# **INTRODUCTION TO NUMERICAL ANALYSIS**

Cho, Hyoung Kyu

Department of Nuclear Engineering

Seoul National University





# **0. MATLAB USAGE**





#### MATLAB

- MATrix LABoratory
- Mathematical computations, modeling and simulations, data analysis and processing, visualization and graphics, and algorithm development
- Standard MATLAB
- Toolboxes
  - Signal processing, symbolic calculations, control system, image processing, etc.

#### MATLAB download

My SNU 서울대학교 포털 마이스누		QUICK MENU	<b>합검색</b> 원롤릭검색 마이스크랩	세계를 선도하는 치능의가ς 지시공동차1		
학사/행정 웹메일 eTL	전지			✔ 전체메뉴보기	QUICK MENU	
학사/행정       웹메일       eTL         관양합니다       조형구님       메일작성         페일수신함 0       메일작성       최종 로그인: 20150816 11:19         교수       •       •         나의 관련사이트       •       •         나의 관련사이트       •       •         한 0       상담 0       수강지도 0         행정 결제       0       공담 1       담당 1         기급       기만       0       대기       0       공지 0         2015학년도 2학기 국가장학금 …       2015,08,25 ~ 2015,09,04       •       •       •	전자 Confi [Obit Invali 교수는 [소식 Come Do yx 2015/ 2015/ 2015/ 2015/ 2015/ 2015/	도서검색 SMS SNU Talk 주간식단표 주차/교통 친절/불친절신고 IT서비스 건의 서울대병원 예약 보건진료소 예약 SW 다운로드 학칙 및 규정 포털이용안내	공광         1ces       2015.08.15         2       2015.08.15         2       2015.08.15         2       2015.08.14         2015.08.14       2015.08.14         2015.08.13       3         대보기       15.08.14         2015.08.13       6         적입력       209평가         학성지도       1         도서관       기초교육원	오지시형       단대공지         연건캠퍼스 전산당 중단       [스트레스클리닉] 9월         [스트레스클리닉] 9월       몇 ፲         무레문예관(67동) 냉년       [지서비스센터 설문조사         [알림] 부속치과의원 후       [양감]         행사 / 이벤트       한구교         한 / 이벤트       한국교         한 / 이벤트       한국교         한 / 이벤트       한국교         한 / 이벤트       한국교         · · · · · · · · · · · · · · · · · · ·	QUICK MENU 도서검색 SMS SNU Talk 주간식단표 주차/교통 천절/발천절신고 IT서비스 건의 서울대병원 예약 보건진료소 예약 SW 다운로드 학칙 및 규정 포함이용안내 고함이용안내 가 문격혁신과 문역혁신과 문역혁신과 문역혁신과 문역혁신과 문역혁신과 문역혁신과 문역혁신과 문역혁신과 문역혁신과 문역혁신과 문역혁신과 문역혁신과 문역혁신과 문역혁신과 문역혁신과 문역혁신과 문역혁신과 문역 관심 전 2015	

#### MATLAB interface



#### Command window

- Commands are typed next to the prompt (>>) and are executed when the *Enter* key is pressed.
- Output generated by the command is displayed in the Command Window, unless a semicolon (;) is typed at the end.
- clc command: clear Command Window
- Up-arrow and down-arrow keys ( $\uparrow$  and  $\downarrow$ ): recall commands typed before

Operation	Symbol	Example	Operation	Symbol	Example	>> 7 + 8/2 ans =
Addition	+	5 + 3	Right division	1	5/3	11
Subtraction	-	5-3	Left division	١	$5 \setminus 3 = 3 / 5$	>> (7+8)/2 + 27^(1/3
Multiplication	*	5 * 3	Exponentiation	^	$5 \wedge 3 \text{ (means } 5^3 = 125)$	ans = 10.5000

명령 창	$\odot$
>> 1+1	
ans =	
2	
≫ 1+1; <b>f</b> x =>	

#### Command window

Numerical values can be assigned to variables.



- Elementary math built-in functions
  - Ex) sqrt(x)

>>

an

>>

an

sqrt(64)	>> sqrt(54 + 9*sqrt(100)) Argument includes a function.
s = 8	ans = 12
sqrt(50 + 14*3)	>> (15 + 600/4)/sqrt(121) Function is included in an expression.
s =	ans =
9.5917	15

### **\*** Built-in elementary math functions

Command	Description	Example			
sqrt(x)	Square root.	>> sqrt(81) ans = 9			
exp(x)	Exponential $(e^x)$ .	>> exp(5) ans = 148.4132			
abs(x)	Absolute value.	>> abs(-24) ans = 24			
log(x)	Natural logarithm. Base e logarithm (ln).	>> log(1000) ans = 6.9078			
log10(x)	Base 10 logarithm.	>> log10(1000) ans = 3.0000			
sin(x)	Sine of angle $x$ ( $x$ in radians).	>> sin(pi/6) >> sind(30) ans = ans =			
sind(x)	Sine of angle $x$ ( $x$ in degrees).	0.5000 0.5000			

### Built-in elementary math functions

Command	Description	Example						
The other trigonometric functions are written in the same way. The inverse trigonometric functions are written by adding the letter "a" in front, for example, $asin(x)$ .								
round(x)	Round to the nearest integer.	>> round(17/5) ans = 3						
fix(x)	Round toward zero.	>> fix(9/4) >> fix(-9/4) ans = ans = 2 -2						
ceil(x)	Round up toward infinity.	>> ceil(11/5) ans = 3						
floor(x)	Round down toward minus infinity.	>> floor(-9/4) ans = -3						

### Display format

Command	Description	Example	
format short	Fixed point with four decimal digits for: 0.001 ≤ number ≤ 1000 Otherwise display format short e.	>> 290/7 ans = 41.4286	명령 창
format long	Fixed point with 14 decimal digits for: 0.001 ≤ number ≤ 100 Otherwise display format long e.	>> 290/7 ans = 41.42857142857143	ans =
format short e	Scientific notation with four decimal digits.	>> 290/7 ans = 4.1429e+001	41.4286
format long e	Scientific notation with 15 decimal dig- its.	>> 290/7 ans = 4.142857142857143e+001	>> format long >> 290/7
format short g	Best of 5-digit fixed or floating point.	>> 290/7 ans = 41.429	ans = 41.428571428571431
format long g	Best of 15-digit fixed or floating point.	>> 290/7 ans = 41.4285714285714	$f_{x} \gg$
format bank	Two decimal digits.	>> 290/7 ans = 41.43	

3. Arrays

#### Arrays 명령 창 1D array: vector a = 2D array: matrix 2 3 Each number in a vector or a matrix: element >> a=[1;2;3] a = Creating a vector in MATLAB 2 Row vector variable name = [number number ... number] З $fx \rightarrow$ variable name=[number; number; number] Column vector ";" is placed! variable name = m:q:n Row vector with constant spacing 명령 창 $\odot$ >> a=1:2:10 m : first element a = q : spacing 1 З 5 7 9 n : last element fx >>111

### Arrays

• Creating a vector in MATLAB

"linspace" command

variable\_name = linspace(xi,xf,n)

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

명	령 창					$\odot$
	>>	a=linspac	e(1,10,5)			
	a =	:				
		1.0000	3.2500	5,5000	7.7500	10.0000
~						
Jx.	>>					

### Arrays

Creating a matrix in MATLAB

variable\_name = [1st row elements ; 2nd row elements ; ....; last row elements]

- Variables and mathematical expressions can be used as elements.
- Array addressing

*ve* = 35 46 78 23 5 14 81 3 55

ve(4) = 23, ve(7) = 81, and ve(1) = 35

$$ma = \begin{bmatrix} 3 & 11 & 6 & 5 \\ 4 & 7 & 10 & 2 \\ 13 & 9 & 0 & 8 \end{bmatrix} \qquad ma(1,1) = 3, \text{ and } ma(2,3) = 10$$

명	령 창 (	•
	>> a= [5 35 43; 4 76 81; 21 32 40]	
	a =	
	5 35 43	
	4 76 81	
	21 32 40	
fx,	»>	
명령	영창 (	Ð
	>> cd = 6; e = 3; h = 4; >> Mat= [e, cd*h, cos(pi/3); h^2, sqrt(h*h/cd), 14 ]	
	Mat =	
	3.0000 24.0000 0.5000	
	16.0000 1.6330 14.0000	
fx,	»»	

**3.** Arrays

#### Arrays

#### Array addressing

>>  $VCT = [35 \ 46 \ 78 \ 23 \ 5$ 14 81 3 55] Define a vector. VCT =35 46 78 23 5 14 81 3 55 Assign new values to the fourth and sixth elements. >> VCT(4) = -2; VCT(6) = 273VCT = 5 273 35 46 78 -2 81 3 55 Use vector elements in a >> VCT(5)^VCT(8) + sqrt(VCT(7)) mathematical expression. ans = 134 >> MAT = [3 11 6 5; 4 7 10 2; 13 9 0 8] Define a matrix. MAT =3 11 6 5 2 10 7 4 8 13 9 0 >> MAT(3,1) = 20Assign a new value to the (3,1) element. MAT =3 11 6 5 7 10 2 4 20 9 0 8 >> MAT(2,4) - MAT(1,2)Use matrix elements in a mathematical expression. ans = -9

#### Arrays

- Array addressing
  - Using a colon ": " in addressing arrays

```
Define a vector.
>> v = [4 \ 15 \ 8 \ 12 \ 34 \ 2 \ 50 \ 23 \ 11]
v =
        15
                     12
                            34
                                         50
   4
               8
                                    2
                                                23
                                                      11
                               Vector u is created from the elements 3 through 7 of vector v.
>> u = v(3:7)
u =
        12
              34
                      2
                            50
   8
>>A= [1 3 5 7 9 11; 2 4 6 8 10 12; 3 6 9 12 15 18; 4 8 12 16 20
24; 5 10 15 20 25 30]
                                                                    Define a matrix.
\mathbf{A} =
           3
                        7
                               9
                                    11
    1
                  5
    2
                 6
                       8
                             10
                                    12
           4
                 9
    3
          6
                      12
                             15
                                    18
    4
                             20
          8
                12
                      16
                                    24
                15
    5
         10
                      20
                             25
                                    30
                                      Vector C is created from the second row of matrix A.
C = A(2, :)
C =
                    8 10 12
    2
            6
         4
                                           Matrix F is created from the elements in rows 1
>> F = A(1:3,2:4)
                                           through 3 and columns 2 through 4 of matrix A.
```

### Arrays

#### • Built-in functions for handling arrays

Command	Description	Example						
length(A)	Returns the number of elements in vector A.	>> A = [5 9 2 4]; >> length(A) ans = 4						
size(A)	Returns a row vector [m, n], where m and n are the size $m \times n$ of the array A. (m is number of rows. n is number of columns.)	<pre>&gt;&gt;A = [6 1 4 0 12; 5 19 6 8 2] A =     6 1 4 0 12     5 19 6 8 2 &gt;&gt; size(A) ans =     2 5</pre>						
zeros(m,n)	Creates a matrix with $m$ rows and $n$ columns, in which all the elements are the number 0.	>> zr = zeros(3,4) zr = 0 0 0 0 0 0 0 0 0 0 0 0						
ones(m,n)	Creates a matrix with <i>m</i> rows and <i>n</i> columns, in which all the elements are the number 1.	>> ne = ones (4,3) ne = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1						
eye(n)	Creates a square matrix with $n$ rows and $n$ columns in which the diagonal elements are equal to 1 (identity matrix).	<pre>&gt;&gt; idn = eye (3) idn =     1     0     0     0     1     0     0     0     1</pre>						



### 3. Arrays

#### Strings

- A string is an array of characters.
- It is created by typing the characters within single quotes.
- Strings can include letters, digits, other symbols, and spaces.
- Concatenate strings horizontally
  - strcat : string concatenate!
  - How can create 'Nuclear Engineering' using strcat command?
  - **32**?





#### Addition and subtraction of arrays

>> VA = [8 5 4]; VB = [10 2 7];Define two vectors VA and VB. Define a vector VC that is equal to VA + VB. >> VC = VA + VB VC =18 7 11 Define two matrices A and B. >> A = [5 -3 8; 9 2 10], B = [10 7 4; -11 15 1] $\mathbf{A} =$ 5 -3 8 9 2 10  $\mathbf{B} =$ 10 7 4 -11 15 1 Define a matrix C that is equal to A + B. >> C = A + BC =15 4 12 -2 17 11 Subtract 8 from the matrix C. >> C - 8 8 is subtracted from each element of C. ans =7 -4 4 3 -10 9

### Multiplications of arrays

>> A =	[2 -	-1; 8	3 3;	67],	B = [4	9 1	-3;	-5	2	4	Define two matrices A and B.
<b>A</b> =											
2	-1										
8	3										
6	7										
B =											
4	9	1	-3								
-5	2	4	6								
>> C =	A*B										Multiply A*B.
C =											C is a $(3 \times 4)$ matrix.
13	16	-2	-12								
17	78	20	-6								
-11	68	34	24								

Array division ("/")  $[a][x] = [b] \implies x = a b [x][a] = [b] \implies x = b/a$ 

Element-by-element operations

Symbol	Description	Symbol	Description
*	Multiplication	./	Right division
.^	Exponentiation	٨.	Left Division

>>A=[2	6 3; 5 8 4]	Define a $(2 \times 3)$ matrix
A =		
2 6	5 3	
5 8	3 4	
>> B = [1	4 10; 3 2 7]	Define a $(2 \times 3)$ matrix
в =		
1 4	1 10	
3 2	2 7	
>>A.*B		Element-by-element multiplication of arrays A and
ans =		
2 24	30	
15 14	20	

#### Element-by-element operations

>> C = A . / B	Element-by-element division of array A by array B.
C =	
2.0000 1.5000 0.3000	
1.6667 4.0000 0.5714	
>> B .^ 3	Element-by-element exponentiation of array B.
ans =	
1 64 1000	
27 8 343	

- Usefulness of element-by-element operation
  - For example,

>> z = [1:2:15]	Define a vector z with eight elements.
z =	
1 3 5 7 9 11 13 15	
>> $y = (z.^3 + 5z)./(4z.^2 - 10)$	Vector z is used in element-by-element calculation of the elements of vector y.
y= -1.0000 1.6154 1.6667 2.0323 2.4	650 2.9241 3.3964 3.8764

### **\*** Built-in functions for handling arrays

Command	Description	Example
mean(A)	If A is a vector, the function returns the mean value of the elements of the vector.	>>A=[5 9 2 4]; >>mean(A) ans = 5
sum(A)	If A is a vector, the function returns the sum of the elements of the vector.	>>A=[5 9 2 4]; >>sum(A) ans= 20
sort(A)	If A is a vector, the function arranges the elements of the vector in ascending order.	>>A=[5 9 2 4]; >>sort(A) ans= 2 4 5 9
det(A)	The function returns the determinant of a square matrix A.	<pre>&gt;&gt;A = [2 4; 3 5]; &gt;&gt; det(A) ans =     -2</pre>

#### Script file

- Program: a file that contains a sequence of MATLAB commands
- Extension: ".m"  $\Rightarrow$  M-file

	MATLAB R2014b	and the second			×
	옴 플롯 앱	편집기 퍼블리시	보기 🛃 🔒 🔓 🖻	🖻 🔁 🍞 도움말 검색	₽ ≖
	사         ····································	☐ 파일 찾기 월 비교 데이터 작업 공간 가져오기 저장	값 새 변수 2 코드 분석 한 변수 열기 ♥ 값 실행 시간 값 작업 공간 지우기 ♥ 값 명령 지유 변수 코드	부측정 Simulink 환경 리소스 각기 ▼ 라이브러리 ▼ ▼	
	< 🖚 🖬 🕅 🚺 🕨 D: 🕨	LECTURE ▶ 수치해석 ▶ 2015년2혁	각기 🕨 Lecture00 🕨 practice		
	현재 폴더	· · · · · · · · · · · · · · · · · · ·		⑦ × 작업 공간	$\odot$
	세부정보				
📣 MATLAB R2014b					
홈 플롯	앱 편집기	기 퍼블리시	보기 🛛 🛃 🔓 🎪 📠	🔓 🗇 🔄 🔁 🕐 도움말 (	검색 🔎 🗖
새로 만들기 열기	[ ] 다 파일 찾기 저장 [ ] 비교 ↓ ★ ] 인쇄 ↓	☆ ☆ 삽입 ☆ 이동 ▼ 주석 ○ 찾기 ▼ 들여쓰기		실행 및 및 진행 실	한 행 시간 측정
L	바일	14	편집 중단점	일행	

### Script file

- Input to a script file
  - Define the variable and assign it a value in the script file
    - File must be edited and the assignment of the variable changed.
  - Define the variable and assign it a value in the Command Window

1	편집기 - D:\LECTURE\수치해석\2015년2학기\Lecture00\practice\input (	
	input_script.m 🗶 🕂	
1 -	a=1;	
2 -	b=2;	
3 -	a+b	
4		
_		
1	년집기 - D:\LECTURE\수치해석\2015년2학기\Lecture00\practice\input ④	) ×
	년집기 - D:\LECTURE\수치해석\2015년2학기\Lecture00\practice\input ⓒ input_script.m 🗶 🕂	×
<b>2</b>	면접기 - D:\LECTURE\수치해석\2015년2학기\Lecture00\practice\input ⓒ input_script.m × + a=2;	) ×
<b>2</b> –	편집기 - D:\LECTURE\수치해석\2015년2학기\Lecture00\practice\input ⓒ input_script.m × + a=2; b=2;	) ×
<b>1</b> – 2 – 3 –	년집기 - D:\LECTURE\수치해석\2015년2학기\Lecture00\practice\input ④ input_script.m ※ + a=2; b=2; a+b	
2 – 3 – 4	편집기 - D:\LECTURE\수치해석\2015년2학기\Lecture00\practice\input ④ input_script.m × + a=2; b=2; a+b	

![](_page_23_Picture_7.jpeg)

#### Script file

- Input to a script file
  - Define the variable in the script file but assign a specific value in the Command Window when the script file is executed.
  - "input" command

![](_page_24_Picture_5.jpeg)

### **5. Script Files**

#### Script file \*

- Output from a script file
  - "disp" command
  - "fprintf" command
    - To Command Window
    - To a file

2	편집기 - D:\LECTURE\수치해석\2015년2학기\Lecture00\pra ③	<
	input_script.m 🗶 🕂	
1 -	- a=2;	
2 -	- b=2;	
3 -	- disp(a+b);	
4 -	<pre>- fprintf('write the a+b value: %d\n,a+b);</pre>	
5		
명	령 창 (	9
명	령 창 ( >> input_script	3
8	령 창 >> input_script 4	
8	령 창 >> input_script 4	
B	령 창 >> input_script 4 write the a+b value: 4	
ख fx	령 창 >> input_script 4 write the a+b value: 4 >>	•

현재 폴더 이름 ▲ 1.txt Input_script.asv input_script.m	
Z 편집기 - D:₩LECTURE₩수치해석₩2015년2학기₩Lecture00₩pra	<b>▼</b> ×
input_script.m × +	
1 – a=2;	
2 – b=2;	
3 - disp(a+b);	
<pre>4 - fprintf('write the a+b value: <u>%d\n',a+b</u>);</pre>	
5 - fileID=fopen('1.txt','w'); New line	
6 - fprintf(fileID,'write the a+b value: %d∰n'.a+b → \#r\#n	);
명령 창 New line of text	file 🕤
>> input_script 4	
write the a+b value: 4 $f_x \gg  $	

2

#### Plots in MATLAB

- Standard plots with linear axes and logarithmic axes
- Bars and stairs plots
- "plot" command
  - With 'line specifiers'

![](_page_26_Figure_7.jpeg)

Line Color	Specifier	Line Style	Specifier	L	ine Color	Spe	cifier	Line Color Spe		Specifier
red	r	blue	b	m	agenta		m	black		k
green	g	cyan	с	ye	ellow	v y		white		W
		-	Line Style solid (default) dashed		Specifie	Line Style       dotted       dash-dot		tyle ot	Specifier	
		Marker	Specifier	N	larker	Spe	ecifier	Marker		Specifier
		plus sign	+	a	sterisk		*	square		S
circle		circle	o p		oint	•		diamond		d
			-							

6. Plotting

#### Plots in MATLAB

- "plot" command
  - With 'line specifiers', labels, title

![](_page_27_Figure_4.jpeg)

'FontWeight', 'bold', 'Color', 'r')

### 6. Plotting

### Plots in MATLAB

- "plot3" command
  - >> t = 0:pi/50:10\*pi;
    >> st = sin(t);
    >> ct = cos(t);
  - >> figure
    >> plot3(st,ct,t)

![](_page_28_Figure_5.jpeg)

## 6. Plotting

### Meshgrid(-1:0.5:1);

### Plots in MATLAB

"surf" command

ſ	x ×						15	y ×				
	5x5 double							5x5 double				
	1	2	3	4	5			1	2	3	4	5
1	-1	-0.5000	0	0.5000	1	*	1	-1	-1	-1	-1	-1
2	-1	-0.5000	0	0.5000	1		2	-0.5000	-0.5000	-0.5000	-0.5000	-0.5000
3	-1	-0.5000	0	0.5000	1		3	0	0	0	0	C
4	-1	-0.5000	0	0.5000	1		4	0.5000	0.5000	0.5000	0.5000	0.5000
5	-1	-0.5000	0	0.5000	1		5	1	1	1	1	1
						- 1 March 1						

0.5

0

-0.5

5

0

-5

-10

-10

>> k = 6;

>> n = 
$$2^{k-1}$$
;

- >> [x,y,z] = sphere(n);
- >> figure
- >> surf(x,y,z);

![](_page_29_Figure_10.jpeg)

[x,y]=meshgrid(-8:0.5:8); r=sqrt(x.^2+y.^2)+eps;

z=sin(r)./r; surf(x,y,z);

![](_page_29_Figure_13.jpeg)

-5

### 7. User-defined Functions and Function Files

### User defined function

Arguments

![](_page_30_Figure_3.jpeg)

- To create user-defined function
  - Similar with script files

![](_page_30_Figure_6.jpeg)

### 7. User-defined Functions and Function Files

### User defined function

- Simple example of function
  - test.m

```
function [result1 result2]=test(a)
result1=a+100;
result2=a+200;
```

- Command Window
  - [a b]=test(100)

![](_page_31_Figure_7.jpeg)

### 8. Anonymous Functions

#### Anonymous Functions

 Users can define functions in the Command Window, within a script file or inside a userdefined function

![](_page_32_Figure_3.jpeg)

### 9. Function Functions

### Function function

- Function that accepts a <u>function handle</u> as an input argument
- Function handle
  - Ex) cosHandle=@cos

```
fhandle=@(x)(2*x);
fplot(fhandle,[0.0,1.0])
hold on
fhandle=@(x)(sin(1.0/x));
fplot(fhandle,[0.0001,1.0])
ylim([-2,2])
```

![](_page_33_Figure_6.jpeg)

### **10. Programming in MATLAB**

### **\*** Relational and logical operators

< Less than >= Greater than or equal to	Relational Operator	Description	Relational Operator	Description	>> 4==5	ans =
	<	Less than	>=	Greater than or equal to	>> 5>10	ans = (

>> 0 | | 1 ans = 1

Logical operator	Name	Description	>> 0&1 ans = 0
& Example: A&B	AND	Operates on two operands (A and B). If both are true, the result is true (1); otherwise the result is false (0).	>> 1&5 ans = 1
 Example: A B	OR	Operates on two operands (A and B). If either one, or both are true, the result is true (1); otherwise (both are false) the result is false (0).	>> 1  2 ans = 1
~ Example: ~A	NOT	Operates on one operand (A). Gives the opposite of the operand. True (1) if the operand is false, and false (0) if the operand is true.	>> ~(1  0) ans = 0

### **10. Programming in MATLAB**

### Conditional Statements, if-else Structures

- if-end
- if-else-end
- if-elseif-else-end
- if-elseif-elseif...-else-end
- In a script file,

```
function test(a)

if(a<10)
    'less than 0'
elseif(a<10)
    'greater than 0, less than 10'
elseif(a<100)
    'greater than 10, less than 100'
else
    'greater than or equal to 100'
end</pre>
```

![](_page_35_Figure_8.jpeg)

![](_page_35_Figure_9.jpeg)

## **10. Programming in MATLAB**

### Loops

• for-end

```
function circle(r)
n=100;
for i=1:n
    theta=2*pi/n*i;
    x(i)=r*cos(theta);
    y(i)=r*sin(theta);
end
```

plot(x,y)

![](_page_36_Figure_5.jpeg)

![](_page_36_Figure_6.jpeg)

![](_page_36_Figure_7.jpeg)