

MATLAB簡介

- 《 MATLAB 程式設計與應用》
<http://www.cs.nthu.edu.tw/~jang/mlbook/>
- The MathWorks - Product Listing - Products by Category
http://www.mathworks.com/products/product_listing/index.html
- INTLAB - INTerval LABoratory
<http://www.ti3.tu-harburg.de/~rump/intlab/>

容易學習的理由

- 不需要 宣告變數
- 屬高階語言：程式語法貼近口語邏輯
- 直譯式：不需要編譯程式

外觀介紹

- Command Window
- Workspace
- Current Directory
- help (善用查詢指令)

基本運算

運算符號：

加(+), 減(-), 乘(*), 除(/), 右除(\), 指數(^)

```
>> (5*2+3.5)/5
```

```
ans =
```

```
2.7000
```

```
>> (5*2+3.5)/5;
```

- 有分號的話則不會顯示計算結果
- `format`
- 大小寫有區別

內建參數及複數

- **pi = 3.141592.....**
- **Inf = infinity (inf)**
- **NaN = not a number (0/0, inf-inf) (nan)**
- **$z = 3 + 4j$**
- **i, j** 為內建虛數符號
- 變數命名: **a-z**為開頭, 之後可接 **_ 1 2 3**數字, 長度可以蠻長的, **63**個字母。

向量

- $s = [1 \ 3 \ 5 \ 2 \ 4]$

`% s = [1,3,5,2,4]` 與上列結果相同

- $t = 2*s+1$

- $t(3) = 2$

- $t(4) = []$

- $t(1:3) \quad \% \ 1:1:3$ 同義於 $1:3$

- $C = [1:2:10]$

- C'

矩陣

- `A = [1 2 3 4; 5 6 7 8; 9 10 11 12];`
- `A(2,3) = 5`
- `A(3)`
- `B = A(2,1:3)`
- `A = [A B']`
- `A(:, 2) = []`
- `A = [A; 4 3 2 1]`
- `A([1 4], :) = []`

size, length

矩陣

- **$n = 5; m = 10;$**
- **$A = \text{zeros}(n,m);$**
- **$A = \text{ones}(n,m);$**
- **$A = \text{eye}(n);$**

矩陣運算

A, B : 已設定好的矩陣

- $A+B$: 矩陣相加
- $A - B$: 矩陣相減
- $A*B$: 矩陣相乘
- A/B : 矩陣左除
- $A\backslash B$: 矩陣右除
- $A.*B$: 矩陣點乘
- $A./B$: 矩陣點除

矩陣運算

- $A = [1:3 ; 4:6 ; 7:9];$
- $B = (5*A+3.5)/5$
- $C = \text{eye}(3)$
- 比較 $A*C$ 與 $A.*C$
- $b = [1:2:5]'$
- % 計算 $Ax=b$, 求 $x = ?$

數學函數

- $y = \text{abs}(x)$
- $y = \text{sin}(x)$
- $y = \text{exp}(x)$
- $y = \text{log}(x)$
- $y = \text{sqrt}(x)$

數學函數(複數)

- $y = \text{angle}(z)$
- $y = \text{real}(z)$
- $y = \text{imag}(z)$
- $y = \text{conj}(z)$
- $y = \text{exp}(j*\text{pi}/6)$

數學函數(向量)

● `y = min(x)`

● `y = max(x)`

● `y = mean(x)`

● `y = sum(x)`

● `y = sort(x)`

● `y = prod(x)`

`% ex: n! = prod([1:n])`

M-file

- 底稿
 1. 變數是基本工作空間的全域變數
 2. 容易進行變數檢視及除錯
- 函式
 1. 通常包含輸入與輸出
 2. 變數預設為局部

百分比符號, %, 後面是註解

邏輯關係運算

- == 等於
- ~= 不等於
- < 小於
- > 大於
- <= 小於或等於
- >= 大於或等於
- & 且 (logical and)
- | 或 (logical or)
- ~ 非 (logical not)
- 1 真 (true)
- 0 假 (false)

流程控制 Program Control Statements

- Conditional Control

 - if

 - switch

- Loop Control

 - for

 - while

條件控制 – if

```
if 條件式
    運算式1
else
    運算式2
end
```

```
Ex: if pi*pi > 2^pi
      disp( 'pi^2 is bigger. ')
else
      disp( '2^\pi is bigger.')
end
```

條件控制 – if

```
if 條件式1
    運算式1
else
    if 條件式2
        運算式2
    else
        運算式3
end
```

```
if 條件式1
    運算式1
elseif 條件式2
    運算式2
else
    運算式3
end
```

條件控制 – switch

```
switch expression (scalar or string)
  case value1
    statements           % Executes if expression is value1
  case value2
    statements           % Executes if expression is value2
  .
  .
  .
  otherwise
    statements           % Executes if expression does not
                        % match any case
end
```

條件控制 – switch

```
switch input_num
    case -1
        disp('negative one');
    case 0
        disp('zero');
    case 1
        disp('positive one');
    otherwise
        disp('other value');
end
```

條件控制 – switch

```
switch var
  case 1
    disp('1')
  case {2,3,4}
    disp('2 or 3 or 4')
  case 5
    disp('5')
  otherwise
    disp('something else')
end
```

迴圈控制 – for loop

```
for n = 2:6  
    x(n) = 2 * x(n - 1);  
end
```

- 間隔不寫預設為 1
for 變數 = 初值 : 終值
 運算式
end

迴圈控制 – while

```
while expression  
    statements  
end
```

```
n = 1;  
while prod(1:n) < 1e100  
    n = n + 1;  
end
```

流程控制 Program Control Statements

- Loop Control
 - continue
 - break
- Program Termination
 - return

迴圈控制 – 跳出迴圈

- **break**

強制中斷並跳出迴圈。

- **continue**

回到迴圈的下一步繼續做。

函式

函式名稱(需與檔案名稱相同)

輸出

輸入

`function y = sum1n(x)`

`y = 0;`

`for ii = 1 : x`

`y = y + ii;`

`end`

程式本體

二維繪圖

- 給定 x 座標(向量)
- 給定 y 座標(向量)
- `plot(x , y , ' 顏色 符號 or 線 ')`

二維繪圖

figure(1)

```
x = linspace(0, 2*pi); % 預設為100點
```

```
y = sin(x);
```

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

figure(2)

```
x = linspace(0,2*pi,10);
```

```
y = sin(x);
```

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

圖形輸出

```
plot(x,y,'-o')  
fn = ['outmap.jpg'];  
print( '-djpeg', fn );  
  
fn = ['outmap.ps'];  
print( '-depsc', fn );  
  
fn = ['outmap.fig'];  
saveas((gcf, fn);
```

延伸學習

- subplot
- plotyy
- semilogy, loglog
- Legend
- gplot
- orient

- Getting Started
 - Examples
 - Development Environment
 - Mathematics
 - Programming and Data Types
 - M-File Programming
 - Character Arrays (Strings)
 - Multidimensional Arrays
 - Structures and Cell Arrays
 - Function Handles
 - MATLAB Classes and Objects
 - Maximizing MATLAB Performance
 - MATLAB Programming Tips
 - Examples
 - Graphics
 - Overview of MATLAB Graphics
 - Basic Plotting**
 - Basic Plotting Commands
 - Line Plots of Matrix Data
 - Plotting Imaginary and Complex Data
 - Plotting with Two Y-Axes
 - Setting Axis Parameters
 - Figure Windows
 - Formatting Graphs
 - Creating Specialized Plots
 - Displaying Bit-Mapped Images
 - Printing and Exporting
 - Handle Graphics Objects
 - Figure Properties

Basic Plotting

[Basic Plotting Commands](#)

Basic commands for creating line plots, specifying line styles, colors, and markers, and setting defaults.

[Line Plots of Matrix Data](#)

Line plots of the rows or column of matrices.

[Plotting Imaginary and Complex Data](#)

How the plot command handles complex data as a special case.

[Plotting with Two Y-Axes](#)

Creating line plots that have left and right y-axes.

[Setting Axis Parameters](#)

Specifying axis ticks location, tick labels, and axes aspect ratio.

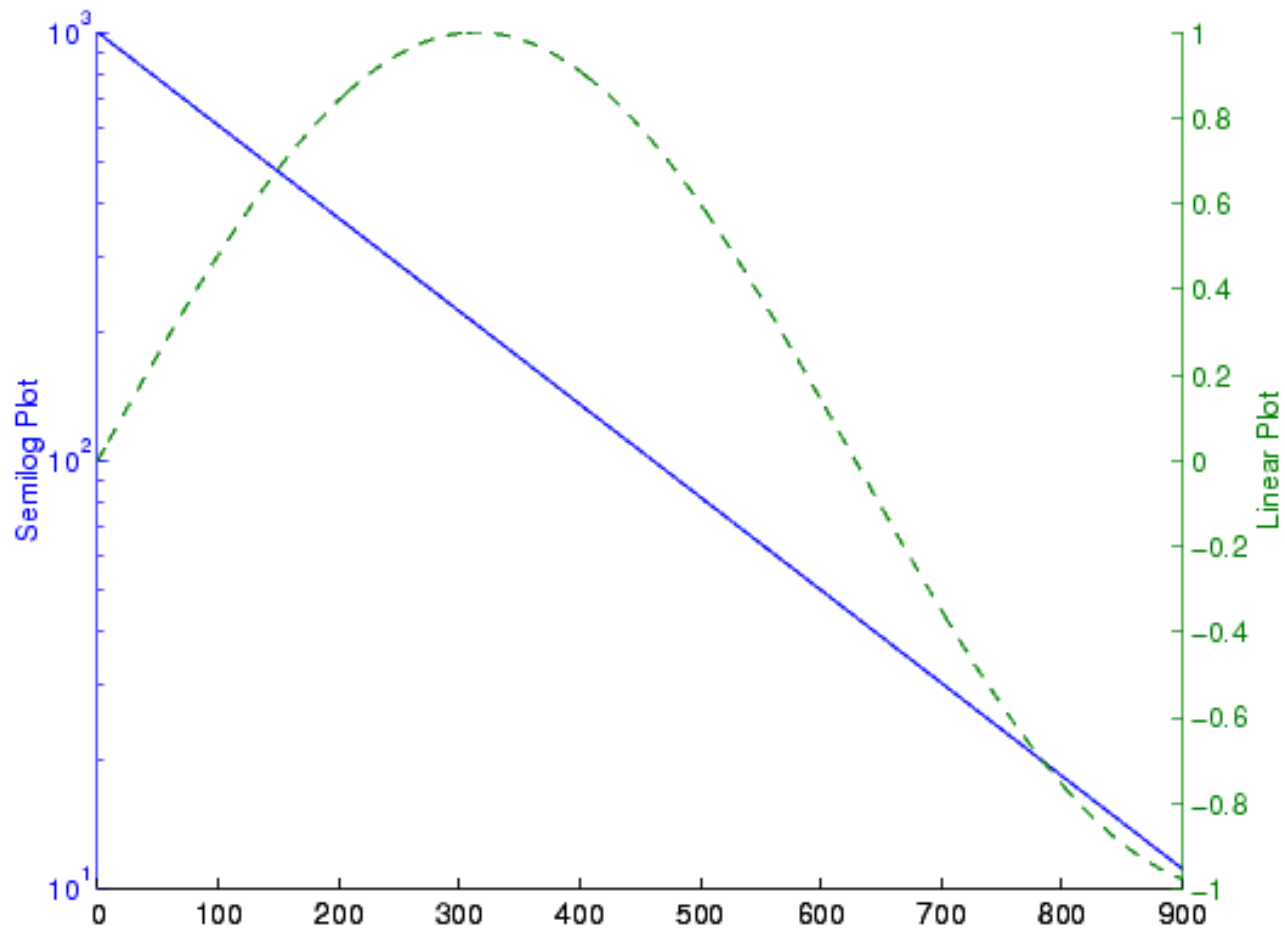
[Figure Windows](#)

Displaying multiple plots per figure, targeting a specific axes, figure color schemes.

Basic Plotting Commands

MATLAB provides a variety of functions for displaying vector data as line plots, as well as functions for annotating and printing these graphs. The following table summarizes the functions that produce basic line plots. These functions differ in the way they scale the plot's axes. Each accepts input in the form of vectors or matrices and automatically scales the axes to accommodate the data.

Function	Description
plot	Graph 2-D data with linear scales for both axes
plot3	Graph 3-D data with linear scales for both axes
loglog	Graph with logarithmic scales for both axes
semilogx	Graph with a logarithmic scale for the x -axis and a linear scale for the y -axis
semilogy	Graph with a logarithmic scale for the y -axis and a linear scale for the x -axis
plotyy	Graph with y -tick labels on the left and right side



- ⊕ Figure Windows
- ⊕ Formatting Graphs
- ⊕ Creating Specialized Plots
- ⊕ Displaying Bit-Mapped Images
- ⊖ Printing and Exporting
 - ⊕ Overview of Printing and Exporting
 - ⊕ How to Print or Export
 - ⊕ Examples of Basic Operations
 - ⊕ Changing a Figure's Settings
 - ⊕ **Choosing a Graphics Format**
 - ⊕ Choosing a Printer Driver
 - ⊕ Troubleshooting
- ⊕ Handle Graphics Objects
- ⊕ Figure Properties
- ⊕ Axes Properties
- Examples
 - 3-D Visualization
 - Creating Graphical User Interfaces
 - Functions - By Category
 - Functions - Alphabetical List

Frequently Used Graphics Formats

Here are some of the more frequently used graphics formats. For a complete list, see the [Graphics Format table](#) on the `print` reference page. For a more complete description of these formats, see [Description of Selected Graphics Formats](#).

Format	Description	Command Line -device Parameter
EPS color, and black and white	Export line plots or simple graphs to a file. Note. An EPS file does not display within some applications unless you add a TIFF preview image to it. See the example Exporting in EPS Format with a TIFF Preview .	-deps (black and white) -depsc (color) -depsc -tiff (TIFF preview)
JPEG 24-bit	Export plots with surface lighting or transparency to a file. This format can be displayed by most Web browsers.	-djpeg -djpeg <i>number</i> , where <i>number</i> is the compression.