TECHNISCHE UNIVERSITÄT DRESDEN

Introduction to Matlab

Basic Operations & Matrices

Pouyan R. Fard
Prof. Dr. Stefan J. Kiebel

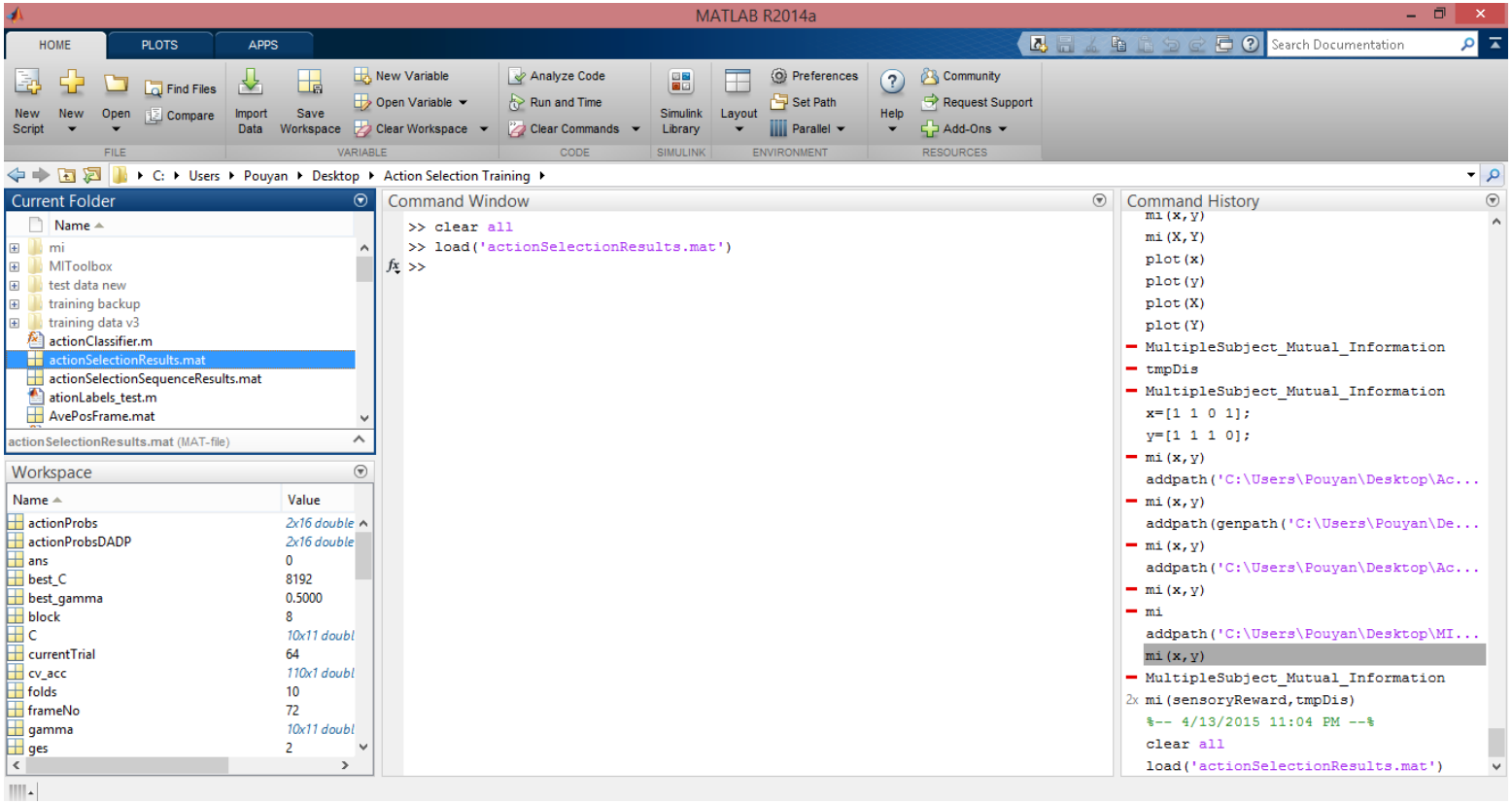
Dresden, 17.04.2015

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Today's Plan

Date	Topics	Projects
17.04.	Intro, basic operations, matrices	
24.04.	Data handling, random numbers, basic plotting	1st Project Assignment
01.05.	Holiday (Labour day)	
08.05	Advanced plotting, scripts, control flow 1st Project Presentation	1st Project Deadline 2nd Project Assignment
15.05.	Control flow statements, signal processing,	
22.05.	Functions, integration, image, and sound	
29.05.	Holiday (Pfingstferien)	
05.06.	Data Analysis, statistics, 2nd Project Presentation	2nd Project Deadline

MATLAB Environment



Basic Operators

- MATLAB as a scientific calculator

To TYPE after prompt >> followed by Enter	MATLAB answer	Meaning of the operation
`35*12`	`ans = 420`	Multiplication
`2/45`	`ans = 0.0444`	Division
`4-1`	`ans = 3`	Subtraction
`2^3`	`ans = 8`	Exponentiation

To TYPE after prompt >> followed by Enter	MATLAB answer	Meaning of the answer
`12/0`	`ans = Inf`	You should not divide by zero, but if you do, the result is **Infinity**
`0/0`	`ans = NaN`	Unable to find the answer, so the result is **NaN = Not a Number**

Basic Operators

- MATLAB as a scientific calculator

To TYPE after prompt >> followed by Enter	MATLAB answer	Meaning of the operation
`cos(12)`	`ans = 0.8439`	Cosine of the element in parentheses
`sin(12)`	`ans = -0.5366`	Sine of the element in parentheses
`tan(4)`	`ans = 1.1578`	Tangent of the element in parentheses
`exp(3)`	`ans = 20.0855`	Exponential of the element in parentheses
`log(10)`	`ans = 2.3026`	Natural logarithm of the element in parentheses
`log10(12)`	`ans = 1.0792`	Base-10 logarithm of the element in parentheses

- Using parentheses will disambiguate the statements, but you need to keep track of the order of your parentheses.

```
>> 5 + 3*8
ans = 1
    29
>> (5 + 3)*8
ans = 1
    64
```

Operators Precedence

- **Parentheses ()**
- **Power (^)**
- **Unary plus (+), unary minus (-)**
- **Multiplication (*), division (/)**
- **Addition (+), subtraction (-)**

- Example:

```
>> (2+(5*3/(7-5)^2)/3)          <ENTER>
ans =
    3.2500
```

Variables

- MATLAB is much more than a scientific calculator!
- You can define and use variables in MATLAB.
- These variables can be used to perform more complex calculations that can help us to implement different algorithms.

```
>> age          <ENTER>
age=
      22


>> Nfriends = 132          <ENTER>
Nfriends =
   132
```

Variables

```
>> whos          <ENTER>

Name        Size   Bytes  Class    Attributes
age         1×1        8  double
Nfriends    1×1        8  double

>> clear all                    <ENTER>
>> whos                         <ENTER>
>>
```

Variables and Operators

- All of the basic MATLAB operators that you learned can be applied on variable, too.

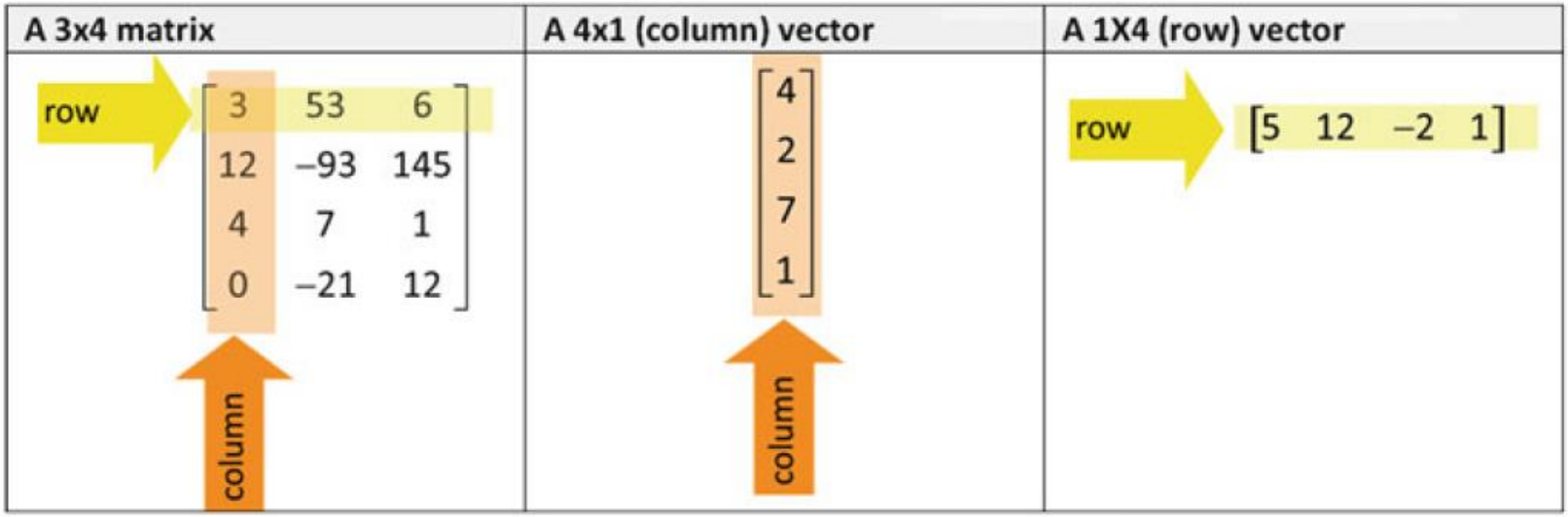
```
>> number=13;                    <ENTER>
>> a=14;                         <ENTER>
>> c=pi*((number+a/2)/10);       <ENTER>
```

- Showing the values of each variable:

```
>> number                        <ENTER>
number =
    13
>> c                             <ENTER>
c =
   6.2832
```

Matrices and Vectors

- Thinking in a matrix way



Matrices and Vectors

- Initializing vectors and matrices in MATLAB

```
>> a=[3,5,7,8]                                    <ENTER>
a =
     5    12    -2     1
>> b=[4;2;7;1]                                    <ENTER>
b =
     4
     2
     7
     1
>> c=[3, 53,6;12,-93,145;4,7,1;0,-21,12]          <ENTER>
c =
     3    53     6
    12   -93   145
     4     7     1
     0   -21    12
```

Matrices and Vectors

- Size of matrices

```
>> size(c)
ans =
     4    3
```

```
>> length(c)
ans=
     4
```

- Usage of spaces, commas, and semi-colons

```
>> x=[ 1 2 3; 2 5 7]
x =
     1     2     3
     2     5     7
```

- Dimensions must be consistent.

```
>> x = [2 3; 2 5 7];
??? Error using ==> vertcat
CAT arguments dimensions are not consistent.
```

Matrices and Vectors

Mathematical representation	MATLAB (type after the prompt >> followed by Enter)	Dimension
$M = \begin{bmatrix} 3 & 12 & \pi \end{bmatrix}$	`M=[3,12,pi];`	1×3 Row vector

$N = \begin{bmatrix} 3 & 12 & \pi \\ 8 & 9 & 10 \end{bmatrix}$	`N=[3,12,pi; 8,9,10];`
	Or equivalently, if you have already inserted M:
	`N=[M; 8,9,10];`
		2×3 Matrix

$P = \begin{bmatrix} 4 \\ 2 \\ -1 \end{bmatrix}$	P=[4;2;–1]	3×1 Column vector
$Q = \begin{bmatrix} 4 & -4 \\ 2 & -2 \\ -1 & 1 \end{bmatrix}$	Q=[4, –4;2, –2;–1, 1]; Or equivalently, if you have already inserted P: Q=[P;–P];	3×2 Matrix

Initializing Matrices

- Initializing an empty matrix

```
>> y = [ ];                    <ENTER>
>> whos y                      <ENTER>
Name        Size        Bytes      Class      Attributes

y           0x0         0                     double
```

- Initializing an identity matrix with size n — **eye(*n*)**
- Initializing a matrix whose all elements are 1 — **ones(*n*,*m*)**
  - n= number of rows , m= number of columns
- Initializing a matrix whose all elements are 0 — **zeros(*n*,*m*)**
  - n= number of rows , m= number of columns

Indexing

Start:Step:Stop

Type the following commands:

To TYPE after prompt >> followed by Enter	MATLAB answer	Meaning of the operation
2:5:25	ans = 2 7 12 17 22	Generate a vector going from 2 to 25 incremented by 5. Note that 22+5=27, which is greater than 25. MATLAB will generate numbers until it reaches or exceeds the Stop value (i.e., 25)
i:j	ans = 2 3 4	Generate a vector going from 2 to 4. Here the step value is not specified, and MATLAB uses the default value 1
10:-3:-5	ans = 10 7 4 1 -2 -5	Generate a vector going from 10 to –5, increasing the first value by –5. This is equivalent to generating a vector of decreasing values

Indexing

- Accessing single elements in matrices

```
>> Q(3,2)                    <ENTER>
ans =
     1
```

```
>> Q([1,3],2)                <ENTER>
ans =
    -4
     1
```

- Accessing multiple elements in matrices

```
>> x=[1 2 3; 4 5 6; 7 8 9; 10 11 12; 13 14 15]    <ENTER>
```

```
>> i=2; j=4;                 <ENTER>
>> x(i:j,2)                  <ENTER>
ans =
     5
     8
    11
```

Indexing

- Accessing multiple elements in matrices

```
>> x(3,1:3)                    <ENTER>
ans =
     7    8    9
>> x(3,:)                      <ENTER>
ans =
     7    8    9
```

- Deleting rows or columns of a matrix

```
>> x(:,2) = [ ]                <ENTER>
x =
     1     3
     4     6
     7     9
    10    12
    13    15
```

Indexing

- Deleting rows or columns of a matrix

```
>> x([3,4],:) =[ ]          <ENTER>
x =
     1     3
     4     6
    13    15
```

- Deleting one single entry in a matrix is not possible!

```
>> x(1,2)=[ ]
??? Subscripted assignment dimension mismatch.
```

- Deleting one single entry is possible in vector.

Matrix Operations

Operation	Definition	Math example	Matlab example

Addition (subtraction)	The result of $\mathbf{A}+\mathbf{B}$ or $(\mathbf{A}-\mathbf{B})$ is calculated entrywise, i.e., the element $B_{i,j}$ is added to (subtracted from) the element in $A_{i,j}$	$A=\begin{bmatrix}1 & 5\\ 2 & 3\end{bmatrix}, B=\begin{bmatrix}2 & 3\\ 4 & 1\end{bmatrix}$
		$A+B=\begin{bmatrix}3 & 8\\ 6 & 4\end{bmatrix}$
		$A-B=\begin{bmatrix}-1 & 2\\ -2 & 2\end{bmatrix}$
			`>>A=[1,5;2,3];`
			`>>B=[2,3;4,1];`
			`>>A+B`
			`ans =`
			`3 8`
			`6 4`
			`>> A-B`
			`ans =`
			`-1 2`
			`-2 2`

Scalar multiplication	The multiplication of a scalar (= number) s by a matrix $\mathbf{C}$ is obtained by multiplying every entry of $\mathbf{C}$ by s	$C=\begin{bmatrix}3 & 2\\ 4 & 1\end{bmatrix}, s=4$
		$s\cdot C=\begin{bmatrix}12 & 8\\ 16 & 4\end{bmatrix}$
			`>>C=[3,2;4,1];`
			`>>s=4;`
			`>>s*C`
			`ans =`
			`12 8`
			`16 4`

Transposition	The transpose of an $m\times n$ matrix $\mathbf{D}$ is an $n\times m$ matrix denoted by $\mathbf{D}^T$ obtained by turning rows into columns and columns into rows	$D=\begin{bmatrix}3 & 12 & 2\\ 8 & 9 & 10\end{bmatrix}$
		$D^T=\begin{bmatrix}3 & 8\\ 12 & 9\\ 2 & 10\end{bmatrix}$
			`>> D=[2,12,2;8,9,10];`
			`>> D'`
			`ans =`
			`3 8`
			`12 9`
			`2 10`

Matrix Operations

- Element-wise addition with a single-element matrix

```
>> p = [1   2;   3   4]
p =
   1   2
   3   4
>> p = p + 2
p =
   3   4
   5   6
```

- When dimensions don't agree:

```
>> r = [2   1; 1   1; 1   1]
r =
   2   1
   1   1
   1   1
>> n = p + r
??? Error using ==> plus
Matrix dimensions must agree.
```

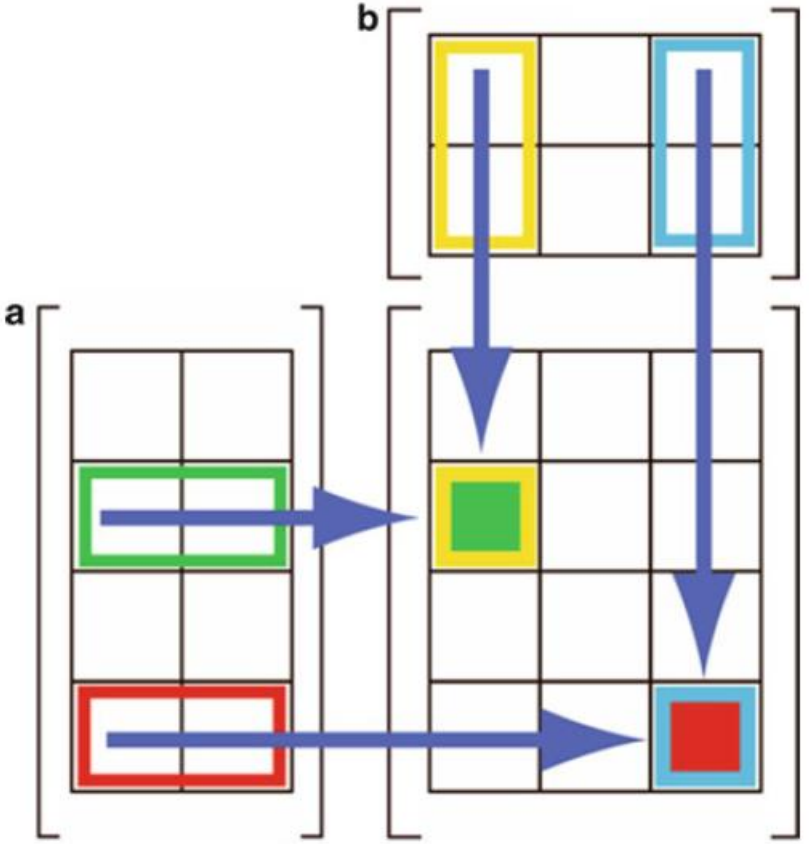
Matrix Operations

- Element-wise operations

Description	MATLAB operator	Example
Element-by-element Multiplication	.*	>> A.*B ans= 2 15 8 3
Element-by-element Right division	./	>> A./B ans = 0.5000 1.6667 0.5000 3.0000
Element-by-element Left division	.\	>> A.\B ans = 2.0000 0.6000 2.0000 0.3333
Element-by-element Exponentiation	.^	>> A.^B ans = 1 125 16 3

Matrix Multiplication

- A is a $m \times n$ matrix
- B is a $n \times p$ matrix
- $(AB)_{i,j} = \sum_{r=1}^{n} A_{i,r} B_{r,j}$



Matrix Multiplication

```
>> D*C
??? Error using ==> mtimes
Inner matrix dimensions must agree.

>> C*D
ans =
    22    54    26
    16    57    18
```

Matrix Multiplication

- **Example 1:** Application in linear algebra

$$\begin{aligned} x + y + 2z &= 9 \\ 2x + 4y - 3z &= 1 \\ 3x + 6y - 5z &= 0 \end{aligned}$$

```
>> A = [1  1  2  9;  2  4  -3  1;  3  6  -5  0]
A =
   1   1   2   9
   2   4  -3   1
   3   6  -5   0
>> rref(A)
ans =
   1   0   0   1
   0   1   0   2
   0   0   1   3
```

Matrix Multiplication

Example 2: Suppose you have five different products in your shop being sold with five different prices, and you sell them in five different quantities. How you can compute your revenue using matrix calculations?

```
>> Prices = [10   20   30   40   50];
>> Sales = [50;   30;   20;   10;   1];
>> Revenue = Prices*Sales
Revenue =
   2150
```

Exercise 1: Solve the Mystery!!

Suppose you are given a matrix and a list of MATLAB operations with the exact number of times they are used. Figure out how this matrix can be constructed with the exact number of performing these operations:

- **Start:step:stop (3)**
- **Element-wise subtraction of two columns(1)**
- **Sine function applied to a column (1)**
- **Assigning one value in a column to zero (1)**
- **Number pi (1)**

```
mysteryMatrix =

    3.0000         0    1.0000    2.0000         0
    6.0000    0.7854    2.0000    4.0000    0.7071
    9.0000    1.5708    3.0000         0    1.0000
   12.0000    2.3562    4.0000    8.0000    0.7071
   15.0000    3.1416    5.0000   10.0000    0.0000
```

Exercise 2: General Linear Model

General Linear Models (GLMs) are widely used to localize brain activity in functional imaging. A standard GLM can be written as:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

Where:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix} \times \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

BOLD Data

Design Matrix

Regression Coefficients

Noise

Exercise 2: General Linear Model

- In this exercise you are asked to reconstruct the brain activity from the parameters of the GLM that has been given to you:

- **Design matrix:** $\boldsymbol{X}$ is a 8×8 matrix.
- **Regression coefficients matrix:** $\boldsymbol{\beta}$ is a 8×1 matrix (column vector).
- **Noise matrix:** $\boldsymbol{\varepsilon}$ is a 8×1 matrix (column vector).

- You can follow the steps to initialize given matrices:
  - $\boldsymbol{\beta}$ is a column vector ranging from 0.1 to 0.8 with step sizes of 0.1.
  - $\boldsymbol{\varepsilon}$ can be created by concatenating this vector [-1 1], 4 times.

Exercise 2: General Linear Model

- You can follow the steps to initialize $\boldsymbol{X}$ matrix:

- 1st column: a column vector of ones with length of 8
- 2nd column: 1.2:0.6:5.4
- 3rd column: addition of first and second columns
- 4th column: second column with its 6th and 8th elements assigned to 0.25
- 5th column: element-wise multiplication of second and fourth column
- 6th column: sine of the values in the 3rd column
- 7th column: cosine of [pi/8:pi/8:pi]
- 8th column: 7th column with elements in the second half of it changed to elements in the first half of 6th column.

- $\boldsymbol{\beta}$ is a column vector ranging from 0.1 to 0.8 with step sizes of 0.1.
- $\boldsymbol{\varepsilon}$ can be created by concatenating this vector [-1 1], 4 times.

References

- **MATLAB for Psychologists (2012),** Borgo, M., Soranzo, A., Grassi, M., Springer-Verlag, 2012, ISBN. 978-1-4614-2196-2.
  - Chapter 1. Basic Operations, pp. 1-23.

- **MATLAB for Neuroscientists, 2nd Ed: An Introduction to Scientific Computing (2014),** *Wallisch, P., Lusignan, M.E., Benayoun, M.D., Baker, T.I., Dickey, A.S. and Hatsopoulos, N.G.,* Academic Press, ISBN. 978-0123838360.
  - Chapter 2. MATLAB Tutorial, pp. 7-56.