

INTRODUCTION TO MATLAB PROGRAMING

SIMPLE CALCULATION & PRECEDENCE

Operators	
+	บวก
-	ลบ
*	คูณ
/	หาร
^	ยกกำลัง

Precedence	
1 st	วงเล็บ เริ่มจากวงเล็บในสุดก่อน
2 nd	ยกกำลัง จากซ้ายไปขวา
3 rd	คูณและหาร จากซ้ายไปขวา
4 th	บวกและลบ จากซ้ายไปขวา

Example 1

$3^2-5-6/3*2 \Rightarrow \underline{3^2}-5-6/3*2 \Rightarrow 9-5-\underline{6/3}*2$
 $\Rightarrow 9-5-\underline{2*2} \Rightarrow \underline{9-5}-4 \Rightarrow \underline{4}-4 \Rightarrow 0$

$3^2-5-6/(3*2) \Rightarrow 3^2-5-6/\underline{(3*2)}$
 $\Rightarrow \underline{3^2}-5-6/6 \Rightarrow 9-5-\underline{6/6} \Rightarrow \underline{9-5}-1 \Rightarrow \underline{4}-1 \Rightarrow 3$

VARIABLES

- การตั้งชื่อตัวแปรต้องประกอบด้วยตัวอักษรตัวเล็ก, ตัวใหญ่, ตัวเลข หรือ “_” โดยต้องขึ้นต้นด้วยตัวอักษรเท่านั้น และ ถือว่าอักษรตัวเล็กและตัวใหญ่ต่างกัน
- หากชื่อตัวแปรยาวมากกว่า 31 ตัวอักษร ตัวอักษรตัวที่ 32 เป็นต้นไปจะถูกตัดทิ้ง

VARIABLES

- คำที่ห้ามนำมาตั้งเป็นชื่อตัวแปร คือ for, end, if, while, function, return, elseif, case, otherwise, switch, continue, else, try, catch, global, persistent, break
- ไม่ควรนำชื่อฟังก์ชัน และ ตัวแปรพิเศษของ MATLAB มาตั้งเป็นชื่อตัวแปร โดยตัวแปรพิเศษของ MATLAB มีดังนี้ ans, beep, pi, eps, inf, NaN, nan, i, j, nargin, nargout, realmin, realmax, bitmax, varargin, varargout

SPECIAL VARIABLES

ans	คำตอบล่าสุด	eps	ระยะห่างที่น้อยที่สุดระหว่าง 2 จำนวน = 2.2204e-016
pi	ค่าคงที่ π = 3.14159265358979		
i,j	หน่วยจินตภาพ	realmin	ตัวเลข floating point บวกที่น้อยที่สุด = 2.2251e-308
inf	ค่าที่เกิดจากการหารด้วย 0 (Infinity)	realmax	ตัวเลข floating point บวกที่มากที่สุด = 1.7977e+308
NaN,nan	การคำนวณที่หาค่าไม่ได้ (Not a Number) เช่น 0/0 หรือ inf-inf	bitmax	ตัวเลขจำนวนเต็ม floating point ที่มากที่สุด = 9.007199254740991e+015
varargin	cell ของ input argument	clock	ใช้บอก ปี-เดือน-วัน-ชั่วโมง-นาที-วินาที
varargout	cell ของ output argument		
nargin	จำนวน input argument	date	วันที่ ในรูปแบบ วัน-เดือน-ปี
nargout	จำนวน output argument		

VARIABLES

Example 2

tile_length = 1
tile_width = 0.5
tile_area = tile_length*tile_width

floorarea = 1
FLOORAREA = 2
FloorArea = 3

TileCost_1 = (15*(floorarea/tile_area))+50
TileCost_2 = (15*(FLOORAREA/tile_area))+50
TileCost_3 = (15*(FloorArea/tile_area))+50

VARIABLES

Example 3

realmax
eps
clock
date

Example 4

pi
r = 1.2
pi*(r^2)
pi = 3
pi

pi*(r^2)
clear pi
pi
pi*(r^2)

VARIABLES

Example 5

2*2+5
ans
-2^2;
ans
2*ans;
ans

Example 6

z1 = -1+i
z2 = -1+j
z1*i
i = 2

z1*i
clear i
i
z1*i

ARRAYS & MATRICES

- การสร้าง array หรือ Matrix ใน MATLAB สามารถกำหนดได้ด้วย []
- ค่าระหว่างหลักจะถูกคั่นด้วย **space** หรือ ,
- การขึ้นแถวใหม่สามารถทำได้ด้วยการกด **Enter** หรือ คั่นด้วย ;

Example 7.1

```
a = [1,2,3]
a = [1 2 3]
```

```
b = [-1;0;1]
b = [-1
      0
      1]
```

```
C = [1,0,-1;0,1,0;0,0,2]
C = [1 0 -1;0 1 0;0 0 2]
C = [1 0 -1
      0 1 0
      0 0 2]
```

ARRAYS & MATRICES

- Array แบบไล่ตัวเลขสามารถกำหนดได้ดังนี้

Start:Step:Stop

- **Start** คือ ค่าตัวเลขเริ่มต้น
- **Step** คือ ระยะห่างระหว่างหลักถัดไปกับหลักปัจจุบัน
- **Stop** คือ การกำหนดให้ค่าสุดท้ายของการไล่ตัวเลข ไม่เลยค่า **Stop**

Example 7.2

```
d = 1:1:3
d = 1:3
```

```
e = 1:2:6
```

```
f = 1:2:5
```

```
g = 3:-1:1
```

```
h = 3:1
```

Start:Stop ⇒ Start:1:Stop

ARRAYS & MATRICES

- การอ้างตำแหน่งใน Array 1 มิติสามารถทำได้โดย

A(index)

- **A** คือ ชื่อ Array และ **index** คือ ตำแหน่งที่ต้องการ
- **index** สามารถเป็น Array ได้เมื่อต้องการอ้างตำแหน่งที่ต้องการมากกว่า 1 ตำแหน่ง
- การอ้างตำแหน่งใน Array n มิติสามารถทำได้โดย

A(index_1, index_2,..., index_n)

ARRAYS & MATRICES

Example 7.3

```
a(1)
b(2)
a(end)
b(end)
a(end-1)

a(1:2)
b(3:-1:2)
a([1 3])
a([1 3 2])
a([1 3 2 3])
b(2:end)
b(1:end)
b(:)
```

Example 7.4

```
C(3,2)
C(end,2)
C(end,end)

C(1:2,1)
C(1:2,2:3)
C([1 3],[1 3])
C(1,2:end)
C(1,1:end)
C(1,:)
C(:,1)
C([1 3],:)
C(:, [1 3])
C(:, :)
```

ARRAYS & MATRICES

Example 7.5

```
a(1,3)
b(2,1)

C(:)
C(5)
C(end)
C(1:8)
C(1:end)
```

Example 7.6

```
a'
[a;e]
[b a']
L = [C b;a 7]

L(1,1) = 5
L(2:3,2:3) = [-1 -2;-4 -6]
L(end,:) = 2:2:8
L([5 10 end]) = 10

save Example_07.mat
```

CHARACTER ARRAYS (STRINGS)

- การสร้างตัวแปรประเภทตัวอักษรสามารถระบุได้โดยใช้ ' '

Example 8

```
s1 = 'DSP'
s2 = ['D' 'SP']

s3 = '371'
[s1 s3]
['s1' s3]

s4 = 100
s5 = '100'
s4+1
s5+1

s1(1)
s1([1 3])
```

OPERATORS AND SPECIAL CHARACTERS

=	การกำหนดค่าตัวแปร	==	เท่ากับ	()	แสดงลำดับในการคำนวณ
+	บวก	~=	ไม่เท่ากับ		แสดงอินพุตในฟังก์ชัน
-	ลบ	>	มากกว่า		อ้างตำแหน่งในเมตริกซ์
*	คูณแบบเมตริกซ์	>=	มากกว่าหรือเท่ากับ	[]	สร้างเมตริกซ์
.*	คูณแบบอาร์เรย์	<	น้อยกว่า	{ }	สร้างหรืออ้างตำแหน่งใน cell
^	ยกกำลังแบบเมตริกซ์	<=	น้อยกว่าหรือเท่ากับ	' '	กำหนดตัวแปรประเภท String
.^	ยกกำลังแบบอาร์เรย์	&	และ	,	ใช้เป็นตัวแบ่งหรือตัวแบ่งหลัก
/	หารทางขวาแบบเมตริกซ์		หรือ	;	ใช้เป็นตัวแบ่งแถว
./	หารทางขวาแบบอาร์เรย์	~	นิเสธ		ไม่แสดงผลของบรรทัดนี้
\	หารทางซ้ายแบบเมตริกซ์	.	จุดทศนิยม	:	ใช้สร้างลำดับเลขคณิต
.\	หารทางซ้ายแบบอาร์เรย์		เรียกค่าของ field ใน structure	%	แสดง comment
'	transpose & conjugate	..	parent directory ใช้กับคำสั่ง cd	@	สร้าง function handle
.'	transpose	...	คำสั่งต่อในบรรทัดถัดไป	!	เรียกใช้คำสั่งของ OS

OPERATORS AND SPECIAL CHARACTERS

- Operator ที่มี . อยู่ข้างหน้า
เช่น .* ./ .\ .^ จะเป็น Array
Operator ซึ่งการคำนวณ
จะทำกับสมาชิกตำแหน่ง
ต่อตำแหน่ง
- Operator ที่ไม่มี . อยู่
ข้างหน้าเช่น * /\ ^ จะเป็น
Matrix Operator ซึ่งการ
คำนวณอิงตามหลักการ
ของ Matrix

Example 9

M1 = [1 2;3 4]
M2 = [-2 1;-1 2]

M1+M2
M1-1

2*M1
M1.*M2
M1*M2

M1./M2
M1/M2 →

M1.\M2
M1\M2 →

M1.^2
M1^2

GENERAL PURPOSE COMMAND

help	แสดงวิธีใช้คำสั่งหรือฟังก์ชัน
lookfor	ค้นหาชื่อคำสั่งหรือฟังก์ชัน
demo	เรียกใช้ Demo
clear	ลบค่าตัวแปรใน Workspace
save	บันทึกค่าตัวแปรใน Workspace
load	ดึงค่าจากไฟล์ .mat มาไว้ใน Workspace
clc	ลบ Command window
format	กำหนดรูปแบบการแสดงผลของตัวเลข
warning	แสดงหรือไม่แสดงคำเตือน
beep	ส่งเสียง Beep
quit	ออกจากโปรแกรม MATLAB

NUMBER DISPLAY FORMAT

format short (default)	เลข 5 หลัก	3.1416
format long	เลข 15 หลัก	3.14159265358979
format short e	เลข 5 หลักมีเลขยกกำลังฐาน 10	3.1416e+000
format long e	เลข 15 หลักมีเลขยกกำลังฐาน 10	3.14159265358979e+000
format bank	ทศนิยม 2 หลัก	3.14
format +	เครื่องหมาย (+,-)	+
format hex	เลขฐาน 16	400921fb54442d18
format rat	เศษส่วน (โดยประมาณ)	355/113

FUNCTIONS

- การเรียกใช้ Function ใน MATLAB สามารถทำได้โดย

F(input_1, input_2,...,input_n)

- **F** คือ ชื่อ Function และ **input** คือ ค่าอินพุตของ Function (จำนวนของตัวแปรอินพุตจะถูกกำหนดโดย Function)

Example 10.1

sqrt(2)

gcd(10,15)

max([1 5 6 2])

inv([1 2;3 4])

FUNCTIONS

- การกำหนดตัวแปรรับค่า Output ของ Function ทำได้โดย

out = F(input_1, input_2,...,input_n)

[out_1,out_2,...,out_n] = F(input_1, input_2,...,input_n)

[out_1 out_2 ... out_n] = F(input_1, input_2,...,input_n)

Example 10.2

out_a = sqrt(2)

out_b = mean([1 5 6 2])

out_c = inv([1 2;3 4])

[out_1,out_2] = size(out_c)

M-FILES

- เขียนในโปรแกรมประเภท Editor แล้วสั่งเก็บเป็นไฟล์ เพื่อนำมาสั่งทำงานได้ในภายหลังใน MATLAB
- MATLAB มีข้อบังคับว่าจะต้องระบุนามสกุลของไฟล์ เป็น **.m** และจะต้องเก็บไว้ในไดเรกทอรีของ MATLAB
- สามารถเรียกโปรแกรมไฟล์ที่เก็บเป็นชนิด **.m** นั้นใน
 - หน้าต่างรับคำสั่งของ MATLAB ด้วยการป้อนชื่อไฟล์นั้นลงไป
 - หรือสามารถเรียกใช้ไฟล์ **.m** จากไฟล์ **.m** อื่นก็ได้

M-FILES

- ข้อระวังก็คือ จะต้องบันทึกไฟล์นี้ไว้ในไดเรกทอรีที่ MATLAB สามารถมองเห็น หรือเรียกว่า **MATLAB Path** ซึ่งสามารถตรวจสอบได้จากคำสั่ง **path**
- หรืออาจจะเก็บไว้ใน ไดเรกทอรีปัจจุบันของ Window ก็ได้ ซึ่งจะแสดงออกมาเมื่อใช้คำสั่ง **pwd** หรือ **Tab Current Directory**
- ในกรณีที่ต้องการเก็บไว้ในไดเรกทอรีอื่น ๆ ก็สามารถเพิ่มชื่อเพิ่มลงไป ใน MATLAB Path โดยใช้คำสั่ง **addpath** (ดู help addpath)

M-FILES

- **Script File**

- โดย Script File เป็นชุดคำสั่ง

- **Function File**

- Function File เป็นการสร้างฟังก์ชันใหม่ ซึ่งโดยทั่วไปจะมีการส่งผ่านค่าตัวแปรเข้า และตัวแปรออก

SCRIPT FILE

- เราสามารถเก็บชุดคำสั่งที่ป้อนให้กับ MATLAB ไว้ในไฟล์ที่มีนามสกุล .m และสั่งให้ MATLAB ไปเรียกชุดคำสั่งจากไฟล์นี้เสมือนกับเราป้อนคำสั่งเข้าไปทีละคำสั่งได้
- การเรียกใช้ไฟล์ที่บันทึกไว้เป็นชุดคำสั่งที่ Command Window สามารถทำได้โดย พิมพ์ชื่อไฟล์โดยไม่ต้องมี .m แล้วกด Enter

SCRIPT FILE

Example 11

Editor

Comment → % Find x from $Ax = b$
clear A b x

A = [1 2 3; 1 0 -1; -2 3 -1];
b = [1; 0; -5];

x = A\b

Save as ex11.m

Command Window

ex11

FUNCTION FILES

- Function เป็น m-file ที่มีการ ‘ส่งค่า’ ไปยัง Function และ ‘รับค่า’ จาก Function
- Function มีการรับส่งค่าตัวแปรได้ script รับส่งตัวแปรไม่ได้
- Function ใช้ตัวแปรของตัวเองไม่รวมกับ Command Window หรือ Function อื่นๆ (ตัวแปรจะถูกสร้างใหม่ทุกครั้งที่มีการใช้ Function)

FUNCTION FILES

Example 12

Editor

```
function x = ex12(A,b)
```

```
    C = A^2;
```

```
    d = b/2
```

```
    x = C\b;
```

Save as ex12.m

Command Window

```
L=[1 2;3 4];
```

```
v = [1 ; -1];
```

```
ex12(L,v)
```

FUNCTION FILES

Example 13

Editor

```
function [x,y] = ex13(a,b)
```

```
    x = (a+b)/2;
```

```
    y = sqrt(a*b);
```

Save as ex13.m

Command Window

```
ex13(5,3)
```

```
[out1,out2]=ex13(5,3)
```

CONTROL FLOW

IF-ELSE

```
if (condition 1)
    (command 1)
elseif (condition 2)
    (command 2)
.
.
.
else
    (command n)
end
```

SWITCH-CASE

```
switch (variable)
case (value 1)
    (command 1)
case (value 2)
    (command 2)
.
.
.
otherwise
    (command n)
end
```

TRY-CATCH

```
try
    (command 1)
catch
    (command 2)
end
```

CONTROL FLOW

Example 14

Editor

```
x = input('x = ');

if x == 0
    disp('command 1')
elseif x> 0 & x <= 3
    disp('command 2')
elseif x > 3
    disp('command 3')
elseif x >= -3
    disp('command 4')
else
    disp('command 5')
end
```

Save as ex14.m

Command Window

```
ex14

Try
x = 0
x = 1
x = 5
x = -2
x = -5
```

CONTROL FLOW

Example 15

Editor

```
x = input('x = ');

switch x
case 1
    disp('command 1')
case {2,3}
    disp('command 2')
case 'abc'
    disp('command 3')
case {'de','f'}
    disp('command 4')
otherwise
    disp('command 5')
end
```

Save as ex15.m

Command Window

ex15

```
Try
x = 1
x = 2
x = '2'
x = 'a'
x = 'abc'
x = 'f'
```

CONTROL FLOW

Example 16

Editor

```
function y = ex16(x)

try
    y = x+[1 2 3];
catch
    y = 0
end
```

Save as ex16.m

Command Window

```
ex16(1)
ex16(2)
ex16([1 0 1])
ex16([3 2 1])
ex16([1 2;3 4])
```


CONTROL FLOW

FOR LOOP

```
for (variable)
    (command)
end
```

WHILE-LOOP

```
while (condition)
    (command)
end
```

break	ออกจากลูปด้านในสุด 1 ลูป
pause	หยุดการทำงานจนกว่าจะมีการกดคีย์

CONTROL FLOW

Example 17

Editor

```
n = input('n = ');
s = 0;
for k = 1:n
    s = s+k;
end
s
```

Save as ex17.m

Command Window

ex17

Try
n = 3
n = 1
n = 0
n = -1

CONTROL FLOW

Example 18

Editor

```
n = input('n = ');
s = 0;
k = 1;
while k <= n
    s = s+k;
    k = k+1;
end
s
```

Save as ex18.m

Command Window

ex18

Try
n = 3
n = 1
n = 0
n = -1

INPUT AND OUTPUT FUNCTIONS

input	แสดงตัวอักษร โดยจะรออินพุตจากคีย์บอร์ดแล้วเก็บค่าไว้
disp	แสดงรายละเอียดของตัวแปร หรือข้อความต่าง ๆ

LOGICAL FUNCTIONS

xor	<i>exclusive or</i>
all	ตรวจสอบว่าสมาชิกทุกตัวไม่เป็น 0
any	ตรวจสอบว่ามีสมาชิกอย่างน้อย 1 ตัวไม่เป็น 0
find	หาตำแหน่งที่ไม่เป็น 0
isempty	ตรวจสอบว่าเมตริกซ์มีขนาดเท่ากับ 0 หรือไม่
exist	ตรวจสอบว่ามีตัวแปรหรือฟังก์ชันชื่อเดียวกับที่ระบุหรือไม่

LOGICAL FUNCTIONS

Example 19

```
all([1 1 1 1 1])
all([1 0 1 1 1])

any([1 0 0 0 0])
any([0 0 0 0 0])

A = [2 5 3 9];
all(A>1)
all(A>3 & A<=10)
any(mod(A,2)==0)
```

```
B = [0 0 1 1;0 1 0 1];
all(B)
all(B,1)
all(B,2)
all(all(B))
all(B(:))

find([1 0 1 0])
find(A>3)
find(B)
[x,y] = find(B)
```

NUMERIC FUNCTIONS

fix	ปัดเลขให้เป็นจำนวนเต็ม โดยปัดเข้าหา 0
floor	ปัดเลขให้เป็นจำนวนเต็ม โดยปัดเข้าหา -∞
ceil	ปัดเลขให้เป็นจำนวนเต็ม โดยปัดเข้าหา +∞
round	ปัดเลขให้เป็นจำนวนเต็ม โดยปัดเข้าหาจำนวนเต็มที่ใกล้ที่สุด
abs	หาค่าสัมบูรณ์
sign	หาเครื่องหมาย (+,-)
rem	หาเศษจากการหาร
mod	<i>modulo</i>

COMPLEX NUMBER FUNCTIONS

real	หาส่วนจริง	conj	หา <i>conjugate</i>
imag	หาส่วนจินตภาพ	angle	หามุมเฟสเป็นเรเดียน

EXPONENTIAL AND LOGARITHM FUNCTIONS

sqrt	หารากที่ 2	log2	ค่า log ฐาน 2
exp	ยกกำลังฐาน e	sqrtm	หารากที่ 2 แบบเมตริกซ์
log	ค่า log ฐาน e	expm	ยกกำลังฐาน e แบบเมตริกซ์
log10	ค่า log ฐาน 10	logm	ค่า log ฐาน e แบบเมตริกซ์

TRIGONOMETRY AND HYPERBORIC FUNCTIONS

sin	sine	asin	arcsine	sinh	hyperboric sine	asinh	hyperboric arcsine
cos	cosine	acos	arccosine	cosh	hyperboric cosine	acosh	hyperboric arccosine
tan	tangent	atan	arctangent	tanh	hyperboric tangent	atanh	hyperboric arctangent
csc	cosecant	acsc	arccosecant	csch	hyperboric cosecant	acsch	hyperboric arccosecant
sec	secant	asec	arcsecant	sech	hyperboric secant	asech	hyperboric arcsecant
cot	cotangent	acot	arccotangen	coth	hyperboric cotangent	acoth	hyperboric arccotangent

↑
อินพุตเป็นเรเดียน

STATISTICTIC FUNCTIONS

min	หาค่าต่ำสุด	median	หามัธยฐาน	cumsum	หาผลรวมสะสม
max	หาค่าสูงสุด	sort	เรียงข้อมูล	cumprod	หาผลคูณสะสม
range	หาพิสัย	sum	หาผลรวม	diff	หาผลต่างระหว่างสมาชิกตัวที่ติดกัน
mean	หาค่าเฉลี่ย	prod	หาผลคูณ	std	หาส่วนเบี่ยงเบนมาตรฐาน

STRING FUNCTIONS

str2num	แปลงตัวอักษรเป็นตัวเลข	bin2dec	แปลงเลขฐาน 2 เป็น ฐาน 10
num2str	แปลงตัวเลขเป็นตัวอักษร	dec2bin	แปลงเลขฐาน 10 เป็น ฐาน 2
strcmp	เปรียบเทียบตัวอักษร	hex2dec	แปลงเลขฐาน 16 เป็น ฐาน 10
strcmppi	เปรียบเทียบตัวอักษร โดยถือว่า ตัวอักษรตัวเล็กกับตัวใหญ่เหมือนกัน	dec2hex	แปลงเลขฐาน 10 เป็น ฐาน 16
		base2dec	แปลงเลขฐานที่กำหนดเป็น ฐาน 10
strfind	หาดำแหน่งตัวอักษรตามที่กำหนด	dec2base	แปลงเลขฐาน 10 เป็นฐานที่กำหนด

STATISTICTIC FUNCTIONS

Example 20

A = [2 5 3 9];
min(A)
max(A)
mean(A)
sum(A)
prod(A)

sort(A)

diff(A)

B = [1 2 1 3;2 4 1 0];
min(B)
min(B,1)
min(B,2)
min(min(B))
min(B(:))

sum(B)
sum(B,1)
sum(B,2)
sum(sum(B))
sum(B(:))

MATRIX FUNCTIONS

zeros	สร้างเมตริกซ์ที่มีสมาชิกเป็น 0 ทั้งหมด	length	หาความยาวของเมตริกซ์
ones	สร้างเมตริกซ์ที่มีสมาชิกเป็น 1 ทั้งหมด	size	หาขนาดของเมตริกซ์
eye	สร้างเมตริกซ์เอกลักษณ์	fliplr	กลับเมตริกซ์ในทิศทาง ซ้าย-ขวา
linspace	สร้างเวกเตอร์ของลำดับเลขคณิต	flipud	กลับเมตริกซ์ในทิศทาง บน-ล่าง
logspace	สร้างเวกเตอร์ของลำดับเรขาคณิต	flipdim	กลับเมตริกซ์ในทิศทางที่กำหนด
rand	สร้างเมตริกซ์ที่มีสมาชิกได้จากการสุ่มค่า 0-1 ที่มีการกระจายแบบ Uniform	rot90	หมุนเมตริกซ์ $\pm 90^\circ, \pm 180^\circ, \dots$
		tril	สร้างเมตริกซ์สามเหลี่ยมล่าง
randn	สร้างเมตริกซ์ที่มีสมาชิกได้จากการสุ่มค่าที่มีการกระจายแบบ Normal มีค่าเฉลี่ยเป็น 0 มีส่วนเบี่ยงเบนมาตรฐานเป็น 1	triu	สร้างเมตริกซ์สามเหลี่ยมบน
		det	หาดีเทอร์มิแนนต์ของเมตริกซ์
		inv	หาเมตริกซ์ผกผัน
diag	สร้างเมตริกซ์ทแยง	pinv	หาเมตริกซ์ผกผันเสมือน
reshape	เปลี่ยนขนาดของเมตริกซ์	rank	หาเรงค์ของเมตริกซ์

MATRIX FUNCTIONS

Example 21

zeros(3,5)

ones(1,7)

eye(5)

diag([1 2 3 4])

linspace(0,pi,20)

rand(1,7)

randn(1,7)

B = [1 2 1 3;2 4 1 0];

length(B)

size(B)

reshape(B,4,2)

C = [1+i 1+2i ; 3-i 2];

C'

transpose(C)

SPARSE MATRIX FUNCTIONS

sparse	สร้าง <i>sparse matrix</i> ที่เป็น เมตริกซ์ศูนย์ หรือ แปลงเมตริกซ์ธรรมดาเป็น <i>sparse matrix</i>	spones	เปลี่ยนสมาชิกในตำแหน่งที่ไม่เท่ากับ 0 ใน <i>sparse matrix</i> ให้เท่ากับ 1
speye	สร้าง <i>sparse matrix</i> ที่เป็น เมตริกซ์เอกลักษณ์	full	แปลง <i>sparse matrix</i> เป็นเมตริกซ์ธรรมดา

SET FUNCTIONS

unique	ตัดสมาชิกตัวที่ซ้ำกันออกพร้อมทั้งเรียงข้อมูล	setdiff	<i>set difference</i>
union	<i>set union</i>	setxor	<i>set exclusive union</i>
intersect	<i>set intersection</i>	ismember	ตรวจสอบการเป็นสมาชิก

POLYNOMIAL FUNCTIONS

roots	หารากของพหุนาม	residue	หา <i>partial fraction</i>
poly	หาพหุนามจากราก	polyder	หาอนุพันธ์ของพหุนาม
polyval	หาค่าของพหุนาม	polyint	หาปริยานุพันธ์ของพหุนาม
conv	คูณพหุนาม	polyfit	หาพหุนามที่จากคู่อันดับที่กำหนด (ประมาณ)
deconv	หารพหุนาม		

SET FUNCTIONS

Example 22

A = [2 3 4 2 4 3 1];
B = [0 3 5 3 2];

unique(A)

union(A,B)

intersect(A,B)

setdiff(A,B)

setdiff(B,A)

ismember(A,B)

CELL FUNCTIONS

cell	สร้าง cell ว่าง	iscell	ตรวจสอบว่าเป็นตัวแปรประเภท cell หรือไม่
deal	กำหนดค่าตัวแปรได้ที่หลายตัว	num2cell	หาอนุพันธ์ของพหุนาม

STRUCTURE FUNCTIONS

struct	สร้าง structure	isfield	ตรวจสอบว่ามีชื่อ field ที่กำหนดหรือไม่
fieldnames	เรียกดูชื่อ field ใน structure	isstruct	ตรวจสอบว่าเป็นตัวแปรประเภท structure หรือไม่
getfield	เรียกค่าของ field ใน structure		
setfield	กำหนดค่าของ field ใน structure	struct2cell	แปลง structure เป็น cell
rmfield	ลบ field	cell2struct	แปลง cell เป็น structure

SYMBOLIC FUNCTIONS

sym,syms	สร้างตัวแปรประเภท <i>symbolic</i>	compose	หา <i>composite</i> ของ <i>function</i>
subs	แทนค่าตัวแปร <i>symbolic</i>	diff	หาอนุพันธ์
simplify	เปลี่ยนให้อยู่ในรูปที่สั้นที่สุด	int	หาปริพันธ์
expand	กระจายพจน์ออกมา	limit	หาขีดจำกัด
numden	หาเศษ และ ส่วน	fourier	<i>Fourier Transform</i>
solve	แก้สมการ	laplace	<i>Laplace Transform</i>
dsolve	แก้สมการเชิงอนุพันธ์	ztrans	<i>Z Transform</i>
finverse	หา <i>inverse</i> ของ <i>fnction</i>	taylor	<i>Taylor Series</i>

GRAPH DISPLAY FUNCTIONS

plot	การพล็อตกราฟเชิงเส้น <i>x-y</i>	polar	พล็อตกราฟในรูปเชิงขั้ว
axis	กำหนดลักษณะแกน	pie	สร้างแผนภูมิวงกลม
grid	แสดง <i>grid</i> หรือไม่	bar	สร้างแผนภูมิแท่ง
hold	คงรูปเดิมอยู่หรือไม่	stem	พล็อตในรูปขบวนอิมเพาส์
xlabel	กำหนดคำอธิบายบนแกน <i>x</i>	hist	สร้าง <i>histogram</i>
ylabel	กำหนดคำอธิบายบนแกน <i>y</i>	scatter	พล็อตจุดแบบกระจายตัว
title	กำหนดหัวข้อกราฟ	contour	สร้างเส้น <i>contour</i>
legend	กำหนดชื่อของกราฟแต่ละเส้น	fill	ระบายสีให้รูป <i>polygon</i>
text	พิมพ์ข้อความเพิ่มเติม	ezplot	พล็อตกราฟจากฟังก์ชัน
line	วาดเส้นตรงเพิ่มเติม	figure	เปิดหน้าต่างรูปภาพใหม่
semilogx	เหมือน <i>plot</i> แต่ค่าในแกน <i>x</i> เป็น <i>log</i> ฐาน 10	subplot	แบ่งหน้าต่างรูปภาพ
semilogy	เหมือน <i>plot</i> แต่ค่าในแกน <i>y</i> เป็น <i>log</i> ฐาน 10	clf	ลบรูปในหน้าต่างรูปภาพ
loglog	เหมือน <i>plot</i> แต่ค่าเป็น <i>log</i> ฐาน 10 ทั้ง 2 แกน	close	ปิดหน้าต่างรูปภาพ

GRAPHIC HANDLING FUNCTIONS

get	อ่าน <i>handle</i>	set	ตั้งค่า <i>handle</i>
-----	--------------------	-----	-----------------------

FILE I/O FUNCTIONS

fopen	เปิดไฟล์	fgetl	รับข้อมูลในบรรทัดต่อไป
fclose	ปิดไฟล์	fprintf	เขียนข้อมูลลงในไฟล์

IMAGE I/O FUNCTIONS

imread	รับข้อมูลประเภทรูปภาพ	imagesc	พล็อตรูปโดยการทำ <i>scaling</i> ข้อมูลก่อน
imwrite	สร้างไฟล์รูปจากข้อมูล	colormap	เปลี่ยนโทนสี
image	พล็อตรูป	colorbar	แสดงแถบสีเทียบค่า

AUDIO I/O FUNCTIONS

wavread	อ่านข้อมูลจากไฟล์ <i>.wav</i>	wavrecord	อัดเสียงจากอุปกรณ์อินพุต
wavwrite	สร้างไฟล์เสียงจากข้อมูล	wavplay	ส่งเสียงออกทางลำโพง