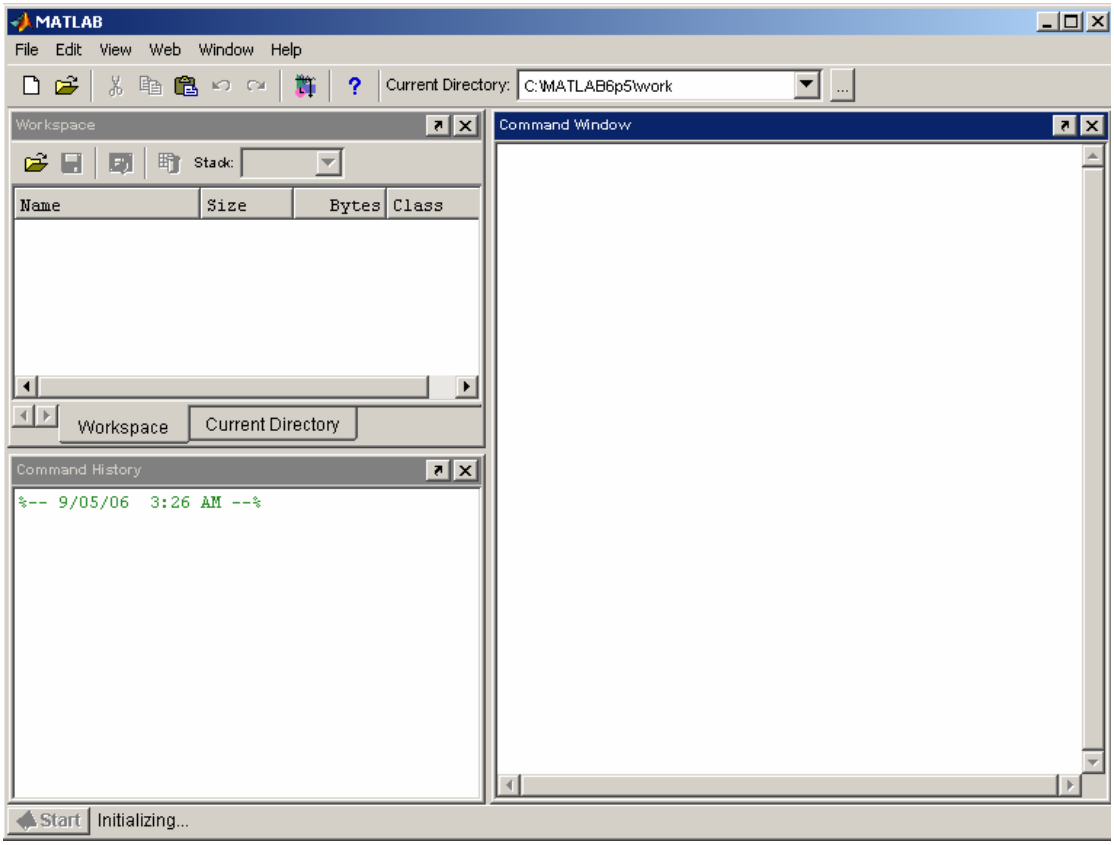
การใช้งาน MATLAB เบื้องต้น

โปรแกรม MATLAB – MATrix LABoratory เป็นโปรแกรมของ The Mathworks, Inc. (http://www.mathworks.com) ใช้สำหรับการคำนวณทางคณิตศาสตร์ ทั้งการคำนวณเชิงตัวเลข และการคำนวณเชิงสัญลักษณ์ (ใช้โปรแกรมเสริมชื่อ Maple) นอกจากนี้ชุดคำสั่งของ MATLAB ยังเป็นภาษาคอมพิวเตอร์ ที่มีความสามารถในการคำนวณ Matrix ซึ่งช่วยในการทำงานด้านวิทยาศาสตร์หลายด้านๆ โดยมี Toolbox สำหรับอำนวยความสะดวกในการทำงาน

1. เริ่มต้นใช้งานโปรแกรม MATLAB



เมื่อเข้าสู่โปรแกรม MATLAB จะปรากฏหน้าต่างดังรูป ซึ่งมีหน้าต่างย่อยคือ Workspace, Command Windows และ Command History

- Workspace หน้าต่างสำหรับแสดงตัวแปรที่กำลังใช้งานอยู่
- Command Windows หน้าต่างสำหรับป้อนชุดคำสั่ง
- Command History หน้าต่างแสดงชุดคำสั่งที่ใช้งานไปแล้ว

การใช้งานโปรแกรม MATLAB ทำโดยการป้อนชุดคำสั่งใน Command Windows เมื่อต้องการความช่วยเหลือ ให้พิมพ์

```
help [ชื่อคำสั่ง]
```

เพื่อดูวิธีการใช้งานคำสั่งต่างๆ

2. การใช้ MATLAB ในการคำนวณทั่วไป

การใช้ MATLAB ในการคำนวณทั่วไปสามารถป้อนคำสั่งเข้าไปใน Command Windows (ข้อความหลัง % เป็นคอมเม็นต์) ดังตัวอย่าง

```
1 + 2 + 3
2^3             % 2 ยกกำลัง 3
sin(pi/2)       % sin 90°
log10(100)      % log10(100)
pi              % ค่า π
22/7            % ไม่เท่ากับ pi
```

การสามารถใช้ตัวแปรในการเก็บผลลัพธ์การคำนวณได้ ดังตัวอย่าง

```
a = 2
b = 3 + 4
c = b/a
```

ถ้าไม่ต้องการให้ MATLAB แสดงผลการคำนวณ ให้ใช้ ";" ต่อท้ายคำสั่ง ดังตัวอย่าง

```
x = 10;
y = 20;
r = x + y;
```

สามารถใช้ "," และ/หรือ ";" ในการแยกคำสั่งที่เขียนบนบรรทัดเดียวกันได้

```
p = sin(pi/2), q = cos(pi/2); z = p * q
```

2.1 ตัวดำเนินการทางคณิตศาสตร์ใน MATLAB

เครื่องหมาย	ความหมาย
+	บวก
-	ลบ
*	คูณ
/	หาร
^	ยกกำลัง

2.2 ตัวแปรที่สำคัญ

ตัวแปร	ความหมาย
pi	ค่า π
i หรือ j	$\sqrt{-1}$
Inf	∞
NaN	ไม่ใช่จำนวน (ไม่นิยาม)
ans	ตัวแปรสำหรับเก็บผลลัพธ์การคำนวณ

2.3 ฟังก์ชันมูลฐาน

ดูรายละเอียดฟังก์ชันมูลฐานเพิ่มเติมโดยใช้คำสั่ง help elfun

ฟังก์ชันตรีโกณมิติ ใช้หน่วยเป็นเรเดียน หากต้องการแปลงจากองศาเป็นเรเดียนใช้ radian = degree*pi/180

sin(x), cos(x), tan(x), sec(x), csc(x) (cosec) และ cot(x)

อินเวอร์สฟังก์ชันตรีโกณมิติ

asin(x), acos(x), atan(x), asec(x), acsc(x) (inverse cosec) และ acot(x)

ฟังก์ชันไฮเพอร์โบลิก

sinh(x), cosh(x), tanh(x), sech(x), csch(x) (cosech) และ coth(x)

อินเวอร์สฟังก์ชันไฮเพอร์โบลิก

asinh(x), acosh(x), atanh(x), asech(x), acsch(x) (inverse cosech) และ acoth(x)

ฟังก์ชันเอ็กซ์โปเน็นเชี่ยนและลอการิทึม

log(x)	log ฐาน e หรือ ln
log10(x)	log ฐาน 10
exp(x)	e ยกกำลัง x
sqrt(x)	หารากที่สองของ x

ฟังก์ชันปัดเศษและหาเศษเหลือ

round(x)	แปลงเป็นตัวเลขเต็มแบบปัดทศนิยม
fix(x)	แปลงเป็นตัวเลขเต็มแบบตัดทศนิยมทิ้ง
floor(x)	หาจำนวนเต็มมากที่สุดที่น้อยกว่า x
ceil(x)	หาจำนวนเต็มน้อยที่สุดที่มากกว่า x
rem(x,y)	หาเศษเหลือจากการหาร x ด้วย y

ฟังก์ชันเกี่ยวกับจำนวนเชิงซ้อน (และจำนวนจริง)

abs(x) ค่าสัมบูรณ์ของ x

3. การใช้งาน Vector และ Matrix

การนิยาม Vector และ Matrix ใช้คำสั่งดังนี้

3.1 การนิยาม Vector (หรือ Matrix ขนาด 1 x N หรือ N x 1)

ใช้ [] ในการระบุถึง Vector และ แยกสมาชิกด้วย space " " หรือ ";"

$x = [R_1\ R_2\ \ldots\ R_N]$ % (Row vector)
$x = [R_1\ ;R_2\ ;\ \ldots\ R_N]$ % (Column vector)

ตัวอย่าง

```
x = [1 2 3]      % x is a row vector
y = [1;2;3]      % x is a column vector
```

3.2 การนิยาม Matrix

ใช้ [] ในการระบุถึง Matrix แยกแต่ละแถวด้วย ";" และ แยกสมาชิกในแต่ละแถวด้วย space " "

$x = [R_{11}\ R_{22}\ \ldots\ R_N;\ R_{21}\ R_{22}\ \ldots\ R_{2N};\ \ldots;\ R_{M1}\ R_{M2}\ \ldots\ R_{MN}]$

ตัวอย่าง

```
a = [1 2 3; 4 5 6; 7 8 9]
```

3.3 การอ้างอิงสมาชิก

การอ้างอิง หรือ ใช้งานค่าของสมาชิกใน Vector หรือ Matrix ใช้

x(n) % อ้างถึงสมาชิกตัวที่ n ของ Vector x (หรือ Matrix x ที่มีแถวเดียว)

x(m,n) % อ้างถึงสมาชิกตัวที่ n ในแถวที่ m ของ Matrix x

ตัวอย่าง

```
x(2)
y(3)
a(2,3)
b(3,2)
```

3.4 การดำเนินงานเกี่ยวกับ Vector/Matrix

สามารถใช้ฟังก์ชัน และ ตัวดำเนินการ +,- ในการดำเนินการกับ Vector/Matrix ได้

ตัวอย่าง

```
x = [0 pi/2 pi 3*pi/2 2*pi]
y = sin(x)        % หาค่า [sin(0), sin(pi/2), sin(pi), sin(3/2 pi)]
z = x + y         % หาค่าผลบวก ของ สมาชิกของ x และ y
```

3.4.1 การหา Transpose (Hermitian)

ใช้เครื่องหมาย " ' "

x' = transpose ของ x ในกรณีที่เป็น real vector/matrix

x'= Hermitian (Transpose conjugate) ของ x ในกรณีที่เป็น complex vector/matrix

ตัวอย่าง

```
x = [1  2 3]
xt = x'
y = [1 2 3; 4 5 6; 7 8 9]
yt = y'
```

3.4.2 การหา Scalar Product

กำหนด vector u และ v ดังนี้

$$\underline{u} = [u_1, u_2, \ldots, u_n], \qquad \underline{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

Scalar Product ของ u และ y คือ

$$\underline{u}\,\underline{v} = \sum_{i=1}^{n} u_i v_i.$$

ใช้เครื่องหมาย "*" ในการหา Scalar Product

```
u * v
```

(ใน matlab ต้องใช้ u เป็น row vector และ v เป็น column vector)

ตัวอย่าง

```
u = [3 6 9]
v = [2; 4; 6]
u * v
u * [2 4 5]'
```

3.4.3 การหา **Element-wise Product**

ให้ u และ v เป็น Vector, Element-wise Product ของ u และ v คือ

$$\underline{u} \cdot \underline{v} = [u_1 v_1, u_2 v_2, \ldots, u_n v_n]$$

(u และ v ต้องเป็น row vector หรือ column vector ทั้งคู่

ใช้ ".*" ในการหา Element-wise Product

ตัวอย่าง

```
u = [3 6 9]
v = [2 4 6]
w = u .* v
```

3.4.4 การหา **Matrix-Marix (Matrix-Vector) Product**

ใช้เครื่องหมาย "*" ในการหาผลคูณ Matrix ของ u และ v โดยมีเงื่อนไขว่าจำนวน column ของ u ต้องเท่ากับ จำนวนแถวของ v เช่น u มีขนาด m x n และ v มีขนาด n x p จะได้ u*v มีขนาด m x p

ตัวอย่าง

```
A = [1 2 3; 4 5 6]
B = [7; 8; 9]
C = A*B
```

ถ้า u เป็น 1xm row vector and v เป็น mx1 column vector, u*v จะเท่ากับ dot product (inner product) ของ u และ v

3.4.5 ฟังก์เกี่ยวกับ **Matrix**

det(X)	หา Determinant ของ X
eig(X)	หา Eigen Vector ของ X
inv(X)	หา Inverse ของ X

(ทั้ง 3 ฟังก์ชัน X ต้องเป็น Square Matrix:จำนวนแถว = จำนวนคอลมน์)

ตัวอย่าง

```
X = [1 2 3; 4 5 6;7 8 9]
w = det(X)
y = eig(X)
z = inv (X)
```

4. การวาดกราฟ

4.1 การวาดกราฟฟังก์ชันมูลฐาน

ในการวาดกราฟฟังก์ชันมูลฐาน เช่น y = sin(x) สำหรับ $0 \leq x \leq \pi$ เราจะใช้วิธีดึงกลุ่มตัวอย่างของจุดออกมาจำนวนหนึ่ง ที่มากพอจะทำให้เห็นรูปร่างของฟังก์ชัน แล้วเชื่อมแต่ละจุดด้วยเส้นตรง วิธีดึงกลุ่มตัวเราจะใช้ฟังก์ชัน linspace(a, b, n) เพื่อเลือก n จุดจากช่วง ระหว่างค่า a และ b ดังตัวอย่าง

```
X = linspace(0, pi, 100)
```

คำสั่งนี้จะเลือกจุดมา 100 จากช่วง 0 ถึง π โดยเก็บใน vector X และ เราจะหาค่า y ที่สัมพันธ์กันจะเก็บใน vector Y ได้ดังตัวอย่าง

```
Y = sin(X)
```

การวาดกราฟจะใช้คำสั่ง

```
plot(X,Y)
```

4.2 การวาดกราฟของเส้นตรงและข้อมูล

ในหัวข้อนี้เราจะกล่าวถึงการใช้ MATLAB ในการวาดกราฟของชุดข้อมูล เช่น

x_i	1	2	6	8	9
y_i	0.1	0.5	0.4	0.6	0.9

การป้อนข้อมูลเข้าไปใน MATLAB ทำดังนี้

```
x = [1 2 6 8 9]
y = [0.1 0.5 0.4 0.6 0.9]
```

การวาดกราฟ 2 มิติ/3 มิติจะใช้คำสั่ง plot, plot3 ซึ่งใช้งานดังนี้

```
plot(Y, [LineSpec])
plot(X, Y, [LineSpec])
plot3(X, Y, Z, [LineSpec])
```

LineSpec ใช้กำหนดค่าต่างของเส้นตรง (ดูเพิ่มเติมในหัวข้อ 4.2.1)

ตัวอย่าง

```
plot(x,y)
plot(x,y,'o')
```

```
plot(x,y,'r^-')
plot3(x,y,z)
plot3(x,y,z,'*--')
```

4.2.1 การกำหนด LineSpec

การกำหนดลักษณะของเส้นตรง

Specifier	Line Style
`-`	Solid line (default)
`--`	Dashed line
`:`	Dotted line
`-.`	Dash-dot line

การกำหนดลักษณะของจุด (ข้อมูล)

Specifier	Marker Type
`+`	Plus sign
`o`	Circle
`*`	Asterisk
`.`	Point
`x`	Cross
`'square' or s`	Square
`'diamond' or d`	Diamond
`^`	Upward-pointing triangle
`v`	Downward-pointing triangle
`>`	Right-pointing triangle
`<`	Left-pointing triangle
`'pentagram' or p`	Five-pointed star (pentagram)
`'hexagram' or h`	Six-pointed star (hexagram)

การกำหนดสี

Specifier	Color
r	Red
g	Green
b	Blue
c	Cyan
m	Magenta
y	Yellow
k	Black
w	White

5. การแก้สมการโดยใช้ MATLAB

5.1 การหารากของสมการพหุนาม

เราสามารถหารากของสมการพหุนามในรูป

$$a_0x^n + a_1x^{n-1} + ... + a_{n-1}x^1 + a_n = 0$$

โดยใส่สัมประสิทธิ์ของพหุนามใน vector และ แก้โดยใช้คำสั่ง roots

ตัวอย่าง

ต้องการแก้สมการ $x^2 - 5x + 6 = 0$
ใช้คำสั่งใน MATLAB ดังนี้

```
x = [1 -5 6]
roots(x)
```

5.2 การแก้ระบบสมการเชิงเส้น

ในการแก้ปัญหาทางด้านวิศวกรรม มักจะนำไปสู่การแก้ปัญหาสมการสมการเชิงเส้นอยู่เสมอ ในการแก้สมการเชิงเส้นนี้ เราจะเขียนปัญหาให้อยู่ในรูปทั่วไปของสมการเชิงเส้น ซึ่งเขียนในรูปของ matrix สัมประสิทธ์ **A**, vector **b** และ ตัวแปรไม่ทราบค่าใน vector **x** ดังนี้

$$A\mathbf{x} = \mathbf{b}$$

หรือเขียนแบบแจกแจงเป็นดังนี้

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 + \cdots a_{1,n}x_n &= b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots a_{2,n}x_n &= b_2 \\ &\vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \cdots a_{n,n}x_n &= b_n \end{aligned}$$

ถ้ามีจำนวนของสมการเชิงเส้นที่เป็นอิสระต่อกันเท่ากับจำนวนตัวแปร เราสามารถหาคำตอบของระบบสมการเชิงเส้นได้ คือ

$$\mathbf{x} = A^{-1}\mathbf{b}$$

ตัวอย่าง

ต้องการแก้ระบบสมการเชิงเส้น

$$x_1 + 2x_2 - x_3 = 1$$

$$-2x_1 - 6x_2 + 4x_3 = -2$$

$$-x_1 - 3x_2 + 3x_3 = 1$$

เขียนในรูป Matrix ได้เป็น

$$A = \begin{bmatrix} 1 & 2 & -1 \\ -2 & -6 & 4 \\ -1 & -3 & 3 \end{bmatrix}, b = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

ต้องการหาค่าของ vector $x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$ ที่ $Ax = b$

ใช้คำสั่งใน MATLAB ดังนี้

```
A=[1 2 -1; -2 -6 4; -1 -3 3]
b=[1; -2; 1]
x = inv(A) * b % หรือใช้ x = A\b
```