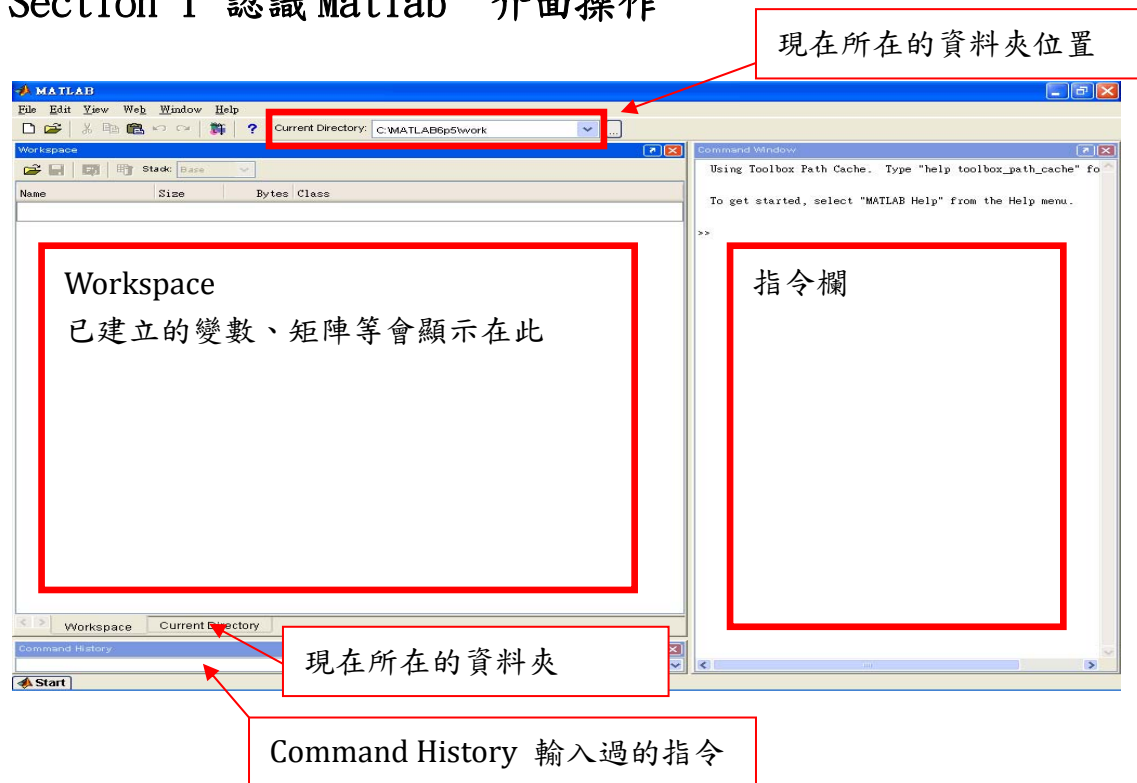


Matlab 簡易教學

Section 1 認識 Matlab 介面操作



Section 2 簡單的 Matlab 計算 宣告變數/型態 運算元

2.1 如何用 Matlab 計算?

• 1+1	• 3^5	• sin(pi/2)
• exp(0.5)	• sqrt(1+2*i)	•

2.2 基本的數學函數(Matlab 內建函數)

函數	Matlab 語法
e^x	exp(x)
\sqrt{x}	sqrt(x)
$\ln x$	log(x)
cos x sin x	cos(x) sin(x)
tan x	tan(x)
$\cos^{-1} x$ $\sin^{-1} x$	acos(x) asin(x)

2.3 變數與變數型態

number = 1.32; 宣告變數 number 並指定為數值

`messenge='hello';` 宣告變數 `messenge` 並指定為字串(注意有單引號)

1. 變數名稱只能用英文、數字和底線，不能有空格，要英文開頭
2. 變數名稱的大小寫不一樣，會視為不同變數，如 `XAve` 和 `xave` 不一樣
3. 在指令最後加上分號(`;`)，按 Enter 後不會出現計算結果
4. Matlab 有一些預設變數，如 `pi`(圓週率)，`i`(虛數)

2.4 陣列(Array)和矩陣(Matrix)

`AAA = [1, 2, 4, 8];` 宣告一 1×4 的陣列並指定數值
`BBB = [0.1, 5, 3; 6, 7, 8];` 宣告一 2×3 的矩陣並指定數值(注意中間的分號)

輸入 `AAA(3)` 出現 `ans = 4`
輸入 `BBB(2,3)` 出現 `ans = 8`

矩陣之運算：

`A.*B` 代表 `A`、`B` 之對應元素相乘
`A./B`
`A.^2` 代表 `A` 裡面的所有元素各自平方

2.5 建立函數

方法一：

`fun = @(x,y) x^2+sin(y);` 建立一個函數 `fun(x,y) = x2+sin y`
輸入 `fun(2,2)` 出現 `ans = 4.9093`

方法二：

`syms x y;` 創造兩個未知數 `x` 和 `y`
`fun = x^2+sin(y);` 建立一個函數 `fun(x,y) = x2+sin y`
輸入 `subs(fun,{x,y},{2,2})` 出現 `ans = 4.9093`

2.6 解一元方程

解出 `f(x)=0`

方法一：

`fun=@(x) x^2-1;`
`fzero(fun,0.5)` 程式會在 `x=0.5` 附近找根

方法二：

`f = x^2-1;`
`solve(f)` 程式會解出所有的根

2.7 讀入檔案

load parameter.txt; (在 Matlab 目前啟的位置)；或

load 'C:\Documents and Settings\User\My Documents\MATLAB\text2.txt'

(檔案所在位置)

在 Workspace 裡會出現一矩陣稱 parameter

注意：若要載入 Excel 檔，必需把 xls 檔存成 csv 檔。

Section 3 繪圖

3.1 平面繪圖

用描點

```
x = [0 1 2 3 4];
```

```
y = [0 1 4 9 16];
```

```
plot(x,y)
```

畫出函數

➤ 方法 1

```
x2=inline('x^2');
```

```
fplot(x2,[0 1])
```

➤ 方法 2

```
ezplot('x^2',[0,1]);
```

 注意：函數要用單引號包住，[0 1]是 x 的範圍

3.2 立體繪圖

畫出函數

➤ 方法 1

```
[X,Y] = meshgrid(-2:0.2:2, -2:0.2:2);
```

```
Z = X.* exp(-X.^2 - Y.^2);
```

```
surf(X,Y,Z)
```

➤ 方法 2

```
ezsurf('x*exp(-x^2 - y^2)',[-1 1 -2 2])
```

 注意：函數要用單引號包住

Section 4 邏輯運算

		例 1	例 2
if	if <u>condition</u> <u>statement</u> ; end	if x > 2 sum = sum + 1; end	if x < 2 && y > 1 z = x + y; end

	if <u>condition1</u> <u>statement1;</u> elseif <u>condition2</u> <u>statement2;</u> else <u>statement3;</u> end	if $x > 0$ output = 'positive'; elseif $x < 0$ output = 'negative'; else output = 'zero'; end	
for	for <u>counter = a:b</u> <u>statement;</u> end	for i = 2:10 A(i) = A(i-1) + 1; sum = i; end	for k = 1:-0.1:0 D = 2*k; end
while	while condition statement; end	while k < 10 B(k) = k; k = k + 1; end	

🔗 邏輯相關運算

|| 或

&& 且

~= 不等於

>= <= 大於等於 小於等於

Section 5 m-file

一般 m-file：把指令寫在 m-file 裡，若把檔名取為 AAA.m，則在 Matlab 指令欄裡輸入 AAA，則會執行 AAA.m 的指令。

function 型的 m-file：

例子

<pre>function result = MyDivide(x,y) if y~=0 % We avoid y=0 result = x/y; end</pre>	<p>在第一行寫 function result 是要輸出的結果 MyDivide 是 function 名稱，m-file 檔名一定要存成 MyDivide.m (x,y)是需要輸入的參數</p> <p>在第二行開始可輸入指令，求出 result %後面為註解，只供 programmer 參考，不會當作程式的一部分</p>
--	---

輸入 MyDivide(1,2)

輸出 ans = 0.5

Section 6 常用財務函數

🚦 Black-Scholes Option pricing model

[Call, Put] = **blsprice**(Price, Strike, Rate, Time, Volatility, Yield)

若等號前方不輸入 [Call, Put]，則只回傳 Call price。

[CallDelta, PutDelta] = **blsdelta**(Price, Strike, Rate, Time, Volatility, Yield)

Gamma = **blsgamma**(Price, Strike, Rate, Time, Volatility, Yield)

Vega = **blsvega**(Price, Strike, Rate, Time, Volatility, Yield)

[CallRho, PutRho] = **blsrho**(Price, Strike, Rate, Time, Volatility, Yield)

🚦 Binomial Option pricing model

[AssetPrice, OptionValue] = **binprice**(Price, Strike, Rate, Time, Increment, Volatility, Flag, DividendRate, Dividend, ExDiv)

🚦 一般函數

平均數 mean(X) 其中 X 是一個矩陣

加總 sum(X)

標準差 std(X)

下高斯 floor(y)

上高斯 ceil(y)

最接近之整數 round(y)

絕對值 abs(y)

PDF pdf('Normal',x,mu,v) 回傳 Normal(mu,v)的 pdf 在 x 的值

CDF cdf('Normal',x,mu,v)

隨機量 randn 產生標準常態分配的一個隨機量

Section 7 例子

Pricing American Put using BOPM：

Amput.m

```
function opt1 = AmPut(s,x,r,q,sig,t,n)
% s:present stock price
% x:strike price
% r:riskfree rate
% q:dividend yield
% sig:standard derivation of return
```

```

% t: time to maturity
% n: number of steps

dt = t/n;
u = exp(sig*sqrt(dt));
d = exp(-sig*sqrt(dt));
a = exp((r-q)*dt);
p = (a-d)/(u-d);

for j = 0:n;
    pr(j+1,2) = s*(d^(n-j))*(u^j);
    opt(j+1,2) = max(x-pr(j+1,2),0);
end;

for i = n-1:-1:0;
    for j = 0:i;
        pr(j+1,1) = s*(d^(i-j))*(u^j);
        opt(j+1,1) =
max( x-pr(j+1,1),(p*opt(j+2,2)+(1-p)*opt(j+1,2))*exp(-r*dt));
    end
    opt(:,2) = opt(:,1);
    pr(:,2) = pr(:,1);
end
opt1 = opt(1,1);

```