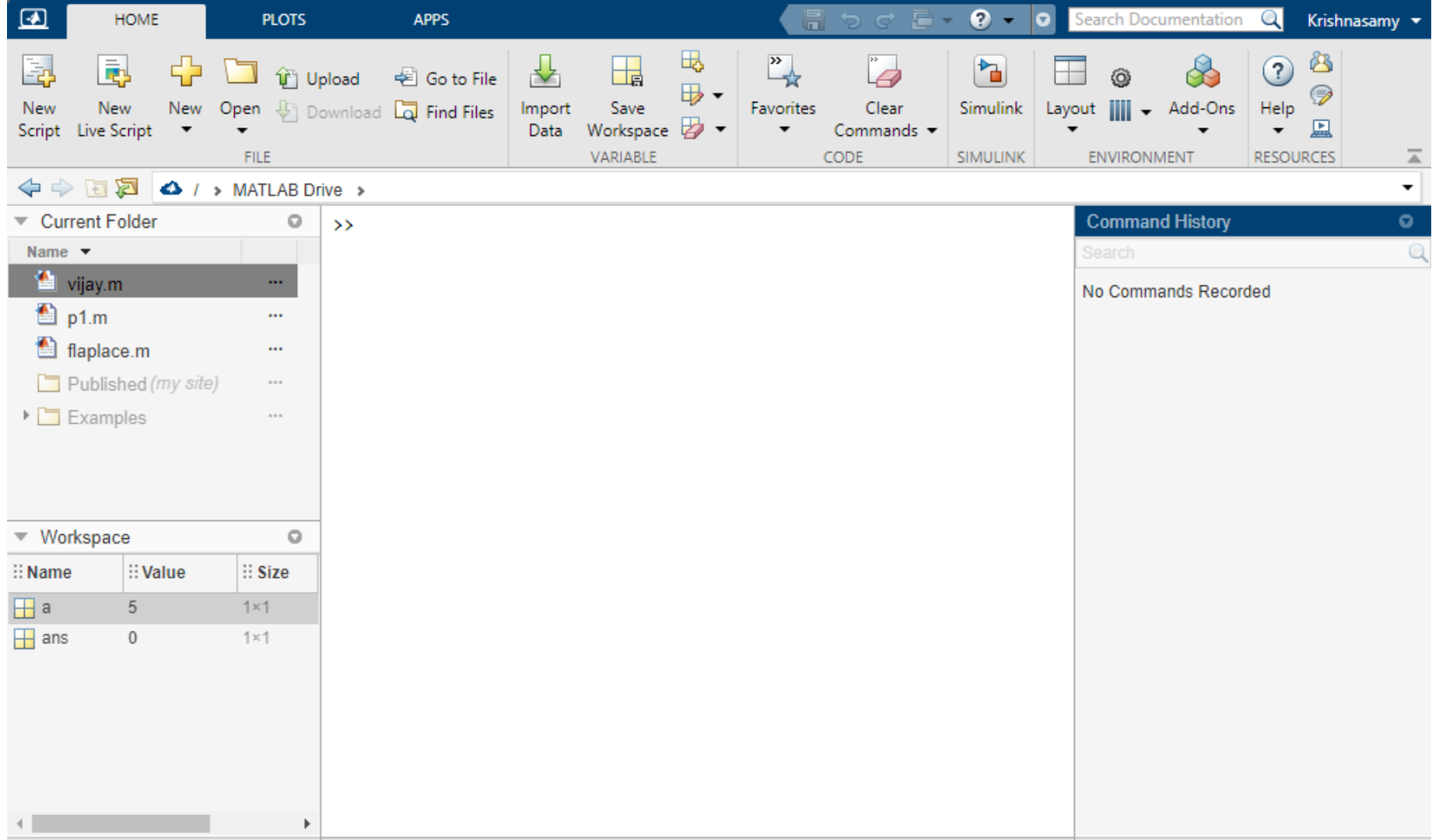


MATHEMATICAL FOUNDATIONS LAB

INTRODUCTION TO MATLAB

STARTING MATLAB, MATLAB WINDOWS

- Once the program starts, the MATLAB desktop window opens with the default layout
- The layout has a Toolstrip at the top, the Current Folder Toolbar below it, and four windows underneath
- At the top of the Toolstrip there are three tabs: HOME, PLOTS, and APPS
- The default layout consists of the following four windows
 - Command Window
 - Current Folder Window
 - Workspace window
 - Command History window



Default view of MATLAB desktop

Table 1-1: MATLAB windows

Window	Purpose
Command Window	Main window, enters variables, runs programs.
Figure Window	Contains output from graphic commands.
Editor Window	Creates and debugs script and function files.
Help Window	Provides help information.
Command History Window	Logs commands entered in the Command Window.

Working in the command window

- To type a command, the cursor must be placed next to the command prompt (>>).
- Once a command is typed and the Enter key is pressed, the command is executed.
- However, only the last command is executed. Everything executed previously is unchanged.
- Several commands can be typed in the same line. This is done by typing a comma between the commands. When the Enter key is pressed, the commands are executed in order from left to right.
- It is not possible to go back to a previous line that is displayed in the Command Window, make a correction, and then re-execute the command.

... continued

Working in the command window

- A previously typed command can be recalled to the command prompt with the uparrow key (↑). When the command is displayed at the command prompt, it can be modified if needed and then executed. The down-arrow key (↓) can be used to move down the list of previously typed commands.
- If a command is too long to fit in one line, it can be continued to the next line by typing three periods ... (called an ellipsis) and pressing the Enter key. The continuation of the command is then typed in the new line.
- If a semicolon (;) is typed at the end of a command, the output of the command is not displayed
- When the symbol % (percent) is typed at the beginning of a line, the line is designated as a comment.
- The clc command (type clc and press Enter) clears the Command Window

Arithmetic operations with scalars

<u>Operation</u>	<u>Symbol</u>	<u>Example</u>
Addition	+	$5 + 3$
Subtraction	—	$5 - 3$
Multiplication	*	$5 * 3$
Right division	/	$5 / 3$
Left division	\	$5 \setminus 3 = 3 / 5$
Exponentiation	^	$5 \wedge 3$ (means $5^3 = 125$)

Order of Preference

MATLAB executes the calculations according to the following order of precedence

Precedence	Mathematical Operation
First	Parentheses. For nested parentheses, the innermost are executed first.
Second	Exponentiation.
Third	Multiplication, division (equal precedence).
Fourth	Addition and subtraction.

- In an expression that has several operations, higher-precedence operations are executed before lower-precedence operations.
- If two or more operations have the same precedence, the expression is executed from left to right.

Table 1-3: Elementary math functions

Function	Description	Example
<code>sqrt (x)</code>	Square root.	<pre>>> sqrt(81) ans = 9</pre>
<code>nthroot (x,n)</code>	Real n th root of a real number x . (If x is negative n must be an odd integer.)	<pre>>> nthroot(80,5) ans = 2.4022</pre>
<code>exp (x)</code>	Exponential (e^x).	<pre>>> exp(5) ans = 148.4132</pre>

Table 1-3: Elementary math functions (Continued)

Function	Description	Example
<code>abs(x)</code>	Absolute value.	<pre>>> abs(-24) ans = 24</pre>
<code>log(x)</code>	Natural logarithm. Base e logarithm (ln).	<pre>>> log(1000) ans = 6.9078</pre>
<code>log10(x)</code>	Base 10 logarithm.	<pre>>> log10(1000) ans = 3.0000</pre>
<code>factorial(x)</code>	The factorial function $x!$ (x must be a positive integer.)	<pre>>> factorial(5) ans = 120</pre>

Table 1-4: Trigonometric math functions

Function	Description	Example
<code>sin(x)</code> <code>sind(x)</code>	Sine of angle x (x in radians). Sine of angle x (x in degrees).	<pre>>> sin(pi/6) ans = 0.5000</pre>
<code>cos(x)</code> <code>cosd(x)</code>	Cosine of angle x (x in radians). Cosine of angle x (x in degrees).	<pre>>> cosd(30) ans = 0.8660</pre>
<code>tan(x)</code> <code>tand(x)</code>	Tangent of angle x (x in radians). Tangent of angle x (x in degrees).	<pre>>> tan(pi/6) ans = 0.5774</pre>
<code>cot(x)</code> <code>cotd(x)</code>	Cotangent of angle x (x in radians). Cotangent of angle x (x in degrees).	<pre>>> cotd(30) ans = 1.7321</pre>

Table 1-5: Rounding functions

Function	Description	Example
<code>round(x)</code>	Round to the nearest integer.	<pre>>> round(17/5) ans = 3</pre>
<code>fix(x)</code>	Round toward zero.	<pre>>> fix(13/5) ans = 2</pre>
<code>ceil(x)</code>	Round toward infinity.	<pre>>> ceil(11/5) ans = 3</pre>
<code>floor(x)</code>	Round toward minus infinity.	<pre>>> floor(-9/4) ans = -3</pre>
<code>rem(x,y)</code>	Returns the remainder after x is divided by y .	<pre>>> rem(13,5) ans = 3</pre>
<code>sign(x)</code>	Signum function. Returns 1 if $x > 0$, -1 if $x < 0$, and 0 if $x = 0$.	<pre>>> sign(5) ans = 1</pre>

Rules About Variable Names

A variable can be named according to the following rules:

- Must begin with a letter.
- Can be up to 63 characters long.
- Can contain letters, digits, and the underscore character.
- Cannot contain punctuation characters (e.g., period, comma, semicolon).
- MATLAB is case-sensitive: it distinguishes between uppercase and lowercase letters. For example, AA, Aa, aA, and aa are the names of four different variables.
- No spaces are allowed between characters (use the underscore where a space is desired).
- Avoid using the name of a built-in function for a variable (i.e., avoid using cos, sin, exp, sqrt, etc.).
- Once a function name is used to for a variable name, the function cannot be used.

Predefined Variables and Keywords

There are 20 words, called keywords, that are reserved by MATLAB for various purposes and cannot be used as variable names. These words are:

break	case
catch	classdef
continue	else
elseif	end
for	function
global	if
otherwise	parfor
persistent	return
spmd	switch
try	while

Predefined Variables

A number of frequently used variables are already defined when MATLAB is started. Some of the predefined variables are:

Keyword	Meaning
ans	A variable that has the value of the last expression that was not assigned to a specific variable. If the user does not assign the value of an expression to a variable, MATLAB automatically stores the result in ans.
pi	The number π
eps	The smallest difference between two numbers. Equal to $2A(-52)$, which is $2.2204e-016$
inf	Used for infinity
i	Defined as $\sqrt{-1}$, which is: $0 + 1.0000i$
j	Same as I
NaN	Stands for Not-a-Number. Used when MATLAB cannot determine a valid numeric value. Example: $0/0$.

Commands For Managing Variables

Command	Outcome
clear	Removes all variables from the memory
clear x y z	Removes only variables x, y and z from the memory
who	Displays a list of the variables currently in the memory
whos	Displays a list of the variables currently in the memory and their sizes together with information about their bytes and class

Creating a vector from a known list of numbers:

`variable_name = [type vector elements]`

Row vector:

To create a row vector type the elements with a space or a comma between the elements inside the square brackets.

Column vector:

To create a column vector type the left square bracket [and then enter the elements with a semicolon between them, or press the Enter key after each element. Type the right square bracket] after the last element.

Creating a vector with constant spacing by specifying the first term, the spacing, and the last term:

`variable_name = [m:q:n]` or `variable_name = m:q:n`

Creating a vector with linear (equal) spacing by specifying the first and last terms, and the number of terms:

`variable_name = linspace(xi,xf,n)`

- xi – first element
- xf – final element
- n – number of elements

CREATING A TWO-DIMENSIONAL ARRAY (MATRIX)

variable_name= [1st row elements; 2nd row elements;
3rd row elements; . . . ; last row elements]

The zeros, ones and, eye Commands

- The zeros (m, n), ones (m, n), and eye (n) commands can be used to create matrices that have elements with special values.
- The zeros (m, n) and the ones (m, n) commands create a matrix with m rows and n columns in which all elements are the numbers 0 and 1, respectively.
- The eye (n) command creates a square matrix with n rows and n columns in which the diagonal elements are equal to 1 and the rest of the elements are 0.

THE TRANSPOSE OPERATOR

The transpose operator is applied by typing a single quote ' following the variable to be transposed

ARRAYADDRESSING

Elements in an array (either vector or matrix) can be addressed individually or in subgroups.

Vector

The address of an element in a vector is its position in the row (or column). For a vector named ve , $ve(k)$ refers to the element in position k .

Matrix

The address of an element in a matrix is its position, defined by the row number and the column number where it is located.

USING A COLON : IN ADDRESSING ARRAYS

For a vector:

- $va(:)$ Refers to all the elements of the vector va (either a row or a column vector).
- $va(m:n)$ Refers to elements m through n of the vector va .

For a matrix:

- $A(:,n)$ - Refers to the elements in all the rows of column n of the matrix A .
- $A(n,:)$ - Refers to the elements in all the columns of row n of the matrix A .
- $A(:,m:n)$ - Refers to the elements in all the rows between columns m and n of the matrix A .
- $A(m:n,:)$ - Refers to the elements in all the columns between rows m and n of the matrix A .
- $A(m:n,p:q)$ - Refers to the elements in rows m through n and columns p through q of the matrix A .