พื้นฐานการใช้งานโปรแกรม MATLAB

1. แนวคิด

ในปัจจุบันคอมพิวเตอร์ได้พัฒนาไปอย่างรวดเร็ว ในการศึกษาวิเคราะห์และออกแบบระบบควบคุมนั้นมีการคำนวณเพื่อวิเคราะห์และออกแบบระบบควบคุมที่สามารถคำนวณได้ด้วยมือแล้ว นอกจากนั้นยังมีโปรแกรมซึ่งสามารถที่ประยุกต์ใช้งานช่วยในการวิเคราะห์และออกแบบระบบควบคุมได้สะดวกรวดเร็ว และมีความถูกต้องแม่นยำ ทั้งนี้ผู้ใช้งานจำเป็นต้องทราบข้อจำกัดในการใช้งานบางประการรวมทั้งนำทฤษฎีที่ได้ศึกษามาช่วยในการแก้ปัญหาในการวิเคราะห์ออกแบบเชิงวิศวกรรมซึ่งในปัจจุบันมีโปรแกรมเป็นจำนวนมากที่สามารถใช้งานในงานด้านวิศวกรรมได้ แต่โปรแกรมพื้นฐานที่เป็นที่นิยมหนึ่งในนั้นคือ โปรแกรม MATLAB ในแต่ละเวอร์ชันมีการใช้งานที่แตกต่างกันอยู่บ้างแต่ในการทดลองนี้จะอ้างอิงเวอร์ชัน MATLAB 6.5 ซึ่งการทดลองจะประกอบด้วย 3 ใบงาน

1.1 พื้นฐานการใช้งานโปรแกรม MATLAB

2. วัตถุประสงค์

2.1 นักศึกษาสามารถเรียกใช้งานพื้นฐานการคำนวณของโปรแกรม MATLAB ได้

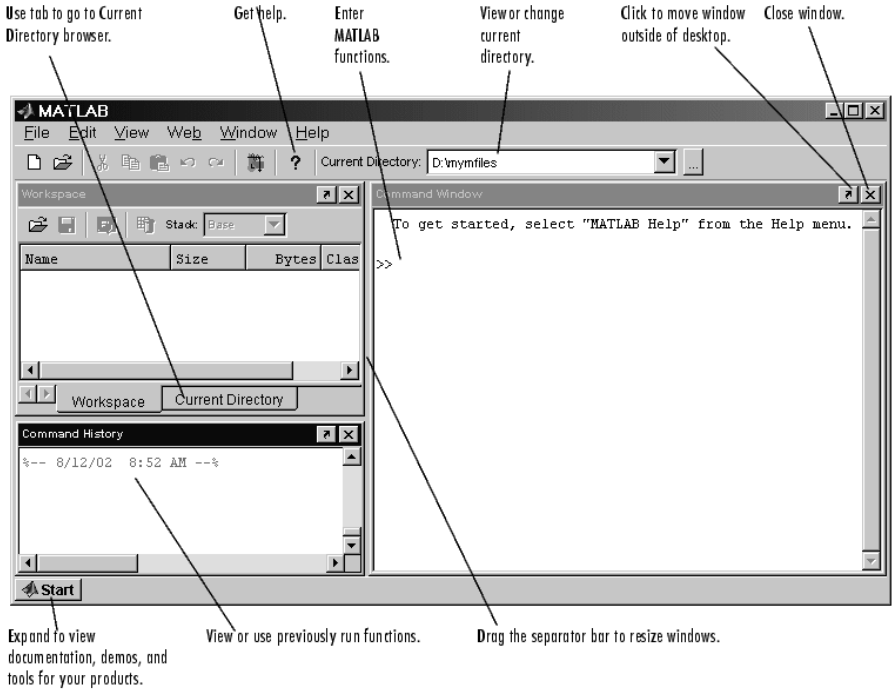
2.2 นักศึกษาสามารถสร้างและเรียกใช้สคริปและฟังก์ชันของโปรแกรม MATLAB ได้

2.3 นักศึกษาสามารถพล็อตกราฟและแก้ไขกราฟในโปรแกรม MATLAB ได้

3. ทฤษฎี

โปรแกรม MATLAB เป็นโปรแกรมที่มีความสำคัญโปรแกรมหนึ่งในการศึกษาและในการประยุกต์ใช้งานเป็นเครื่องมือในการคำนวณคณิตศาสตร์ในเชิงวิศวกรรม ซึ่งในโปรแกรมนี้เตรียมเครื่องมือไว้ให้ใช้ในหลายสาขาทางวิศวกรรม เช่น วิศวกรรมระบบควบคุม ก็ได้เตรียมเครื่องมือไว้ให้ใช้คือ Control Toolbox ซึ่งมีประโยชน์มากในการศึกษาวิชาระบบควบคุม สามารถที่ใช้ในการวิเคราะห์และออกแบบระบบควบคุม และสามารถจำลองระบบควบคุมก่อนที่จะทำการสร้างระบบ

จริง อีกทั้งสามารถลดเวลาและความสิ้นเปลืองในการวิเคราะห์และออกแบบระบบควบคุมทำให้งานวิจัยในปัจจุบันลดระยะเวลาโครงการในเวลาอันรวดเร็ว นอกจากนี้ยังมีเครื่องมือต่าง ๆ ที่เตรียมไว้ให้อีกหลายสาขาทางวิศวกรรม ซึ่งในการศึกษาโปรแกรมนี้จำเป็นต้องอาศัยทักษะหลายด้าน เช่น ระบบคำนวณเชิงเลข การโปรแกรมภาษา และ โจทย์ปัญหาทางวิศวกรรมที่ต้องการจะหาคำตอบ ซึ่งในการเริ่มต้นใช้งานโปรแกรมให้ดับเบิลคลิกที่ไอคอน  โปรแกรมจะเริ่มใช้งานดังรูปที่ 1 ซึ่งจะแนะนำลักษณะองค์ประกอบที่สำคัญในหน้าต่างการใช้งานในแต่ละส่วนดังนี้



รูปที่ 1 ลักษณะโครงสร้างโปรแกรม MATLAB 6.5 พร้อมใช้งาน

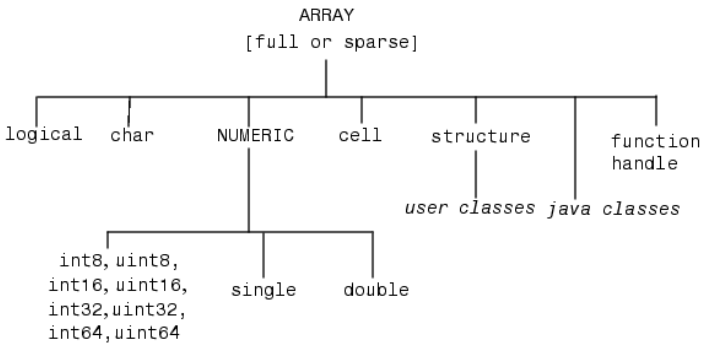
1. Menu Bar คือ เมนูบาร์เกี่ยวกับคำสั่งในการจัดการหน้าต่างของโปรแกรม
2. Workspace คือ หน้าต่างของตัวแปรที่ใช้งาน พื้นที่หน่วยความจำ และขนาดตัวแปร
3. Command History คือ หน้าต่างที่ใช้เก็บคำสั่งที่ใช้งานย้อนหลังสามารถใช้โดยการดับเบิลคลิก
4. Command Windows คือ หน้าต่างที่ใช้เรียกคำสั่งโดยการพิมพ์คำสั่งที่ต้องการประมวลผล

หลังเรียกใช้งานโปรแกรมได้แล้วในการทดลองนี้ต้องศึกษาคำสั่งพื้นฐานที่จำเป็นเบื้องต้นซึ่งสามารถ แบ่งออกเป็น 6 ส่วนใหญ่ ดังต่อไปนี้

3.1 พื้นฐานการใช้งานเบื้องต้น

3.1.1 ตัวแปรและประโยค

ตัวแปร (variable) รูปแบบตัวแปรในโปรแกรมมีหลายแบบดังรูปที่แสดงโครงสร้างของลักษณะตัวแปรที่ใช้ในโปรแกรม



รูปที่ 2 โครงสร้างรูปแบบของตัวแปรที่ประกาศในโปรแกรม

Logical ตัวแปรแบบตรรก, Char ตัวแปรแบบตัวอักษร, Numeric ตัวแปรแบบเลข

Cell ตัวแปรแบบเซลล์ , Structure ตัวแปรแบบโครงสร้าง, Java classes ตัวแปรคลาสแบบจาวา

Function Handle ตัวแปรแบบจัดการข้อมูล

ประโยค(Expression) รูปแบบของประโยคที่ใช้ในโปรแกรมมีลักษณะเป็นการกำหนดชื่อตัวแปรแล้วขั้นด้วยเครื่องหมาย “=”แล้วตามด้วยฟังก์ชันหรือค่าคงที่ดังตัวอย่างดังนี้

```
>> variable= expression
>> A=B+2
```

3.1.2 พื้นฐานคำสั่งการจัดการ workspace

Who ให้แสดงตัวแปรที่ประกาศในหน่วยความจำ

Whos ให้แสดงตัวแปรที่ประกาศพร้อมแสดงรูปแบบตัวแปรพร้อมขนาดหน่วยความจำที่ใช้ไป

clear ลบตัวแปรจากหน่วยความจำ

load นำตัวแปรจากดิสก์สู่หน่วยความจำ

save เก็บตัวแปรจากหน่วยความจำสู่ดิสก์

quit ออกจากโปรแกรม MATLAB

ตัวอย่างการเรียกใช้งานคำสั่งทีละบรรทัด

```
>> a=1;b='INSTRUMENTATION';c=pi;
>> who
Your variables are:
a  b  c
>> whos
Name    Size              Bytes  Class
 a       1x1                 8  double array
 b       1x15               30  char array
 c       1x1                 8  double array
Grand total is 17 elements using 46 bytes
>> save number a b c
>>clear a b
>>whos
 Name     Size              Bytes  Class
 c        1x1                 8  double array
Grand total is 1 element using 8 bytes
>>load number
>>who
Your variables are:
a  b  c
>>quit
```

3.1.3 เมตริกซ์และเวคเตอร์

การป้อนเมตริกซ์ใน MATLAB กระทำได้โดยการสร้างตัวแปรให้เก็บอยู่ในรูปแบบเมตริกซ์สามารถกำหนด โดยเครื่องหมายคอมม่า (,) แทนการแบ่งระหว่างคอลัมน์และเครื่องหมายเซมิโคลอน (;) แทนการแบ่งระหว่างแถว (ส่วนการเว้นวรรคมากน้อยไม่มีผลใน MATLAB) ส่วนเวคเตอร์คือ เมตริกซ์ที่มีแถวเดียวหรือคอลัมน์เดียว (เรียกว่าเวคเตอร์แถวหรือเวคเตอร์คอลัมน์)

ตัวอย่างในการป้อนเมตริกซ์ $a=\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}$

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

ตัวอย่างในการป้อนเวคเตอร์แถว $b=[1\quad 2\quad 3\quad 4\quad 5\quad 6\quad 7\quad 8\quad 9\quad 10]$

```
>>b=[1,2,3,4,5,6,7,8,9,10]; หรือ
>>b=[1 2 3 4 5 6 7 8 9 10]; หรือ
>>b=[1:10]; รูปแบบคือ เพิ่มค่าที่ละหนึ่งเสมอ หรือ
>>b=[1:1:10]; รูปแบบคือ [ค่าเริ่มต้น:เพิ่มที่ละ:ค่าสุดท้าย]
```

ตัวอย่างในการป้อนเวคเตอร์แบบคอลัมน์ $c=\begin{bmatrix}1\\ 2\\ 3\\ 4\end{bmatrix}$

```
>> c = [1; 2; 3; 4];
```

การกระทำเมตริกซ์

สามารถบวกลบคูณและหารได้ตามปกติ แต่ต้องถูกต้องตามหลักการของเมตริกซ์ ซึ่งใช้สัญลักษณ์ในการบวกคือ (+) การลบคือ(-) และการคูณ (*) การหาร(/) การยกกำลัง (^) ส่วนทรานสโพส (.‘) แต่ถ้าต้องการให้คอนจูเกตทรานสโพส (‘)

```
>>D=a'                %การทรานสโพสเมตริกซ์
>>E=a+D;H=a-D;        %การบวกลบเมตริกซ์
>>F=2*E               %การคูณเมตริกซ์
>>G=E^2               %การยกกำลังเมตริกซ์
>>E=a/d               %การหารเมตริกซ์
```

การกระทำที่เข้าถึงสมาชิกทุกตัวในเมตริกซ์

การกระทำทางคณิตศาสตร์ที่สามารถเข้าถึงสมาชิกทุกตัวในเมตริกซ์หรือเวคเตอร์โดยการใช้เครื่องหมายจุด (.) เพิ่มจากการกระทำทางคณิตศาสตร์ทั้งการคูณและการหารรวมถึงการยกกำลัง ซึ่งสามารถจะกระทำสมาชิกตรงตำแหน่งนั้นตัวอย่างเช่น

```
>> K=[1 2 3]; L=[4 5 6];
>>K.*L
   ans = [4 10 18]
```

การอ้างถึงสมาชิกภายในเมตริกซ์หรือเวคเตอร์

สามารถกระทำได้โดยการอ้างตัวแปรที่เป็นเมตริกซ์หรือเวคเตอร์โดยการชี้ไปที่ตำแหน่งต่างๆ ของสมาชิกในเมตริกซ์หรือเวคเตอร์ ซึ่งที่พิเศษคือโปรแกรม MATLAB สามารถยอมให้อ้างได้ที่ละหลายค่าในการสร้างเมตริกซ์หรือเวคเตอร์ขึ้นใหม่ โดยรูปแบบในการอ้างอิงตำแหน่งต่างๆ คือ ตัวแปร (แถว, คอลัมน์) เช่น a (1, 2) จะได้ค่าเป็น 4 เช่นดังตัวอย่างในการสร้างเมตริกซ์หรือเวคเตอร์ใหม่เช่น $A=\begin{bmatrix}1 & 4 & 7\\ 2 & 5 & 8\\ 3 & 6 & 9\end{bmatrix}$ สังเกตว่าการอ้างทั้งแถวหรือคอลัมน์ใช้โคลอน (:)

```
>> A=[1,4,7;2,5,8;3,6,9];
>> B=A([1 2],[2 3])
 B =    4    7
        5    8
>> C=A(2,:)
 C =  2    5    8
>> D=A(:,2)
 D =    4, 5 ,6
```

3.1.4 จำนวนเชิงซ้อน

ในการคำนวณเกี่ยวกับจำนวนเชิงซ้อนในโปรแกรม MATLAB สามารถกระทำได้โดยตรงโดยการอ้างตัวแปรเป็นแบบจำนวนเชิงซ้อนโดยที่โปรแกรม MATLAB เตรียมตัวแปร $i, j = \sqrt{-1}$ ซึ่งเป็นส่วนจินตภาพของจำนวนเชิงซ้อน ดังนั้นจึงควรหลีกเลี่ยงการตั้งชื่อตัวแปรเป็น

สองอักษรนี้ในโปรแกรม MATLAB การตั้งชื่อตัวแปรเป็นตัวเล็กหรือตัวใหญ่จะถือเป็นคนละตัวแปรกัน เพราะถ้าตั้งชื่อเป็น j จะทำให้ค่าผิดไปแต่ในการใช้ตัวแปร i ไม่มีปัญหาในการอ้างซ้ำ ส่วนตัวแปรพิเศษ pi ใช้แทนค่า π สำหรับการป้อนค่าแบบโพล่า ซึ่งตัวอย่างการป้อนค่าจำนวนเชิงซ้อน

```
>>a=1+j*2; b=3+j*4;
>> a*b
ans =  -5.0000 +10.0000i
>>a=2*exp(j*pi/4)
a =  1.4142 + 1.4142i
```

โปรแกรม MATLAB ยังเตรียมฟังก์ชันพื้นฐานที่เกี่ยวข้องกับจำนวนเชิงซ้อนอีกเช่น

```
>>real(a)      % หาส่วนจำนวนจริงของตัวแปร a
>>imag(a)      % หาส่วนจำนวนจินตภาพของตัวแปร a
>>abs(a)       % หาค่าขนาดของตัวแปร a
>>angle(a)     % หาค่ามุมของตัวแปร a
>>conj(a)      % หาค่าคอนจูเกตของตัวแปร a
```

3.1.5 ฟังก์ชันทางคณิตศาสตร์

โปรแกรม MATLAB มีฟังก์ชันทางคณิตศาสตร์มากมายแต่จะยกมาแสดงพอสังเขปได้ดังต่อไปนี้

ฟังก์ชันเกี่ยวกับสเกลา

ฟังก์ชันตรีโกณมิติ sin(x), cos(x),tan(x), asin(x), acos(x), atan(x), atan2(x)

ฟังก์ชันไฮเปอร์โบลิก sinh(x), cosh(x),tanh(x), asinh(x), acosh(x), atanh(x)

ฟังก์ชันเลขยกกำลัง exp(x),log(x),log10(x),

ฟังก์ชันทั่วไป abs(x),sign(x),rem(x),round(x),floor(x),ceil(x),sqrt(x)

ฟังก์ชันเกี่ยวกับเวคเตอร์

max(x),min(x),length(x),mean(x),median(x),std(x),sum(x),sort(x),roots(x)

ฟังก์ชันเกี่ยวกับการสร้างเมตริกซ์

Zeros(n,m) สร้างเมตริกซ์ที่มีสมาชิกเป็น 0 ทั้งหมดขนาด n x m

Ones(n,m) สร้างเมตริกซ์ที่มีสมาชิกเป็น 1 ทั้งหมดขนาด n x m

ฟังก์ชันเกี่ยวกับเมตริกซ์

size(x),eig(x),det(x)

ซึ่งรายละเอียดในการใช้คำสั่งที่กล่าวมาข้างต้นสามารถหารายละเอียดได้จากคำสั่ง help command

3.1.6 การจัดการเกี่ยวกับไดเรกทอรีและไฟล์

โปรแกรม MATLAB เตรียมคำสั่งเกี่ยวกับการจัดการไดเรกทอรีและไฟล์ในเครื่องคอมพิวเตอร์

cd เปลี่ยนไดเรกทอรี

copyfile คัดลอกไฟล์หรือไดเรกทอรี

movefile เคลื่อนย้ายไฟล์หรือไดเรกทอรี

delete ลบไฟล์

pwd แสดงไดเรกทอรีปัจจุบัน

dir แสดงรายละเอียดในไดเรกทอรี

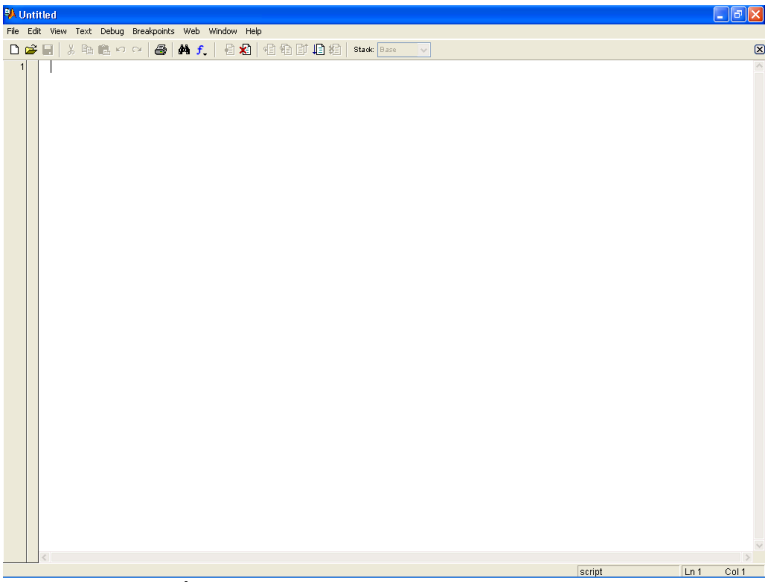
isdir ตรวจสอบว่าเป็นไดเรกทอรีหรือไม่

mkdir สร้างไดเรกทอรีใหม่

rmdir ลบไดเรกทรอรี

3.2 พื้นฐานการเขียนสคริปและฟังก์ชัน

การป้อนคำสั่งในการเรียกใช้งานที่ command line ที่ละคำสั่งสามารถที่จะรวบรวมและสั่งใช้งานหลายคำสั่งโดยการเขียนเป็นลักษณะ m-file ซึ่งภายในโปรแกรม MATLAB editor เป็นตัวสร้างและแก้ไข โดยการใช้จากเมนู ซึ่งเลือก file new m-file จากเมนู จะปรากฏ editor ดังรูปที่ 2



รูปที่ 2 แสดง Editor ของโปรแกรม MATLAB

3.2.1 การเขียนสคริป

ในการเขียนสคริปคือการเรียกใช้งานคำสั่งทีละคำสั่งซึ่งมีลักษณะเหมือนกับการเขียนโปรแกรมภาษาโดยทั่วไปแล้วเก็บให้มีนามสกุลเป็น .m จากนั้นสามารถสั่งรันได้จาก editor ของโปรแกรม MATLAB ซึ่งเป็นตัว debug ได้ด้วย หรือจะเรียกจาก Command line โดยการเรียกชื่อไฟล์เป็นคำสั่ง

```
% เครื่องหมายเปอร์เซ็นต์จะใช้ในการเขียน Comment
```

ในการสร้างสคริปมีโครงสร้างคำสั่งที่ควบคุมทิศทางของโปรแกรมคล้ายกับทางภาษาสูงโดยทั่วไปในการเขียนโปรแกรมคือ

คำสั่ง if, elseและ elseif ใช้ในการตัดสินใจทางเลือก รูปแบบในการใช้งานคือ

```
if <เงื่อนไขตัดสินใจ>
        <ประโยค>
elseif <เงื่อนไขตัดสินใจ>
        <ประโยค>
else
        <ประโยค>
end
```

คำสั่ง while ใช้ในวนรอบโดยมีการตรวจสอบเงื่อนไข รูปแบบในการใช้งานคือ

```
while <เงื่อนไขตรวจสอบ>
        <ประโยค>
end
```

คำสั่ง switch ใช้ในการตัดสินใจทางเลือกในกรณีที่มีทางเลือกมากกว่าสามรูปแบบการใช้งานคือ

```
switch ( ตัวแปร)   <scalar or string>
case เงื่อนไขที่ 1
        <ประโยค> % Executes if expression is value1
case เงื่อนไขที่ 2
        <ประโยค> % Executes if expression is value2
.
.
otherwise
        <ประโยค>% Executes if expression does not
                % does not match any case
end
```

คำสั่ง for ใช้ในวนรอบ รูปแบบในการใช้งานคือ

```
for ตัวแปร= ค่าเริ่ม:เพิ่มที่ละ:ค่าสุดท้าย
<ประโยค>
end
```

สรุปเกี่ยวกับการเขียนสคริป

คือการนำคำสั่งมาเขียนเรียงต่อกันแล้วเก็บอยู่ในนามสกุล .m ซึ่งสามารถเรียกใช้งานภายหลัง ทำให้สะดวกในการใช้งานซ้ำ แต่ข้อจำกัดในการเรียกใช้งานได้หลายครั้งแต่ไม่สามารถรับค่าและคืนค่าตัวแปรได้ ถ้าต้องการรับส่งค่าต้องเขียนในรูปแบบของฟังก์ชันแทน

3.2.2 การเขียนฟังก์ชัน

การเขียนฟังก์ชันมีลักษณะคล้ายกับการเขียนสคริปแต่สามารถรับส่งค่าไปมาได้ แต่ในการเรียกใช้งานตัวแปรที่ใช้อยู่ในฟังก์ชันนั้นจะไม่คงอยู่หลังจากการเรียกใช้ฟังก์ชันผ่านไป ซึ่งรูปแบบในการเขียนเป็นฟังก์ชันกำหนดได้ดังนี้

function ชื่อฟังก์ชัน(อินพุต1,อินพุต2)

ประโยค

....

ชื่อฟังก์ชัน=ค่าที่จะคืนกลับ

end

3.2.3 ตัวอย่างเรียกใช้งานสคริปและฟังก์ชัน

ตัวอย่างในการคำนวณเกี่ยวกับค่าเฉลี่ยเลขคณิตโดยการเขียนเป็นฟังก์ชันแล้วเรียกใช้โดยการสร้างสคริปมาใช้งานดังนี้

1.สร้างฟังก์ชันดังตัวอย่างข้างล่างแล้ว save ชื่อ average.m

2.สร้างสคริปดังตัวอย่างข้างล่างแล้ว save ชื่อ testmath.m

3.ทดสอบใช้งานโดยพิมพ์ testmath แล้วตามด้วย enter ที่ command line

ตัวอย่างฟังก์ชันที่รับค่าเวคเตอร์แล้วคืนค่าเป็นค่าเฉลี่ยเลขคณิต average.m

```
function y = average(x)
% AVERAGE Mean of vector elements.
% AVERAGE(X), where X is a vector, is the mean of vector elements.
% Nonvector input results in an error.
[m,n] = size(x);
if (~((m == 1) | (n == 1)) | (m == 1 & n == 1))
   error('Input must be a vector')
end
y = sum(x)/length(x);      % Actual computation
```

ตัวอย่างสคริปการเรียกใช้ฟังก์ชัน testmath.m

```
z = 1:99;
average(z)
```

3.3 พื้นฐานคำสั่งเกี่ยวกับกราฟ

3.3.1 Plot เป็นคำสั่งที่ใช้ในการวาดกราฟในระนาบ XY ซึ่งรูปแบบในการใช้งานมีสามแบบดังนี้

plot(ข้อมูลแกนนอน,ข้อมูลแกนตั้ง)

plot(ข้อมูลแกนตั้ง,ข้อมูลแกนนอน,รูปแบบของเส้นกราฟ)

plot(ข้อมูลแกนตั้ง1,ข้อมูลแกนนอน1,รูปแบบของเส้นกราฟ1,ข้อมูลแกนตั้ง2,ข้อมูลแกนนอน2,รูปแบบของเส้นกราฟ2,…)

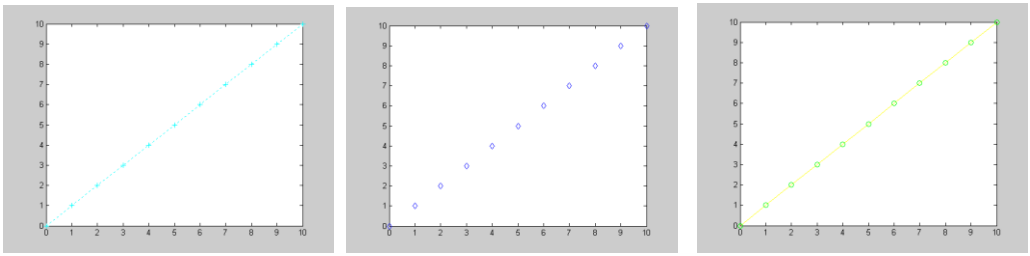
ข้อกำหนดรูปแบบของเส้นกราฟ

สี		ลักษณะมาร์ค		ลักษณะเส้น	
b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star		
y	yellow	s	square		
k	black	d	diamond		
		v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

ตัวอย่างการเรียกใช้งานในการพล็อตกราฟ

```
>>X=[0:1:10];Y=[0:1:10];
>>plot(X,Y,'c+:')
>>plot(X,Y,'bd')
>>plot(X,Y,'y-',X,Y,'go')
```

ผลการใช้คำสั่งในการพล็อตกราฟข้างต้น



รูปที่ 4 การพล็อตกราฟในรูปแบบลักษณะต่าง ๆ

3.3.2 Axis เป็นคำสั่งในการกำหนดขนาดเกี่ยวกับแกนตั้งและแกนนอนที่ปรากฏในกราฟ ซึ่งรูปแบบในการใช้งานรูปแบบดังนี้

AXIS ([XMIN XMAX YMIN YMAX])	การกำหนดแนวแกนตั้งและแกนนอน
V = AXIS	เป็นคำสั่งอ่านค่าแนวแกนตั้งและแกนนอน
AXIS AUTO	กำหนดแกนแบบอัตโนมัติ
AXIS EQUAL	กำหนดแกนทั้งสองให้เท่ากัน
AXIS OFF	ไม่ให้แสดงแกนแนวตั้งและแนวนอน
AXIS ON	ให้แสดงแนวแกนตั้งและแนวแกนนอน

3.3.3 Subplot

เป็นคำสั่งในการแบ่งรูปย่อยในหน้าต่างเดียวกันให้มี n x m รูปย่อยและโฟกัสไปที่รูปย่อยที่ p ซึ่งรูปแบบคำสั่งดังนี้ SUBPLOT(n,m,p)

3.3.4 Lable

xlable(‘ข้อความที่ต้องการให้ปรากฏบนแกน x’)

ylable(‘ข้อความที่ต้องการให้ปรากฏบนแกน y’)

title(‘ข้อความที่ต้องการให้ปรากฏชื่อกราฟ’)

3.3.5 hold on และ hold off

hold on เป็นคำสั่งในการโฟกัสกราฟรูปที่ต้องการไว้เมื่อมีคำสั่งเกี่ยวกับกราฟจะกระทำที่กราฟรูปเดิมจนกระทั้งใช้คำสั่ง hold off เมื่อกระทำกับกราฟจะกระทำกับกราฟรูปใหม่