Lecture 1: Matlab Environment, Basic Math Operators

BE0B17MTB – Matlab

Miloslav Čapek, Viktor Adler, Michal Mašek, and Vít Losenický

Department of Electromagnetic Field
Czech Technical University in Prague
Czech Republic
matlab@elmag.org

September 23, 2020
Winter semester 2020/21

Outline

1. MATLAB Environment
2. Scalars, Vectors, Matrices
3. Basic Math Operations
4. Excercises

The MATLAB Environment

Introduction
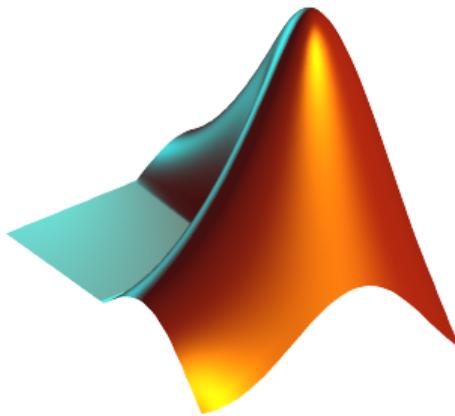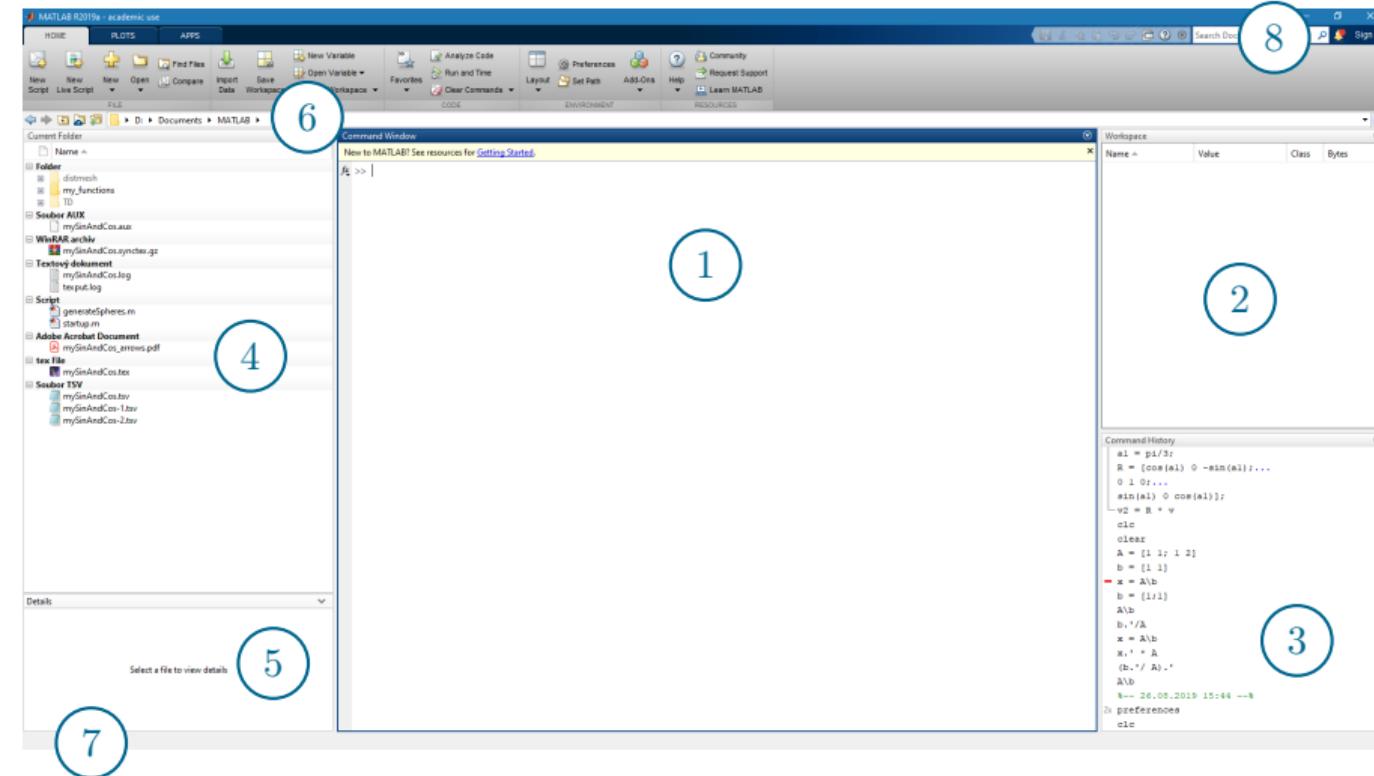
The MATLAB Environment – Panels

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
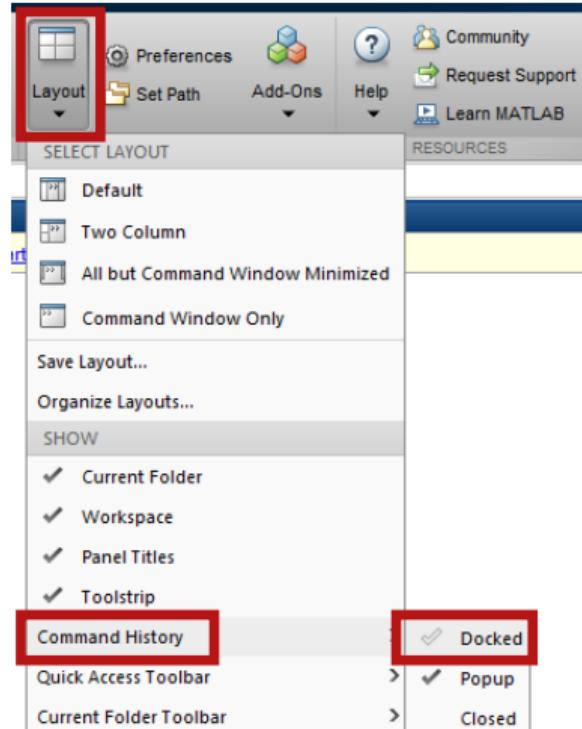
Preferences

▶ Command:

```
>> preferences
```

▶ Ribbon menu:
  ▶ Change font size.

Introduction
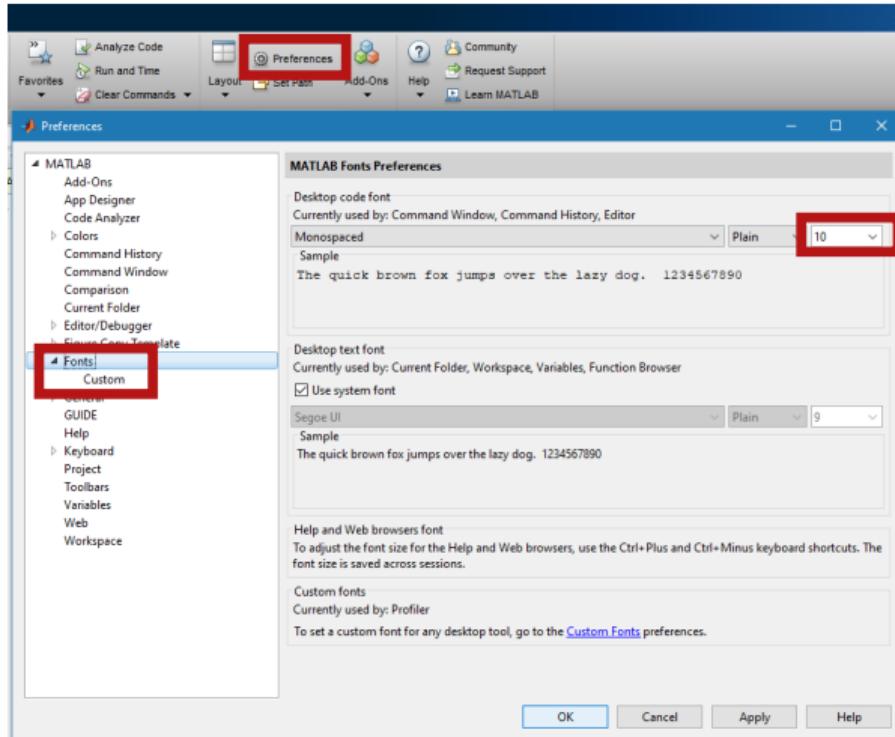
Documentation

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

```
>> help % MATLAB help
```

```
>> demo % tutorials
```

Introduction

The Help Structure

▶ Command:

```
>> help sin
```

▶ Output:

```
sin   Sine of argument in radians.
  sin(X) is the sine of the elements of X.

  See also asin, sind, sinpi.

  Reference page for sin
```

The Documentation Structure I.

▶ Command:

```
>> doc sin
```

1. Documentation page
2. Search field
3. Documentation contents
4. Bookmarks of this page

Introduction

The Documentation Structure II.

- ▶ Check the origin of the function.
  - ▶ Several functions with the same name may exist.
- ▶ Functions types by origin:
  - ▶ MATLAB core functions – most of them build-in, some are available for editing (not recommended!).
  - ▶ Functions from installed toolboxes.
  - ▶ User-created functions.
- ▶ Calling priority for functions will be discussed later.
- ▶ During this course, always open a function from core installation.

| sin| | 🔍 |

Functions	
_fx_ **sin** - _Sine of argument in radians_	MATLAB
_fx_ **sin** - _Symbolic sine function_	Symbolic Math Toolbox
_fx_ **sin** - _Sine of fixed-point values_	Fixed-Point Designer
_fx_ **sind** - _Sine of argument in degrees_	MATLAB
_fx_ **sinh** - _Hyperbolic sine of argument in radians_	MATLAB
» 136 more	

Introduction

Workspace Browser

- ▶ List of variables.
- ▶ Deleting/modification of existing variables.
- ▶ Saving/loading.
- ▶ Values, Class and Memory information.
  - ▶ Other information can be added: size, min, max, . . .
  - ▶ All information can be obtained using MATLAB functions that we learn later, *e.g.*, `min`, `max`, `max`, `length`.
- ▶ Fast data plotting option (in ribbon).

Introduction
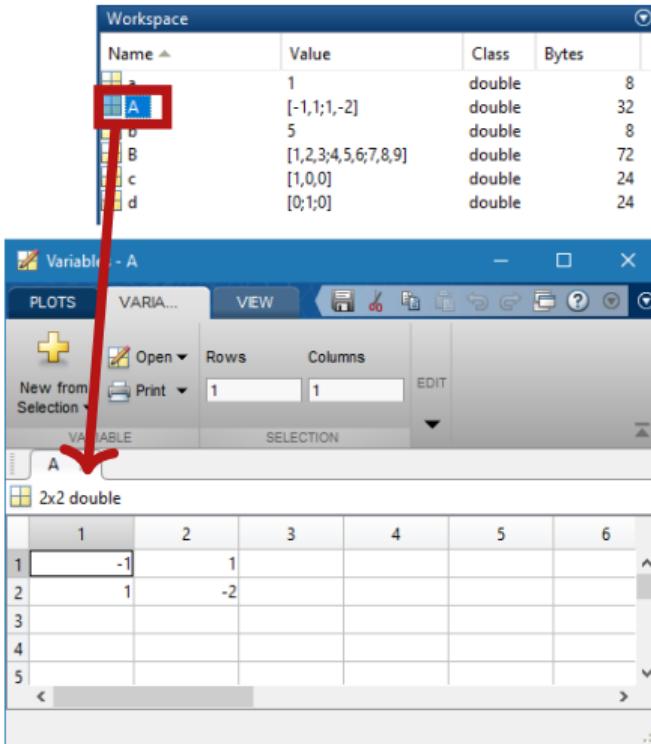
MATLAB Commands

▶ 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.

```
>> AA = [1 1 1]
>> Aa
```

▶ Beware of different syntax in Mathematica.

    ▶ Following syntax is incorrect both in MATLAB and Mathematica:

```
>> Sin(pi/2) % function names start with lower case
>> cos[pi/2] % function input is in parentheses ()
```

    ▶ Will be discussed in the next lectures.

Matrix Operations

Naming Conventions

- Names of variables can have max. 63 characters starting with letter
  (» `namelengthmax`)
  - 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!).
- 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).
- Chose the names to correspond to the purpose of the variable.
- Avoid, if possible, standalone letter "`l`" (to be confused with one "`1`") and predefined variables in MATLAB environment (see later).
- Choose names corresponding to the meaning of each particular variable.
- Avoid using names of existing functions or scripts (overloading can occur).
- The same conventions are valid for names of functions and scripts.

Program Flow

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
```

Data Types

Functions `who`, `whos`

- ▶ Function `who` lists all variables in MATLAB Workspace.
  - ▶ Wide variety of options.
- ▶ Functions `whos` lists the variable names + dimension, size and data type of the variables or displays content of a file.
  - ▶ Wide variety of options.

```
>> whos('-file', 'matlab.mat');
```

```
>> a = 15; b = true; c = 'test'; d = 1 + 5j;
>> who
>> whos
>> Ws = whos;
```

Data Types

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)

Program Flow

Command History Window

▶ Command History window stores all commands from the Command Window.

▶ Command History is accessible though ↑ or ↓.

▶ it is possible to filter out past commands by, *e.g.*:
  » A = [+ ↑.

▶ It is possible to copy-and-paste entire Command History:
  SHIFT / CTRL / CTRL + A → CTRL + C.

Program Flow

Matrices in MATLAB

▶ Matrix is a basic data structure in MATLAB.

▶ There are following variables types depending on size:
  ▶ scalar: 1×1
  ▶ 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 contains text as well (beware the length).

▶ M-by-N matrix:

$$
a_{i,j} \quad \xrightarrow{\substack{N \text{ columns} \\ j \text{ changes}}}
$$

$$
\substack{M \text{ rows} \\ i \text{ changes}} \Big\downarrow \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots \\ a_{4,1} & a_{4,2} & a_{4,3} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}
$$

Data Types
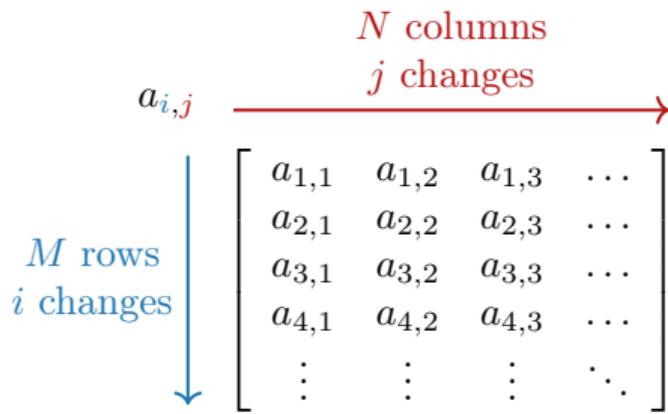
Matrix Creation

► 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`, ...).

Data Types

Matrix Construction Element-by-element I.

▶ Test following commands to construct matrices by element enumeration.

  ▶ Suitable for small matrices only.

```
>> a1 = -1
```

```
>> 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
```

$$a_1 = a_2 = -1$$

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

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

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

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

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

Data Types

Matrix Construction Element-by-element II.

► Construct following matrices:
  ► Matrix values are defined inside square brackets [ ],
  ► semicolon ";" separates individual rows of a matrix.

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

90

Matrix Construction

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

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

- When there is more than one command on the same line, comma is used to separate each of the commands.

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

  - Note: it is possible to copy and paste code including "»"

- Row vs column vector:

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

120

Basic Math Operators I.

- ▶ Operator types:
  - ▶ arithmetics:
    - ▶ 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
$\star$	multiplication
^	power
'	transpose
\	left matrix division
/	right matrix division
.	dot notation

Matrix Operations

Operator Precedence in MATLAB

- ▶ According to the following table:
  - ▶ see MATLAB → Language Fundamentals → Operators and Elementary Operations → Arithmetic

Matrix Operations

1	parentheses	( )					
2	transpose, power	'	.'	^	.^		
3	unary plus, unary minus, logical negation	+	−	~			
4	multiplication, division	*	.*	/	\	./	.\
5	addition, subtraction	+	−				
6	colon operator	:					
7	relation operators	<	>	<=	>=	==	~=
8	logical AND (element-wise)	&					
9	logical OR (element wise)	|					
10	logical AND (short-circuit)	&&					
11	logical OR (short-circuit)	||					

Basic Math Operators II.

- ▶ Type in 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 (wee see in the next slides):

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

- ▶ Explain the difference between a2/a3*a4 and a2/(a3/a4).
- ▶ Verify the rules of operator precedence from the previous slide.

200

Lengthy commands in MATLAB

▶ It is suitable to structure command blocks for clarity:
  ▶ next line: SHIFT + ENTER

```
>> A = [1 1 1]; B = [2 2 2]; % SHIFT + ENTER
C = [2 3 2];
```

▶ Three dots notation:
  ▶ For continuation of the same command on the next line.
  ▶ Compare results:

```
>> A1 = [ 1 1 ...
2 3]
```

```
>> A2 = [ 1 1
2 3]
```

120

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`.

Matrix Operations

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:

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

600

Basic Math Functions III.

- Assume following vectors $\mathbf{u} = (1, 2, 3)$ and $\mathbf{v} = (3, 2, 1)$.
  - Calculate:
    $$\begin{array}{ll} \mathbf{u}\mathbf{v}^{\mathrm{T}} & \mathbf{v}\mathbf{u}^{\mathrm{T}} \\ \mathbf{v}^{\mathrm{T}}\mathbf{u} & \mathbf{u}^{\mathrm{T}}\mathbf{v} \\ \mathbf{u} \cdot \mathbf{v} & \mathbf{u} \times \mathbf{v} \end{array}$$
  - Following functions are needed:
    - `transpose` (`.'`) of a matrix,
    - `dot` scalar product,
    - `cross` product.
  - What is the result of the above mentioned operations?

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

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

600

Matrix Division in MATLAB

- Two cases are distinguished:
    - left division (\ – mldivide),
    - right division (/ – mrdivide).
- Solution of a linear system of equations:
    - $\mathbf{A}$ is an invertible (regular) matrix,
    - $\mathbf{b}$ is a column (row) vector.

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

```
>> x = A \ b
```

$$\mathbf{xA} = \mathbf{b}$$

$$\mathbf{x} = \mathbf{b}\mathbf{A}^{-1}$$

```
>> x = b / A
```

Basic Math Functions IV.

▶ Find the sum of diagonal elements (trace of a matrix) of the matrix **T** with elements coming from normal distribution with mean equal to 10 and standard deviation equal to 4.

```
>> T = 10 + 4*randn(7, 7);
```

▶ Find determinant of matrix **U**.

$$\mathbf{U} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 2 & 0 \\ 0 & -2 & -1 \end{bmatrix}$$

```
>> U = [1 2 3; 0 2 0; ...
0 -2 -1];
```

▶ Solve the linear system of equations:

$$x_1 + 2x_2 + 3x_3 = 6$$
$$4x_1 + 5x_2 + 6x_3 = 15$$
$$7x_1 + 8x_2 + x_3 = 16$$

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$
$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

500

Predefined Values in MATLAB

- ▶ MATLAB contains several predefined values:
    - ▶ eps – precision of single/double numbers (Determines the shortest distance between two single/double numbers).
    - ▶ ans – *answer* – most recent answer.
    - ▶ NaN – *not a number* (every expression containing NaN is NaN)
        - ▶ NaN can be used advantageously in some cases.
    - ▶ Inf – *infinite number* (variable Inf can be used in calculation:))
        - ▶ Pay attention to Inf propagation throughout your code (use allowed operations only).
    - ▶ i, j – complex unit.
        - ▶ They are all basically functions (without input parameter).
    - ▶ Check results of the following expressions:

    ```
    >> t1 = 10/0    % t1 = Inf
    >> t2 = 0/0     % t2 = NaN
    >> t3 = t1*5    % t3 = Inf
    >> t4 = t1 + t2 % t4 = NaN
    ```

    - ▶ pi, intmin, intmax, realmin, realmax, . . . (functions)

Matrix Operations

Format of Command Line Output

- ▶ Up to now we have been using basic setup.
- ▶ MATLAB offers number of other formatting options
  - ▶ Use format style.
  - ▶ Output format does not change neither the computation accuracy nor the accuracy of stored results (eps, realmax, realmin, ... still apply).

style	format description
short	fixed 4 decimal points are displayed
long	15 decimal points for double precision, 7 decimal points for single precision
shortE	floating-point format (scientific notation)
longE	-//-
bank	two decimal points only (eur – cents)
rat	MATLAB attempts to display the results as a fraction
compact	suppressed the display of blank lines
and others...	note: omitting style parameter restores default setup

Program Flow

Format of Command Line Output

- ▶ Try following output format settings:
  - ▶ Each format is suitable for different type of problems.

```
>> s = [-5 1/2 1/3 10*pi sqrt(2)];
>> format long; s
>> format rat; s
>> format bank; s
>> format hex; s
>> format +; s
>> format; s
```

- ▶ There exist other formats with slight differences.
  - ▶ Check doc format
- ▶ Later, we will learn how to use formatted conversion into strings (commands sprintf and fprintf).

240

Complex Numbers I.

▶ More entry options in
  MATLAB.

```
>> C1 = 1 + 1j % prefered
>> C2 = 1 + 5i % prefered
>> C3 = 1 + 5*i % NO!
>> C4 = sqrt(-1)
>> C5 = complex(1, 2)
>> C6 = 1e1i
>> C7 = exp(1j*pi/4)
```

▶ Frequently used functions:

real, imag	real and imaginary part of a complex number
conj	complex conjugate
abs	absolute value of a complex number
angle	angle in complex plane [rad]
complex	constructs complex number from real and imaginary components
isreal	checks if the input is a complex number (more on that later)
i, j	complex unit
cplxpair	sort complex numbers into complex conjugate pairs

Matrix Operations

Complex Numbers II.

- Create complex number $z = 1 + 1\mathrm{j}$ and its complex conjugate $s = z^*$.
- Switch between Cartesian and polar form (find $|z|$ and φ).

$$z = \mathrm{Re}\{z\} + \mathrm{Im}\{z\} = a + \mathrm{j}b$$
$$z = |z|\,\mathrm{e}^{\mathrm{j}\varphi},\, |z| = \sqrt{a^2 + b^2}$$
$$z = |z|\,(\cos\varphi + \mathrm{j}\sin\varphi)$$

- Verify Moivre's theorem:

$$z^n = \left(|z|\,\mathrm{e}^{\mathrm{j}\varphi}\right)^n$$
$$z^n = |z|^n\,(\cos(n\varphi) + \mathrm{j}\sin(n\varphi))$$

300
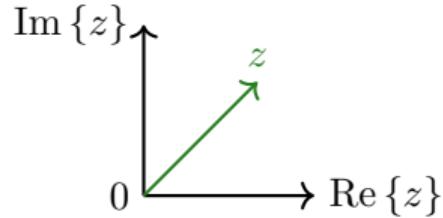
Exercises

# Exercise I.

▶ Following forces were localized at point **P** in xy plane:

$$\mathbf{F}_1 = [2, 2] \quad \mathbf{F}_3 = [2, 0]$$
$$\mathbf{F}_2 = [1, -3] \quad \mathbf{F}_4 = [2, -1.5]$$

 ▶ What is the direction of the resultant force **F**?

▶ Normalize the resulting vector.

$$\mathbf{n}_{\mathrm{F}} = \frac{\mathbf{F}}{|\mathbf{F}|} = \frac{\mathbf{F}}{\sqrt{F_x^2 + F_y^2 + F_z^2}}$$

180
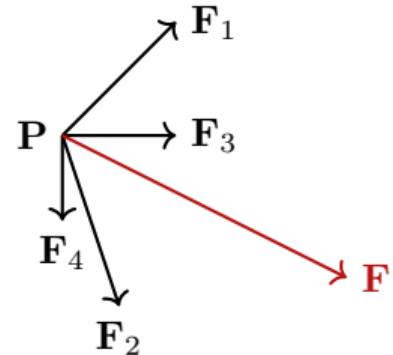
Exercise II.

▶ Type-in following commands:

```
>> clear, clc;
>> w1 = [1 2 3 4]
>> w2 = [-2 -3 -4]
>> w3 = [-2; -3; -4]
>> w4 = w1^2, w5 = w2 - w1
```

  ▶ Compare differences.
  ▶ What is the cause of error in calculation of w4 and w5?

▶ Try also:

```
>> w3*3, w1 - 3
>> w1 + [5 5 5 5]
>> w6 = 5*w1 - [3 5 6] - w2
```

▶ Calculate the norm (magnitude) of vector w1.
  ▶ Try more options.

  ▶ How to modify the calculation in the case of a complex vector?

240

Exercise III.

▶ Calculate roots of the quadratic function:

$$-2x^2 - 5x = 3.$$

▶ First, rearrange the terms of the function.

$$2x^2 + 5x + 3 = 0 \implies a = 2, b = 5, c = 3$$
$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-5 \pm \sqrt{25 - 24}}{4}$$
$$x_1 = -1, \ x_2 = -\tfrac{3}{2}$$

▶ MATLAB provides particular function for calculation of roots a function, try to search it out.

180

Exercise IV.

- ▶ Think over how many ways there are to calculate the length of hypotenuse when two legs of a triangle are given.
  - ▶ Make use of various MATLAB operators and functions.
  - ▶ Consider also the case where the legs are complex numbers.

Exercise V.

▶ Create an arbitrary vector **v** and rotate it around arbitrary angle α in xz plane using rotation matrix **R**.

$$\mathbf{v}' = \mathbf{R}\mathbf{v}$$

$$\mathbf{R} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix}$$

200

Exercise VI.

▶ Use the following code and round the resulting number to:

```
>> r = 1 + 10*rand(1)
```

   ▶ nearest integer,
   ▶ nearest integer greater than r,
   ▶ nearest integer lower than r,
   ▶ zero,
   ▶ zero with precision of 2 decimal digits.
▶ Find remainder after r is divided by 0.1.
   ▶ *modulus* vs. *remainder after division*

300

Exercise VII.

▶ Find out the magnitude of a complex vector (avoid indexing).

  ▶ Use `abs` and `sqrt`.

$$\mathbf{Z} = \left[\begin{array}{cc} 1 + 1\mathrm{j} & \sqrt{2} \end{array} \right]$$

$$\|\mathbf{Z}\| = ?, \quad \mathbf{Z} \in \mathbb{C}^2$$

▶ Alternatively, use following functions:

  ▶ `norm`
  ▶ `dot` (*dot product*)
  ▶ `hypot` (*hypotenuse*)
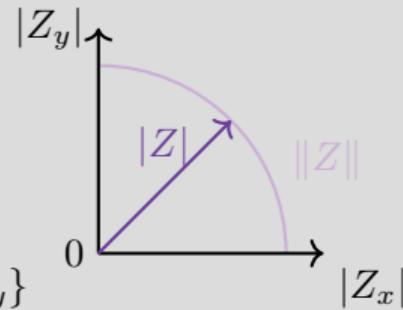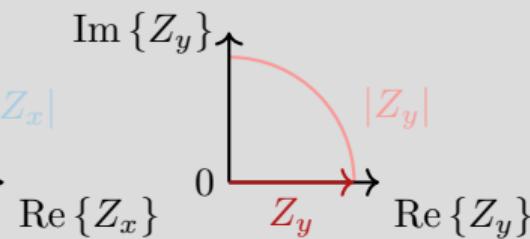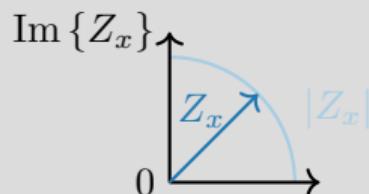
Questions?

BE0B17MTB – Matlab
matlab@elmag.org

September 23, 2020
Winter semester 2020/21