

DE498/598 การใช้คอมพิวเตอร์ทั่วไปในการแก้
ปัญหาทางวิศวกรรม

Lecture 4: การใช้ Matlab เบื้องต้น 3

ดร. กฤษฎา ไชยสาร

Outline

- การเขียนฟังก์ชัน

การเขียน Matlab ฟังก์ชัน

- แมทแลปฟังก์ชันผู้ใช้สามารถเขียนได้เองซึ่งเรียกใช้งานได้ใน Command Window
- สำหรับฟังก์ชันในแมทแลป การส่งผ่านค่าสามารถทำได้โดยวิธีเดียวคือการ pass by value ไม่สามารถ pass by reference ได้ หรือสรุปอย่างง่ายคือการส่งผ่านค่าในแมทแลปคือ input เป็นตัวเลขและได้ output เป็นตัวเลขเช่นเดียวกัน
- ฟังก์ชันในแมทแลปมีนามสกุลคือ .m
- การเขียนฟังก์ชันในแมทแลปคล้ายกับการเขียนสคริปต์ แตกต่างกันในบรรทัดแรกต้องขึ้นต้นด้วย
 - `function [output1, output2, ...] = name(input1, input2, ...)`
- ฟังก์ชันในแมทแลปต้องมีชื่อ name ชื่อเดียวกันกับไฟล์ เช่น ฟังก์ชันชื่อ sind ไฟล์นี้ต้องมีชื่อว่า sind.m

ตัวอย่างที่ 1 ฟังก์ชัน sine โดยใช้ degree

```
function s = sind(x)
%SIND(X)   Calculates sine(x) in degrees
s = sin(x*pi/180);
```

- บรรทัดที่ 1 บอกชื่อฟังก์ชัน input คือ x และ output คือ s
- บรรทัดที่ 2 คือคำอธิบายซึ่งเป็น comment แมทแลปไม่ใช้ในการ execute สำหรับคำอธิบายใต้ฟังก์ชันนี้แมทแลปจะใช้เป็นคำอธิบายเวลาผู้ใช้ใช้ help ในแมทแลป เช่น help sind
- บรรทัดที่ 3 คือบรรทัดที่ใช้ในการคำนวณต่างโดยจะเห็นได้ว่าฟังก์ชันนี้ใช้ input ที่ชื่อว่า x ที่เป็นองศาแล้วเปลี่ยนองศาให้เป็น radian หลังจากนั้นก็เรียกฟังก์ชัน sin ธรรมดาเพื่อหาค่า sin และส่งต่อเป็น output ให้กับตัวแปร s

การใช้ฟังก์ชัน

- สามารถสร้างฟังก์ชันได้จากการเปิดจาก editor
- ลองพิมพ์ `>>help sind` ใน command window เพื่อที่จะเห็นคำอธิบายของฟังก์ชัน
- สามารถใช้ฟังก์ชันได้ใน command window ได้เลย เช่น

```
>>sind(0)
>>sind(45)
>>t=sind(45)
>>m=sind([30 60 90])
>>sind
```
- คำสั่งข้างบนแต่ละคำสั่งตามกันอย่างไร

ตัวอย่างที่ 2 Unit step

$$y = \begin{cases} 0 & \text{if } t < t_0 \\ 1 & \text{otherwise} \end{cases}$$

ตัวอย่างที่ 2 Unit step

```

function y = ustep(t, t0)
%USTEP(t, t0) unit step at t0
% A unit step is defined as
%     0 for t < t0
%     1 for t >= t0
[m,n] = size(t);
% Check that this is a vector, not a matrix i.e. (1 x n) or (m x 1)
if m ~= 1 & n ~= 1
    error('T must be a vector');
end
y = zeros(m, n); %Initialise output array
for k = 1:length(t)
    if t(k) >= t0
        y(k) = 1; %Otherwise, leave it at zero, which is correct
    end
end
end

```

ตัวอย่างที่ 2 Unit step

- บรรทัดที่ 1 บอกชื่อของฟังก์ชันโดยมี input คือ t และ t0 และมี output คือ y
- บรรทัดที่ 2 – 5 เป็นคำอธิบายฟังก์ชัน
- บรรทัดที่ 6 เป็นการใช้ฟังก์ชัน size เพื่อเช็คขนาดของตัวแปร t
- บรรทัดที่ 7 ถึง 10 คือการเช็คค่า t ต้องเป็นเวกเตอร์
- บรรทัดที่ 11 คือการสร้างเวกเตอร์เปล่าๆขึ้นมาหนึ่งตัวแล้ว initialise ให้เป็น 0 ทั้งหมด ซึ่งเวกเตอร์นี้คือเวกเตอร์ที่เป็น output
- บรรทัดที่ 12 ใช้ for ในการหาค่าแต่ละค่าในเวกเตอร์ input t
- บรรทัดที่ 13 – 15 คือการเขียนเงื่อนไขตามสมการที่กล่าวไว้ข้างต้นสำหรับการหา unit step ฟังก์ชัน

ตัวอย่างที่ 2 Unit step

- หลังจากเขียนฟังก์ชันเสร็จแล้วให้ save ไฟล์ให้ชื่อว่า ustep.m
- หลังจากนั้นให้ลองวาดกราฟแสดงผลของ unit step


```
>>x = -1:0.1:4;
>>plot(x,1)
```
- ให้สร้างกราฟ pulse จากคำสั่งข้างล่าง


```
>>v = ustep(x,0) - ustep(x,1)
>>plot(x,v)
>>axis([-1 4 -1 2])
```

Assignment

ค่าอินทิเกรตของ unit step คือฟังก์ชัน unit ramp ซึ่งมีนิยามคือ

$$y = \begin{cases} 0 & \text{if } t < t_0 \\ t - t_0 & \text{otherwise} \end{cases}$$

ดังที่โชว์ในภาพ (a) ให้เขียนฟังก์ชัน uramp ตามค่านิยามข้างต้นและให้เขียนสคริปต์ เพื่อให้ได้ภาพตามที่โชว์ในภาพ (b) และให้ส่งคำตอบเว็บที่หน้าเพจของวิชาโดยใช้ชื่อไฟล์ว่า week4_รหัสนักศึกษา_ชื่อนักศึกษา

