugging #1</td>
</tr>
<tr>
<td>6</td>
<td>Set operations, sorting, searching, user-defined functions #1</td>
</tr>
<tr>
<td>7</td>
<td>User interface (main functions, subfunctions, nested functions, anonymous functions)</td>
</tr>
<tr>
<td>8</td>
<td>Strings, ‘eval’ and ‘feval’ functions, MATLAB path</td>
</tr>
<tr>
<td>9</td>
<td>Visualization in MATLAB #2, GUI #1</td>
</tr>
<tr>
<td>10</td>
<td>GUI #2</td>
</tr>
<tr>
<td>11</td>
<td>Date and time functions, error handling, cell, struct, I/O, basics of symbolic computations</td>
</tr>
<tr>
<td>12</td>
<td>MATLAB profile, p-code, numerical accuracy, publishing MATLAB code, programming style guidelines</td>
</tr>
<tr>
<td>13</td>
<td>Exercises, test</td>
</tr>
<tr>
<td>14</td>
<td>/reserve/</td>
</tr>
</tbody>
</table>
# AE0B17MTB – Deadlines

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>call for project proposals</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><em>bonus example</em> <em>(1-3 points)</em>, list of projects, discussion on own topics</td>
</tr>
<tr>
<td>4</td>
<td><em>short test</em> <em>(approx. 10-15 min)</em> aimed on solving given problem in Matlab, 10 points</td>
</tr>
<tr>
<td>5</td>
<td>project choice</td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><em>bonus example</em> <em>(1-3 points)</em></td>
</tr>
<tr>
<td>8</td>
<td><em>short test</em> <em>(approx. 10-15 min)</em> aimed on solving given problem in Matlab, 10 points</td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><em>bonus example</em> <em>(1-3 points)</em></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td><em>project hand-in</em> <em>(next-to-last week of the semester)</em>, <em>test</em></td>
</tr>
<tr>
<td>14</td>
<td>test evaluation, credit award</td>
</tr>
</tbody>
</table>
this is how the bonus slides look like...

harmonogram of WS 2015/2016:
AE0B17MTB – Principles

- the aim of the course is to teach you something – if the presentation is to fast, be heard

- if you have an idea / proposal how to solve a problem in a more efficient way, put it forward

- can happen that the lecturer is not able to answer your question immediately, in that case the answer will be provided during the next lecture
You will be able to...
Recommended literature, resources

- Matlab documentation

  >> doc % opens the help browser

- Basic web-based textbooks on Matlab (so called primers)

  - available at Department’s library

  - available at Department’s library

- and others ...
Launching Matlab

- command line
  - `matlab`

- Matlab can be launched with a set of optional parameters (see later)
  - `matlab -r "test(10)"`

- version dependent, up to 500MB RAM (win7) per matlab thread
The Matlab Environment ($\leq$ R2011b)

1. Command Window
2. Workspace
3. Command History
4. Current Folder
5. Select a file to view details
6. Current Folder path
7. Start
8. Ready
The Matlab Environment (≤ R2011b)

Introduction
The Matlab Environment – panels

1. Command Window (CTRL+0)
2. Workspace (CTRL+3)
3. Command History (CTRL+1) – not activated in case of $\geq$ R2015a; to activate...
4. Current Folder (CTRL+2)
5. Current Folder – Details
6. Current Folder (with history)
7. Start (Windows like), only for $\leq$ Matlab R2011b
8. status
Environment setting – basics

- Matlab R2012a and later
  - ribbon menu
- Matlab R2011b and older

- Font size

```
>> preferences
```
Matlab termination

- always terminate Matlab in the command window

```matlab
>> quit % terminates Matlab (and all windows)
>> exit % -/-/
```

- More advanced options (see documentation)

```matlab
>> quit cancel
>> exit force
```
Command line, documentation

`>> doc % opens documentation window`

`>> help % Matlab help`

`>> demo % tutorials`
The Help structure

>> help  % displays basic help contents
>> help sin  % displays help related to sine function
The Help structure, Matlab >R2011b

sin
Sine of argument in radians

Syntax

\[ Y = \sin(X) \]

Description

\( Y = \sin(X) \) returns the circular sine of the elements of \( X \). The \( \sin \) function operates element-wise on arrays. The function's domains and ranges include complex values. All angles are in radians.

Examples

Graph the sine function over the domain \(-\pi \leq x \leq \pi\).

\[
\begin{align*}
x &= -\pi:0.01:pi; \\
pplot(x,\sin(x)), & grid on
\end{align*}
\]
Matlab Help

- start and terminate Matlab
- set the Matlab environment to your taste

- try to launch the help
- find the documentation of the following functions: sin, cos, abs
- browse through individual help chapters
  - pay attention to the part Getting Started
# Shortcuts Command Window

<table>
<thead>
<tr>
<th>key</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTER</td>
<td>sends line for processing</td>
</tr>
<tr>
<td>ESC</td>
<td>deletes whole line</td>
</tr>
<tr>
<td>DEL</td>
<td>deletes one character (right to the cursor)</td>
</tr>
<tr>
<td>BACKSPACE</td>
<td>deletes one character (left to the cursor)</td>
</tr>
<tr>
<td>HOME</td>
<td>moves cursor to the beginning of line</td>
</tr>
<tr>
<td>END</td>
<td>moves cursor to the end of line</td>
</tr>
<tr>
<td>CTRL + ↑</td>
<td>moves cursor to the beginning of next word</td>
</tr>
<tr>
<td>CTRL + ↓</td>
<td>moves cursor to the beginning of previous word</td>
</tr>
<tr>
<td>SHIFT + ENTER</td>
<td>sends cursor to the next line</td>
</tr>
<tr>
<td>CTRL + K</td>
<td>deletes all to the right of cursor</td>
</tr>
<tr>
<td>CTRL + C</td>
<td>Forced interruption of Matlab (e.g. long / erroneous calculation)</td>
</tr>
<tr>
<td>CTRL + TAB</td>
<td>Switching between windows of Matlabu Environment</td>
</tr>
<tr>
<td>↓ a ↑</td>
<td>command history listing (searching is available CTRL+F)</td>
</tr>
<tr>
<td>F1</td>
<td>context help related to the word where the cursor is placed (Command Window, Editor)</td>
</tr>
<tr>
<td>TAB</td>
<td>function or variable name hint</td>
</tr>
</tbody>
</table>

+ usual Windows shortcuts for text processing
# Searching the Help

<table>
<thead>
<tr>
<th>key / command</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHIFT + F1</td>
<td>when pressed in command line, opens searchable function library</td>
</tr>
<tr>
<td>F9</td>
<td>evaluation of selected part of the code in Editor</td>
</tr>
<tr>
<td>NOT, OR, AND</td>
<td>it is possible to use logical operators in documentation search</td>
</tr>
<tr>
<td>*</td>
<td>it is possible to use wildcards in documentation search</td>
</tr>
<tr>
<td>&quot;&quot;</td>
<td>to search exact phrase in documentation</td>
</tr>
</tbody>
</table>

```
>> docsearch "plot tools"
```

```
>> docsearch plot* tools
```
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>quit, exit</td>
<td>terminates Matlab</td>
</tr>
<tr>
<td>preferences</td>
<td>opens Matlab preferences</td>
</tr>
<tr>
<td>doc, help, demo</td>
<td>commands related to documentation and help</td>
</tr>
<tr>
<td>sin, cos</td>
<td>sample goniometric functions</td>
</tr>
<tr>
<td>abs</td>
<td>absolute value</td>
</tr>
</tbody>
</table>
Thank you!

Apart from educational purposes at CTU, this document may be reproduced, stored or transmitted only with the prior permission of the authors. Document created as part of A0B17MTB course.