Tutorial of MATLAB

Tsien Dragon

Department of Mathematics

National University of Singapore

August 27, 2018

qian.lilong@u.nus.edu

Contents

1 Introduction to MATLAB

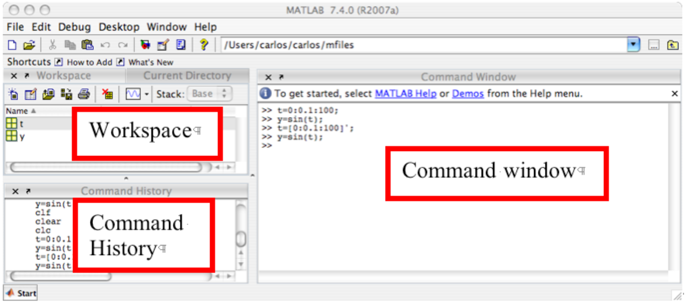
- MATLAB: Matrix Laboratory
- multi-paradigm numerical computing system and proprietary programming language
- Object-oriented programming and procedure languages
- Developed by MathWorks Inc. in USA
- Alternatives: Octave
- Nus student software:
  `https://nusit.nus.edu.sg/services/software_and_os/software/software-student/#install-matlab`
- Octave:
  `https://www.gnu.org/software/octave/`

2 Starting the software

1. Login with your NUS Net ID:`NUSSTU\*******` and corresponding password

Figure 1: MATLAB Icon

Double click the MATLAB icon



To get started, select MATLAB Help or Demos from the Help menu.

```
>> t=0:0.1:100;
>> y=sin(t);
>> t=[0:0.1:100]';
>> y=sin(t);
>>
```

```
y=sin(t
clf
clear
clc
t=0:0.1
y=sin(t
t=[0:0.
y=sin(t
```

Figure 2: MATLAB Interface

3 Basic statements

- Single statements
- assign to a variable
- Usage of semicolon”;”
- variable ”ans”
- Comments ”%”
- Rules for name variable: start by a letter, including letters, numbers and underscore ”_”. Case sensitive

```
 >> 1+2
>> a = 1+2
>> 1+2;
>> a = 1+2;
>> % a = 1+2;
>> A = 1+2;
>> 1A = 1+2;
>> A1 = 1+2;
>> A_1 = 1+2;

```

4 Variable

4.1 real number and complex number

- real number : Ex 1,1.1,1.1e+1(1.1×10^1),1.1e-1(1.1×10^{-1}),pi(π =3.1415...)
- Complex number
  Default imaginary unit i or j, which is $\sqrt{-1}$ Ex 1+i,1-i,1+j,1-j,(1+j)’

For operation of complex number:https://www.mathworks.com/help/matlab/complex-numbers.html

4.2 vector and matrix

- Row vector: a,b
- Column vector: c,d,e
- Matrix: A,B,C (`Vector can be regarded as a special matrix`)
- Special matrix: 0,I,1

```
    a = [1,2,3];
    b = [1 2 3];
    c = [1;2;3];d = a';
    e = transpose(b);
    A = [1 2;3 4];
    B = [1,2,3;4,5,6];
    C = B';
    O = zeros(4,3);
    I = eye(4,4);
    one = ones(5,5);
```

4.3 Special variable

symbol	pi	1	0	true
meaning	$\pi = 3.14\ldots$	default double 1 or ”true”	default 0 or ”false”	logical 1
symbol	false	inf	-inf	NaN
meaning	logical 0	∞	$-\infty$	non a number: $\frac{0}{0}$

5 Operators

5.1 Arithmetic Operators

Symbol	+	-	*	/	\	^
Example	1+2	1-2	1*2	1/2	1\ 2	2^ 2
Result	2	-1	2	0.5	2	4

Table 1: Arithmetic Operators

Refs. `https://www.mathworks.com/help/matlab/matlab_prog/matlab-operators-and-special-characters.html`

5.2 Relation operators

Symbol	==	∼ =	>	>=	<	<=
Example	1==2	1∼=2	2>2	2>=2	2<2	2<=2
Result	0	1	0	1	0	1

Table 2: Relation Operators

Note that here “1” is of the logical type, means “true” and “0” means the logical value“false”. See the detail of variables by “who var”.

5.3 Logical operators

symbol	|	&	-
meaning	Or	And	Not
Ex	1|0	1& 0	∼ 0
Result	1	0	1
Equal exp.	or(1,0)	and(1,0)	not(0)
Another exp.	1||0	1&& 0	∼ 0

$abs(x)$	$sqrt(x)$	sign(x)	sin(x)	cos(x)	tan(x)	cot(x)	sec(x)	csc(x)
|x|	$\sqrt{x}$	signum function	sin(x)	cos(x)	tan(x)	cotangent of x	secant of x	The cosecant of x

Table 3: basic function

$asin(x)$	$acos(x)$	atan(x)	acot(x)	asec(x)	acsc(x)
$\sin^{-1}$(x)	$\cos^{-1}$(x)	$\tan^{-1}$(x)	$\cot^{-1}$(x)	$\sec^{-1}$(x)	$\csc^{-1}(x)$

Table 4: Inverse Trigonometric Functions

6 Some commands

cmd:	clc	clear a	clear all	1:3	1:2:3
Result	clear screen	clear variable a	remove all variables	row vector [1,2,3]	row vector [1,3]
cmd:	who a	whos	clf	help cmd	doc cmd
Results:	see detail of variable a	see the details of all variables	clear the graph window	see help information of cmd	see document details of cmd

7 Math functoin

8 Command window display output format

- format short (default): display 4 digits
- format long: display 15 digits
- format short e (format shorte): scientific notation with 4 digits
- format long e: Short scientific notation with 15 digits
- format long g: scientific notation with a total of 15 digits for double values, and 7 digits for single values.
- format rat: Ratio of small integers.
- format compact:Suppress excess blank lines to show more output on a single screen.
- format loose:Add blank lines to make output more readable.

9 matrix operation

- Input matrix: A=[1,2,3;4,5,6]; or A(1,1)=1,...A(2,3)=5;
- Get the size: [n1,n2] = size(A); n1:row length, n2: column length.
  length(A) gets the row length of A.

syntax	$exp(x)$	$log(x)$	log2(x)	log10(x)
value	e^x	$\log_e(x)$	$\log_2$(x)	$\log_{10}$(x)

Table 5: Exponential and Logarithm Functions

- Increase the matrix: or A(1,4)=1;A(2,4) = 2;
- Matrix concatenation: row concatenation,A= [B,C] if column length equals. For example, B = [1;2];C=[3;4];
  column concatenation, A= [B;C] if row length equals. For ex. B = [1,2];C = [3,4];

10 matrix indexing

A is a matrix of size m×n

- A(i,j) : (i,j)-th entry of A
- A(i,:) : i-th row of A
- A(:,j): j-th column of A
- A(end,:): last row of A
- A(:,end-1); second last column of A
- A(a:b,c:d): submatrix of A from a to b row and c to d column.
- A(e,f)(e,f are two vectors): sub matrix of A row indexing in e, column indexing in f.
  For Ex. A = eye(5);e = [1,3,5];f=[3,4];A(e,f)

Note that index should not exceed the size of the matrix. A vector can be regarded as a matrix.

11 Matirx computation

matrix operation	
A+B	matrix addition
A-B	matrix subtraction
t * A	scalar-matrix,$t \in R$
A * B	matrix multiplication
A^n	A*A*A...*A, n times
A\ B	inv(A)*B
A/B	B*inv(A)
matrix entrywise operation	
A.+B	=A+B
A.-B	=A-B
t.*A	=t*A
A. * B	A(i,j)*B(i,j)
A.^n	A(i,j).^ n
A.\ B	$\frac{B(i,j)}{A(i,j)}$
A./B	$\frac{A(i,j)}{B(i,j)}$

12 related operation of matrix

- A’: conjugate transpose of A
- A.’: transpose of A
- det(A): determinant of a square matrix A
- rank(A): rank of a square matrix A
- eig(A): eigenvalues of a square matrix A
- inv(A):inverse of a square nonsingular matrix A

13 Logical flow of programming

14 Conditional statements

If statement

```
a=1;b=2;
if a>b
           y=a;
else
           y=b;
end
```

Find the largest number of {a,b}.

```
a=1;b=2;c=3;%
if a>b
   if a>c
     y=a;
   else
     y=c;
   end
elseif b>c
   y=b;
else
   y=c;
end
```

Find the largest number of {a,b,c}.
switch statement

15 Loops

- While loop:

```
k=0;
y=0;
while k<10
k=k+1;
y=y+k;
end
```

$y = \sum_{i=1}^{10} i$

- for loop

```
y=0;
for  k=1:10
y=y+k;
end
```

$y = \sum_{i=1}^{10} i$

16 Termination of loops

- break exits from the innermost loop

```
k=0;
y=0;
while 1
  k=k+1;
  y=y+k;
  if k==10
     break
  end
end
```

$y = \sum_{i=1}^{10} i$

```
A= [1,2,3;4,5,6];[n1,n2]=size(A);
y=0;
for i=1:100
  for j=1:100
    y = y+ A(i,j);
    if j==n2
      break
    end
  end
  if i==n1
    break
  end
end
```

$y = \sum_{i,j}^{n1,n2} A_{ij}$

17 Termination of loops

- break exits from the innermost loop
- return exists the scripts or function

```
y=0;
for  k=1:10
  if mod(k,2)==0 % if k is even
    continue
    %% if condition holds, then pass to
    % next loop without executing the
    % following statements in the loop
  end
    y=y+k;
end
```

The sum of odd numbers from 1 to 10.

```
A = randi(2,10,20)-1;
% create a matrix with 0 or 1
for i=1:10
  for j=1:20
     if A(i,j)==0;
      y=1; disp(A(i,j)); % display this variable
       return
     end
  end
end
```

Check if A has a zero entry.