

# Lecture 1: MATLAB in a Nutshell

A8B17CAS

Miloslav Čapek

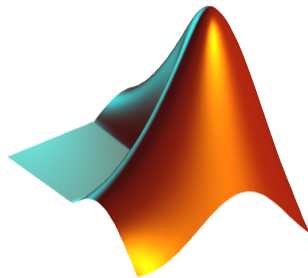
Department of Electromagnetic Field  
Czech Technical University in Prague  
Czech Republic  
`miloslav.capek@fel.cvut.cz`

September 23  
Winter semester 2024/25





1. What is MATLAB?
2. Launching and Termination
3. MATLAB Environment
4. Scalars, Vectors, Matrices
5. Basic Math Operations





# MATLAB is...

- ▶ High-level programming language (*4th generation language*).
- ▶ Interpreted language (not compiled, but... JIT).
  - ▶ 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++, Fortran, Python, .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 (not always the advantage).
- ▶ Multi-licensed for CTU.
  - ▶ Available for students as well!
    - ▶ CTU students: [download.cvut.cz](http://download.cvut.cz)
    - ▶ FEE students: [svti.fel.cvut.cz/cz/services/software/matlab.html](http://svti.fel.cvut.cz/cz/services/software/matlab.html)



# Historical Development of MATLAB

- ▶ the 1970's
  - ▶ Cleve Moler<sup>1</sup>, MATLAB used instead of Fortran.
  - ▶ MATrix LABoratory → matrix is the basic data structure.
  - ▶ Fortran-based syntax.
- ▶ 1983
  - ▶ Jack Little rewrote Matlab in C.
  - ▶ New functionality and new mathematical libraries added.
- ▶ 1984 (MATLAB is so far for free!)
  - ▶ MathWorks founded in 1984
- ▶ 2004
  - ▶ Matlab used by more than 1 million of active users.
- ▶ now...
  - ▶ R2022a is the newest version of Matlab.
  - ▶ local distribution: Humusoft s.r.o.

---

<sup>1</sup>see: <http://www.mathworks.com/company/aboutus/founders/clevemoler.html>



# Alternatives to MATLAB

**Fortran:** most of the libraries still in Fortran, used mostly by physicists

**Python:** for free, fast and intuitive; Spyder provides MATLAB-like features

**Julia:** very fast, still limited community

**Mathematica:** symbolic and numerical calculations, excellent symbolic kernel, extensive applicability, mostly for mathematicians and physicists

**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



# Launching MATLAB

- ▶ Desktop icon



- ▶ Command line:

- ▶ `matlab`

```

Příkazový řádek
Microsoft Windows [Version 10.0.17134.950]
(c) 2018 Microsoft Corporation. Všechna práva vyhrazena.
C:\>matlab_
  
```

- ▶ MATLAB can be launched with a set of optional parameters.

- ▶ `matlab -r "test(10)"`

- ▶ System requirements<sup>2</sup> for MATLAB R2022a+:
  - ▶ Windows 10+
  - ▶ 4 GB RAM
  - ▶ 3.1 GB of HDD (MATLAB only), 5-8 GB for a typical installation
  - ▶ Any Intel or AMD x86-64 processor
- ▶ Available also for Mac and Linux!

<sup>2</sup><https://www.mathworks.com/support/requirements/matlab-system-requirements.html>



# MATLAB Termination

- ▶ Close button in the top right of MATLAB window.



- ▶ Possibility to terminate MATLAB in the command window.

```
>> quit % terminates Matlab  
>> exit % -//-
```

- ▶ More advanced options (see documentation).

```
>> quit cancel  
>> exit force
```



# The MATLAB Environment

The screenshot shows the MATLAB R2019a desktop environment with the following components highlighted by red circles:

- 1**: Command Window (central area for running code)
- 2**: Workspace (table showing current variables)
- 3**: Command History (log of executed commands)
- 4**: File Explorer (left sidebar showing the current folder structure)
- 5**: Details pane (bottom left area for file details)
- 6**: Home tab of the ribbon (top left area with file operations)
- 7**: MATLAB logo (bottom left corner)
- 8**: Search box and Sign In button (top right corner)

The Command Window contains the following code:

```

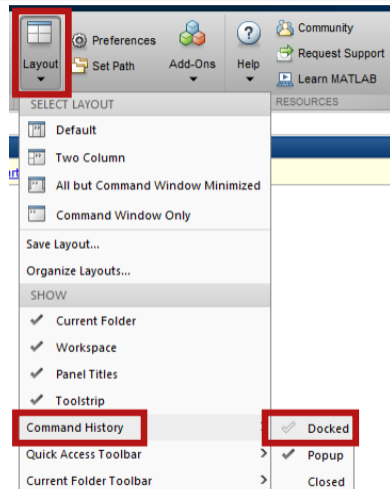
>>
a1 = pi/3;
B = [cos(a1) 0 -sin(a1);...
      0 1 0;...
      sin(a1) 0 cos(a1)];
v2 = B * v
clear
A = [1 1; 1 2]
b = [1 1]
x = A\b
b = [1;1]
A\b
b./A
x = A\b
x.' * A
(b.' / A) *
A\b
b -- 26.08.2019 15:44 -->
D:\preferences
clear
  
```





# The MATLAB Environment – Panels

1. Command Window
2. Workspace
3. Command History – *not activated, to activate* →
4. Current Folder
5. Current Folder – Details
6. Current Working Directory
7. Status (“Busy” when MATLAB is executing your code)
8. Search in documentation





# MATLAB Online

- ▶ [matlab.mathworks.com](https://matlab.mathworks.com)
- ▶ Runs in a web browser.
- ▶ Requires (CTU) log in.
- ▶ Slower than regular MATLAB.

```

MATLAB
https://matlab.mathworks.com
HOME PLOTS APPS
New Script New Live Script New Open Upload Go to File Import Data Save Workspace New Variable Open Variable Favorites Clear Commands Simulink Layout Parallel Add-Ons Help Feedback Community Learn MATLAB
FILE VARIABLE CODE SIMULINK ENVIRONMENT RESOURCES
MATLAB Drive
Current Folder
Name Git
final
Published (my site)
Workspace
Name Value Size Cl
a [1;1;1;1;1;1;1;1;1;1] 10x1 dou
>> a = ones(10, 1)
a =
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
>> |
  
```

## Documentation



```
>> doc % opens documentation window
```

```
>> doc sin % opens documentation of sin function
```

Documentation Home

Search Documentation

CONTENTS

All Examples Functions Blocks Apps

Explore Examples Explore Add-Ons

Category

MATLAB

Simulink

5G Toolboxes

Antenna Toolbox

Communications Toolbox

Control System Toolbox

Curve Fitting Toolbox

Data Acquisition Toolbox

Deep Learning HDL Toolbox

Deep Learning Toolbox

DSP System Toolbox

Fixed Point Designer

GPU Coder

Image Acquisition Toolbox

Image Processing Toolbox

LTE Toolbox

Mapping Toolbox

MATLAB Coder

MATLAB Compiler

MATLAB Compiler SDK

MATLAB Parallel Server

MATLAB Report Generator

Navigation Toolbox

Optimization Toolbox

Parallel Computing Toolbox

Partial Differential Equation Toolbox

Phased Array System Toolbox

Reinforcement Learning Toolbox

R2021b  
Release Notes

Explore MATLAB

Explore Simulink

View Installation Help

Applications

Math, Statistics, and Optimization

Data Science and Deep Learning

Signal Processing and Wireless Communications

Control Systems

Image Processing and Computer Vision

Parallel Computing

Test and Measurement

Application Deployment

Robotics and Autonomous Systems

FPGA, ASIC, and SoC Development

Code Generation

Database Access and Reporting

Systems Engineering

Documentation Home

Search R2021b Documentation

CONTENTS

All Examples Functions

Trial Software Product Updates

Documentation Home

MATLAB

Mathematics

Elementary Math

Trigonometry

sin

ON THIS PAGE

Syntax

Description

Examples

Input Arguments

Output Arguments

More About

Extended Capabilities

See Also

Sin

Sine of argument in radians

R2021b

Syntax

$Y = \sin(X)$

Description

$Y = \sin(X)$  returns the sine of the elements of  $X$ . The `sin` function operates element-wise on arrays. The function accepts both real and complex inputs.

Examples

- For real values of  $X$ , `sin(X)` returns real values in the interval  $[-1, 1]$ .
- For complex values of  $X$ , `sin(X)` returns complex values.

Examples

Plot Sine Function

Plot the sine function over the domain  $-\pi \leq x \leq \pi$ .

Open Live Script

```
x = -pi:0:pi;
plot(x, sin(x), 'g'); grid on
```

1

0.8

0.6

0.4

0.2



# MATLAB Commands

Matlab is **cAsE sEnSiTiVe!**

- ▶ Almost entirely, with certain exceptions (properties of graphics objects, ...).
- ▶ Pay attention to typos and variable names (see later).
  - ▶ New versions of MATLAB offer certain options.

```
a, A, b, c, x1, x2, M_12, test1, matrix_A, fx, fX
```

- ▶ Beware of different syntax in MATHEMATICA, *e.g.*,  $\sin(x)$  vs.  $\text{Sin}[x]$ , etc.
  - ▶ Following syntax is incorrect both in MATLAB and MATHEMATICA:

```
matrix A % contains space  
coef.a % possible only if coef is of type 'struct'
```

- ▶ Will be discussed in the next lectures.



# Naming Conventions

- ▶ Choose names corresponding to the meaning of each particular variable.
  - ▶ Letters and numbers are allowed, other symbols (colon “:”, hyphen “-” and others) are not.
  - ▶ Underscore is allowed in the variable name “\_” (not at the beginning, though!).



# Naming Conventions

- ▶ Choose names corresponding to the meaning of each particular variable.
  - ▶ Letters and numbers are allowed, other symbols (colon “:”, hyphen “-” and others) are not.
  - ▶ Underscore is allowed in the variable name “\_” (not at the beginning, though!).

## Recommendations:

- ▶ Lowercase letters in the names of scalars and variables (`a = 17.59;`).
- ▶ Matrix names usually start with a capital letter (`A = [ .. ];`).
- ▶ Iteration variables, variables used in `for` cycles usually named `m`, `n`, `k`, etc.
  - ▶ It is advisable to avoid `i` and `j` (complex unit).
- ▶ Avoid, if possible, standalone letter “`l`” (to be confused with one “`1`”) and predefined variables in MATLAB environment (see later).
- ▶ Avoid using names of existing functions or scripts (overloading can occur).
- ▶ The same conventions are valid for names of functions and scripts.



# Variable Names

- ▶ Examples of valid variable names:

```
a, A, b, c, x1, x2, M_12, test1, matrix_A, fx, fX
```

- ▶ Examples of invalid variable names:

```
1var      % starts with a number (not possible in MATLAB)  
matrix A  % contains space  
coef.a    % possible only if coef is of type 'struct'  
Test-1    % algebraic expressing: ans = Test - 1  
f(y)      % makes sense when using symbolic expressions
```

- ▶ Examples of valid numbers in MATLAB,

```
3, -66, +0.0015, .015, 1e2, 1.6025e-10, 05.1
```



# Workspace – Output Deletion

- ▶ To clean (erase) command window:

```
>> clc
```

- ▶ To clean one (or more) variable(s):

```
>> clear          % whole Workspace is deleted  
>> clear XX      % variable XX is deleted  
>> clear XX YY   % variables XX and YY are deleted  
>> clear z*      % everything starting with 'z' is deleted
```

- ▶ clear has a number of other options (graphics, I/O)





# Command History Window

- ▶ Command History window stores all commands from the Command Window.
- ▶ Command History is accessible though  $\uparrow$  or  $\downarrow$ .
- ▶ it is possible to filter out past commands by, *e.g.*:
  - » `A = [ +  $\uparrow$ .`
- ▶ It is possible to copy-and-paste entire Command History:  
`SHIFT / CTRL / CTRL + A  $\rightarrow$  CTRL + C.`



# Matrices in MATLAB

- ▶ Matrix is a basic data structure in MATLAB.
- ▶ There are the following variables' types depending on size:
  - ▶ scalar:  $1 \times 1$
  - ▶ vector:  $M \times 1$  or  $1 \times N$
  - ▶ matrix:  $M \times N$
  - ▶ array (multidimensional matrices):  
 $M \times N \times P \times Q \times R \times \dots$
- ▶ Matrices can be complex.
- ▶ It can contain text as well (beware of the length).

- ▶  $M$ -by- $N$  matrix:

$$\begin{array}{c}
 a_{i,j} \\
 \downarrow \\
 M \text{ rows} \\
 i \text{ changes}
 \end{array}
 \begin{array}{c}
 \xrightarrow{N \text{ columns}} \\
 j \text{ changes}
 \end{array}
 \begin{bmatrix}
 a_{1,1} & a_{1,2} & a_{1,3} & \dots \\
 a_{2,1} & a_{2,2} & a_{2,3} & \dots \\
 a_{3,1} & a_{3,2} & a_{3,3} & \dots \\
 a_{4,1} & a_{4,2} & a_{4,3} & \dots \\
 \vdots & \vdots & \vdots & \ddots
 \end{bmatrix}$$



# Matrix Creation

- ▶ Following techniques are available:
  - ▶ element-by-element entering (suitable for small matrices only),
  - ▶ colon notation “:” to define elements of series,
  - ▶ generation by built-in functions,
  - ▶ generation of matrices in m-files,
  - ▶ import and export from/to external files(.mat, .txt, .xls, ...).



# Matrix Construction Element-by-element I.

- Test the following commands to construct matrices by element enumeration.

```
>> a1 = -1
>> a2 = [-1] % brackets are redundant
```

```
>> v1 = [-1 0 1]
>> v2 = [-1; 0; 1]
```

```
>> M1 = [-1 0 1; -2 0 2]
>> M2 = [-1 -2; 0 0 ; 1 2]
>> M3 = [[-1 -2]; [0 0]] % inner brackets are redundant
```

- Suitable for small matrices only.

$$a_1 = a_2 = -1$$

$$\mathbf{v}_1 = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$\mathbf{v}_2 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{M}_1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \end{bmatrix}$$

$$\mathbf{M}_2 = \begin{bmatrix} -1 & -2 \\ 0 & 0 \\ 1 & 2 \end{bmatrix}$$

$$\mathbf{M}_3 = \begin{bmatrix} -1 & -2 \\ 0 & 0 \end{bmatrix}$$



# Matrix Construction Element-by-element II.

- ▶ Construct the following matrices:
  - ▶ Matrix values are defined inside square brackets [],
  - ▶ semicolon “;” separates individual rows of a matrix.

$$\mathbf{A} = \begin{bmatrix} -1 & 1 \\ 1 & -2 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$



# Matrix Construction

- ▶ Semicolon placed at the end of a command suppresses the display of the output in the Command Window.

- ▶ Note: it is possible to copy and paste code including “»”

```
>> a = 1  
>> b = 5;
```

- ▶ When there is more than one command on the same line, a comma is used to separate them.

```
>> a = 1, b = 5  
>> a = 1; b = 5;
```

- ▶ Row vs. column vector:

```
>> c = [1 0 0]  
>> d = [0; 0; 1]
```



# Basic Math Operators I.

- ▶ Operator types:
  - ▶ arithmetic:
    - ▶ matrix,
    - ▶ vector,
  - ▶ relational,
  - ▶ logical and other (to be mentioned later ...).
- ▶ Other operations using MATLAB functions:
  - ▶ complex conjugate,
  - ▶ sum, determinant, square root,
  - ▶ and hundreds of other functions ...

---

+	addition
-	subtraction
*	multiplication
^	power
.'	transpose
\	left matrix division
/	right matrix division
.	dot notation

---

Notice the operator's precedence (to be discussed later).

- ▶ see MATLAB → Language Fundamentals → Operators and Elementary Operations → Arithmetic



## Basic Math Operators II.

Type in the following commands:

- ▶ Zero can be omitted with a decimal number beginning with zero (not recommended).

```
>> a3 = -2/4  
>> a4 = -0.5  
>> a5 = -.5
```

- ▶ What is the difference between  $a_3$ ,  $a_4$  and  $a_5$ ?
- ▶ Beware the precedence of operators:

```
>> 3*5*6  
>> a1 = 15  
>> a2 = 10;  
>> a2/a3  
>> a2/a3*a4  
>> a2/(a3*a4)
```

- ▶ Explain the difference between  $a_2/a_3*a_4$  and  $a_2/(a_3/a_4)$ .





# Basic Math Functions I.

Math functions in MATLAB are generally divided in three groups:

▶ **Scalar**

- ▶ Function operates over individual elements of a matrix,
- ▶ *e.g.*: `sin`, `sqrt`, `log`, `factorial`.

▶ **Vector**

- ▶ Function operates over individual rows/columns of a matrix,
- ▶ *e.g.*: `sum`, `max`.

▶ **Matrix**

- ▶ Function operates over a whole matrix,
- ▶ *e.g.*: `det`, `trace`.



## Basic Math Functions II.

- ▶ Using MATLAB help, calculate the following expression:  $a \sin^2(\alpha) + a \cos^2(\alpha) - a$ 
  - ▶ Use numerical values your own choice.
  
- ▶ Verify following logarithmic identity:  $\log_{10}(a) + \log_{10}(b) - \log_{10}(ab) = 0$
  
- ▶ Find sum of all elements in individual rows of the following matrix:

$$T = \begin{bmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 6 & 7 & 8 & 9 \\ 0.2 & 0.3 & 0.4 & 0.5 \end{bmatrix}$$



# Basic Math Functions III.

- ▶ Assume following vectors  $\mathbf{u} = (1, 2, 3)$  and  $\mathbf{v} = (3, 2, 1)$ .

- ▶ Calculate:

$$\begin{array}{cc} \mathbf{u}\mathbf{v}^T & \mathbf{v}\mathbf{u}^T \\ \mathbf{v}^T\mathbf{u} & \mathbf{u}^T\mathbf{v} \end{array}$$

- ▶ Following function is needed:
    - ▶ `transpose (.')` of a vector/matrix,
  - ▶ What is the result of the above-mentioned operations?

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

$$\mathbf{A}^T = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$



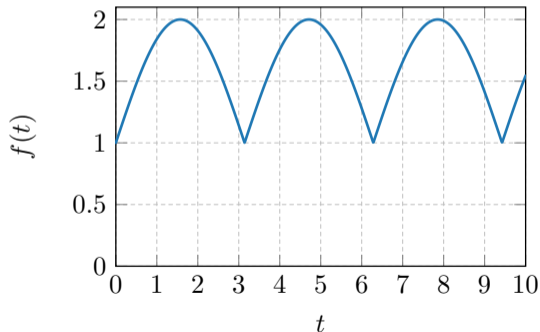
# Plot a Function

- ▶ Plot the following function in the interval  $t \in (0, 10)$ .
  - ▶ Choose your sampling of the axis  $t$ .

$$f(t) = |\sin(t)| + 1$$

```
% write your code here  
%
```

```
plot(t, f);  
ylim([0 3]);  
grid on;
```



# Questions?

A8B17CAS

`miloslav.capek@fel.cvut.cz`

September 23

Winter semester 2024/25

---

This document has been created as a part of A8B17CAS course.  
Apart from educational purposes at CTU in Prague, this document may be reproduced, stored, or transmitted only with the prior permission of the authors.