

# Matlab Tutorial

Berkan Solmaz

Office: HEC 254 (Computer Vision Lab)

Office Hours: Mon 16:00-17:00

Wed 12:00-13:00

Email: [berkansolmaz@yahoo.com](mailto:berkansolmaz@yahoo.com)

# Introduction to mathematical programming

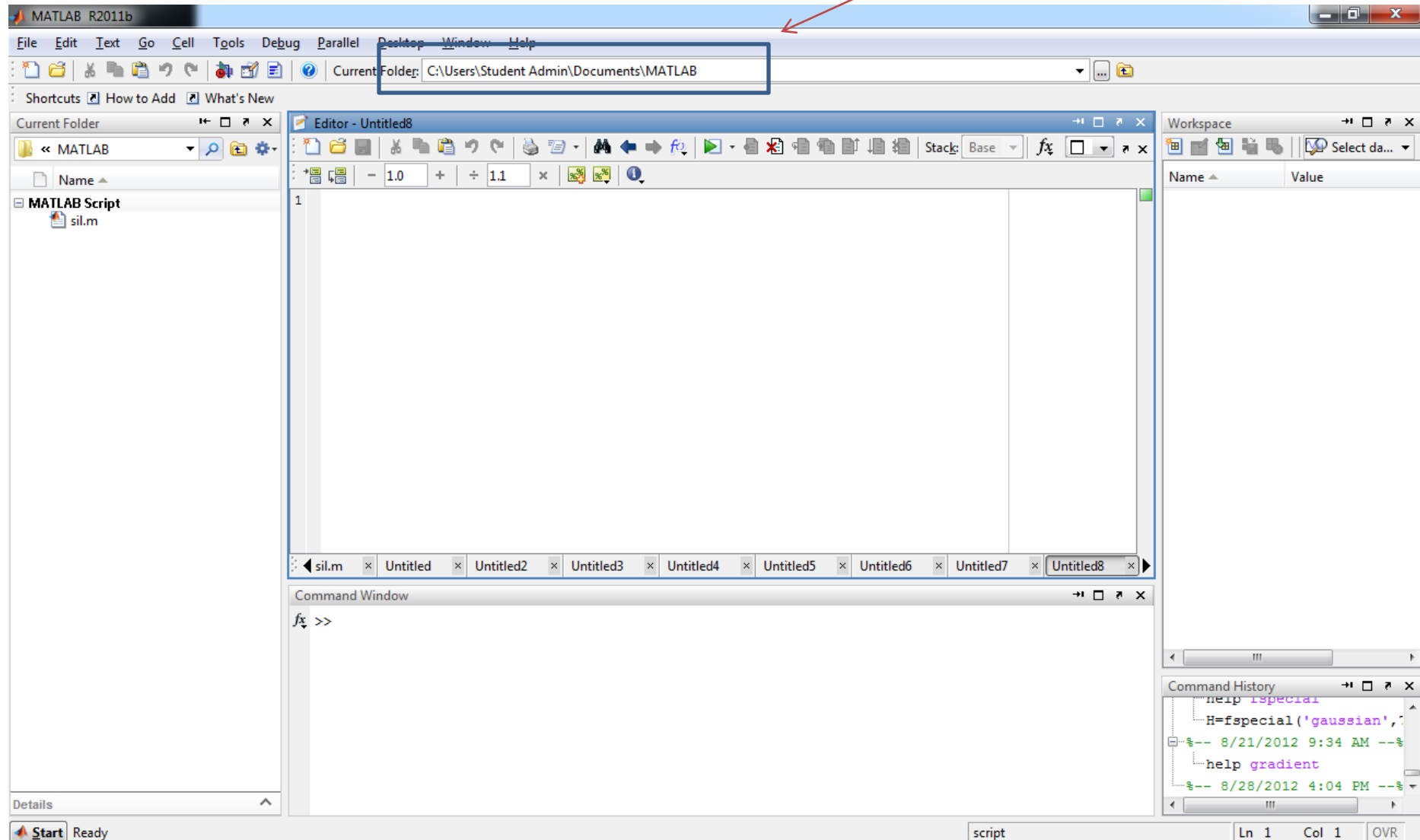
- Commonly used coding/computing environment in research: MATLAB (Matrix laboratory)
  - Ideal for numerical computation
  - Easy to learn
  - No compiling hassles
  - Quick testing of ideas!
  - Helpful tools and tricks for engineers/researchers
  - Execution of tasks on command prompt

# Matlab

- Allows:
  - Matrix manipulations
  - Visualization of data
  - Implementation of algorithms
  - Creation of user interfaces.
- Default Layout:
  - Current Folder
  - Editor
  - Command Window
  - Workspace
  - Command History

# Matlab Layout

Current directory



# Help

- help
- help *command*
- Help on toolbar

# Matlab

## Variables

- dynamically typed
- variable types may be changed

## Basic Operations

- Sum, subtract, multiply, divide, power, sqrt, sin, cos, etc.

## Vector (1-D matrix, array)

- Ordered set of numbers
- Example:  $(x,y,z)$  coordinates of pt in space.

$$x = (x_1, x_2, \dots, x_n)$$

$$\|x\| = \sqrt{\sum_{i=1}^n x_i^2}$$

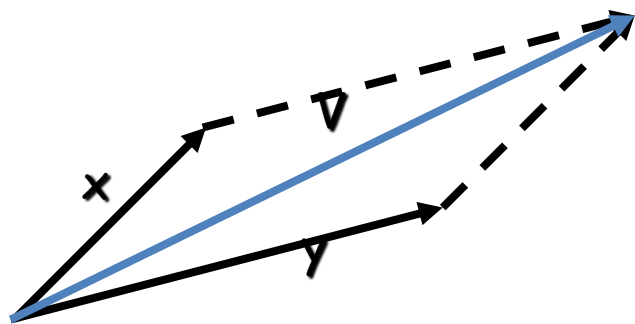
If  $\|x\| = 1$ ,  $x$  is a unit vector

# Indexing into vectors



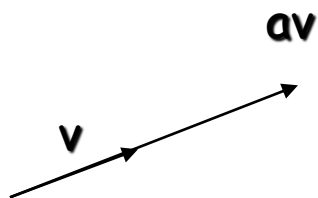
# Vector Addition

$$\begin{aligned}\mathbf{v} &= (x_1, x_2) + (y_1, y_2) \\ &= (x_1 + y_1, x_2 + y_2)\end{aligned}$$



# Scalar Product

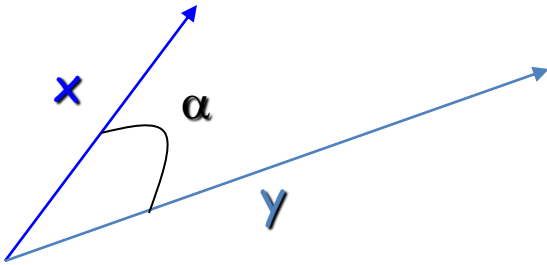
$$a\mathbf{v} = a(x_1, x_2) = (ax_1, ax_2)$$



# Operations on Vectors

- sum
- max, min, mean, ...
- Pointwise:  $\wedge$
- sort
- hist

# Inner (dot) Product



$$\begin{aligned}x \cdot y &= (x_1, x_2) \cdot (y_1, y_2) \\&= x_1 y_1 + x_2 y_2\end{aligned}$$

The inner product is a **SCALAR!**

$$\begin{aligned}x \cdot y &= (x_1, x_2) \cdot (y_1, y_2) \\&= \|x\| \cdot \|y\| \cos \alpha\end{aligned}$$

$$x \cdot y = 0 \Leftrightarrow x \perp y$$

# Matrices 1

$$A_{n \times m} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ a_{31} & a_{32} & \cdots & a_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

**Sum:**

$$C_{n \times m} = A_{n \times m} + B_{n \times m}$$

$$c_{ij} = a_{ij} + b_{ij}$$

**A and B must have the same dimensions**

# Matrices 2

**Product:**

$$C_{n \times p} = A_{n \times m} B_{m \times p}$$

$$c_{ij} = \sum_{k=1}^m a_{ik} b_{kj}$$

$$A_{n \times n} B_{n \times n} \neq B_{n \times n} A_{n \times n}$$

**A and B must have compatible dimensions**

**Identity Matrix:**

$$I = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \ddots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} \quad IA = AI = A$$

# Matrices 3

**Transpose:**

$$C_{m \times n} = A^T_{n \times m}$$

$$c_{ij} = a_{ji}$$

If  $A^T = A$ , **A is symmetric**

$$(A + B)^T = A^T + B^T$$

$$(AB)^T = B^T A^T$$

# Matrices 4

**Determinant:** *A must be square*

$$\det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12}$$

$$\det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \\ = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$



# Matrices 5

**Inverse:**

$$A_{n \times n} A^{-1}_{n \times n} = A^{-1}_{n \times n} A_{n \times n} = I$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \frac{1}{a_{11}a_{22} - a_{21}a_{12}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

# Images

- 2-D array of numbers (intensity values, gray levels)
- Gray levels 0 (black) to 255 (white)
- Color image is 3 2-D arrays of numbers
  - Red
  - Green
  - Blue

- Images
  - Indexing 2-D Matrices

- Images
  - Indexing 2-D Matrices
  - Reading/Showing Images

- Images
  - Indexing 2-D Matrices
  - Reading/Showing Images
  - RGB to Grayscale
  - Finding size of an image

- Images

- Indexing 2-D Matrices
- Reading/Showing Images
- RGB to Grayscale
- Finding size of an image
- Thresholding
- Selecting indices
- Initializing an image

- Images

- Indexing 2-D Matrices
- Reading/Showing Images
- RGB to Grayscale
- Finding size of an image
- Thresholding
- Selecting indices
- Initializing an image
- Histogram of image

- Images

- Indexing 2-D Matrices
- Reading/Showing Images
- RGB to Grayscale
- Finding size of an image
- Thresholding
- Selecting indices
- Initializing an image
- Histogram of image
- Binary thresholding image



- Images

- Indexing 2-D Matrices
- Reading/Showing Images
- RGB to Grayscale
- Finding size of an image
- Thresholding
- Selecting indices
- Initializing an image
- Histogram of image
- Binary thresholding image
- Averaging with box kernel

- Images

- Indexing 2-D Matrices
- Reading/Showing Images
- RGB to Grayscale
- Finding size of an image
- Thresholding
- Selecting indices
- Initializing an image
- Histogram of image
- Binary thresholding image
- Averaging with box kernel
- Median filtering

- Images

- Indexing 2-D Matrices
- Reading/Showing Images
- RGB to Grayscale
- Finding size of an image
- Thresholding
- Selecting indices
- Initializing an image
- Histogram of image
- Binary thresholding image
- Averaging with box kernel
- Median filtering
- Derivatives/Convolution
- Displaying images

- Other
  - Gaussian kernel

- Other

- Gaussian kernel
- Loops/Initializing variables

- Other

- Gaussian kernel
- Loops/Initializing variables
- Repeating matrices

- Other

- Gaussian kernel
- Loops/Initializing variables
- Repeating matrices
- Generating random matrices

- Other

- Gaussian kernel
- Loops/Initializing variables
- Repeating matrices
- Generating random matrices
- Reading video frames



- Other

- Gaussian kernel
- Loops/Initializing variables
- Repeating matrices
- Generating random matrices
- Reading video frames
- Reading video file

- Other

- Gaussian kernel
- Loops/Initializing variables
- Repeating matrices
- Generating random matrices
- Reading video frames
- Reading video file
- Plotting 1D and 2D

- Other

- Gaussian kernel
- Loops/Initializing variables
- Repeating matrices
- Generating random matrices
- Reading video frames
- Reading video file
- Plotting 1D and 2D
- Saving workspace

- Other

- Gaussian kernel
- Loops/Initializing variables
- Repeating matrices
- Generating random matrices
- Reading video frames
- Reading video file
- Plotting 1D and 2D
- Saving workspace
- Defining functions

Format: `function o = myTestFunc(x,y)`

Name function and file the same.

Only first function in file is visible outside the file.

- Other

- Gaussian kernel
- Loops/Initializing variables
- Repeating matrices
- Generating random matrices
- Reading video frames
- Reading video file
- Plotting 1D and 2D
- Saving workspace
- Defining functions

Format: `function o = myTestFunc(x,y)`

Name function and file the same.

Only first function in file is visible outside the file.

- Debugging

- Other

- Gaussian kernel
- Loops/Initializing variables
- Repeating matrices
- Generating random matrices
- Reading video frames
- Reading video file
- Plotting 1D and 2D
- Saving workspace
- Defining functions

Format: `function o = myTestFunc(x,y)`

Name function and file the same.

Only first function in file is visible outside the file.

- Debugging
- String operations

Good Luck