Chapter 7
使用者定義函式的進階功能

Hung-Yuan Fan (范洪源)

Department of Mathematics,
National Taiwan Normal University, Taiwan

Spring 2017

Outline

- 7.1 含函式的函式
- 7.2 局部函式、專用函式與巢狀函式
- 7.3 函式握把
- 7.4 匿名函式
- 7.5 遞迴函式

Section 7.1
含函式的函式

- **含函式的函式（function function）**是一種輸入引數包含其他函式名稱的函式。
  - 這些傳入的函式會在執行過程中被呼叫來使用。
  - `fzero('cos',[0 pi])`: 找出在 0 到 π 之間，函數 $f(x) = \cos(x)$ 的零根。
- 含函式的函式運作的關鍵: `eval` 與 `feval` 函式。
  - `eval` 函式會針對一個字元字串求值，如同這個字串的指令被直接鍵入指令視窗一樣。

```
x = eval('sin(pi/4)')  % eval(' 字元字串')
```

  - `feval` 函式則會針對一個具名的函式在特定輸入值上求值。

```
x = feval('sin',pi/4)  % feval(' 函式名', 數值)
```

- 常用的 MATLAB 含函式的函式:

函數名稱	敘述
fminbnd	對只含一個變數的函式最小化
fzero	找出只含一個變數函式之零點
quad	以數值方式對一個函式求其積分
ezplot	容易使用的函數繪圖器
fplot	根據函式名稱來畫函式圖形

Section 7.2
局部函式、專用函式與巢狀函式

函式的作用域

- **函式的作用域 (scope)** 是函式在 MATLAB 內可以被呼叫的範圍。
- 一般而言，MATLAB 函式的作用域是現行工作資料夾 (current working directory) 或目錄。
- 如果函式位於 MATLAB 搜尋路徑的目錄裡，則函式的作用域可延伸到所有在程式中的 MATLAB 函式。
- 作用域受限的函式類型:
  1. 局部函式 (或稱子函式)
  2. 專用函式 (或稱私有函式)
  3. 巢狀函式

局部函式

- 我們可以在單一函式檔案裡放置一個以上的函式。
- 最上層的函式是**主要函式（primary function）**，而在它下層的其他函式便稱為**局部函式（local functions）**或**子函式（subfunctions）**。
- 主要函式的命名必須與存檔名稱相同。
- 子函式只能被同一檔案內的其他函式所呼叫。
- 局部函式的作用域僅受限於所在檔案內，檔案外其他函式都不能呼叫這些子函式，但可呼叫主要函式。
- 子函式通常被當成主要函式的計算「工具」使用。

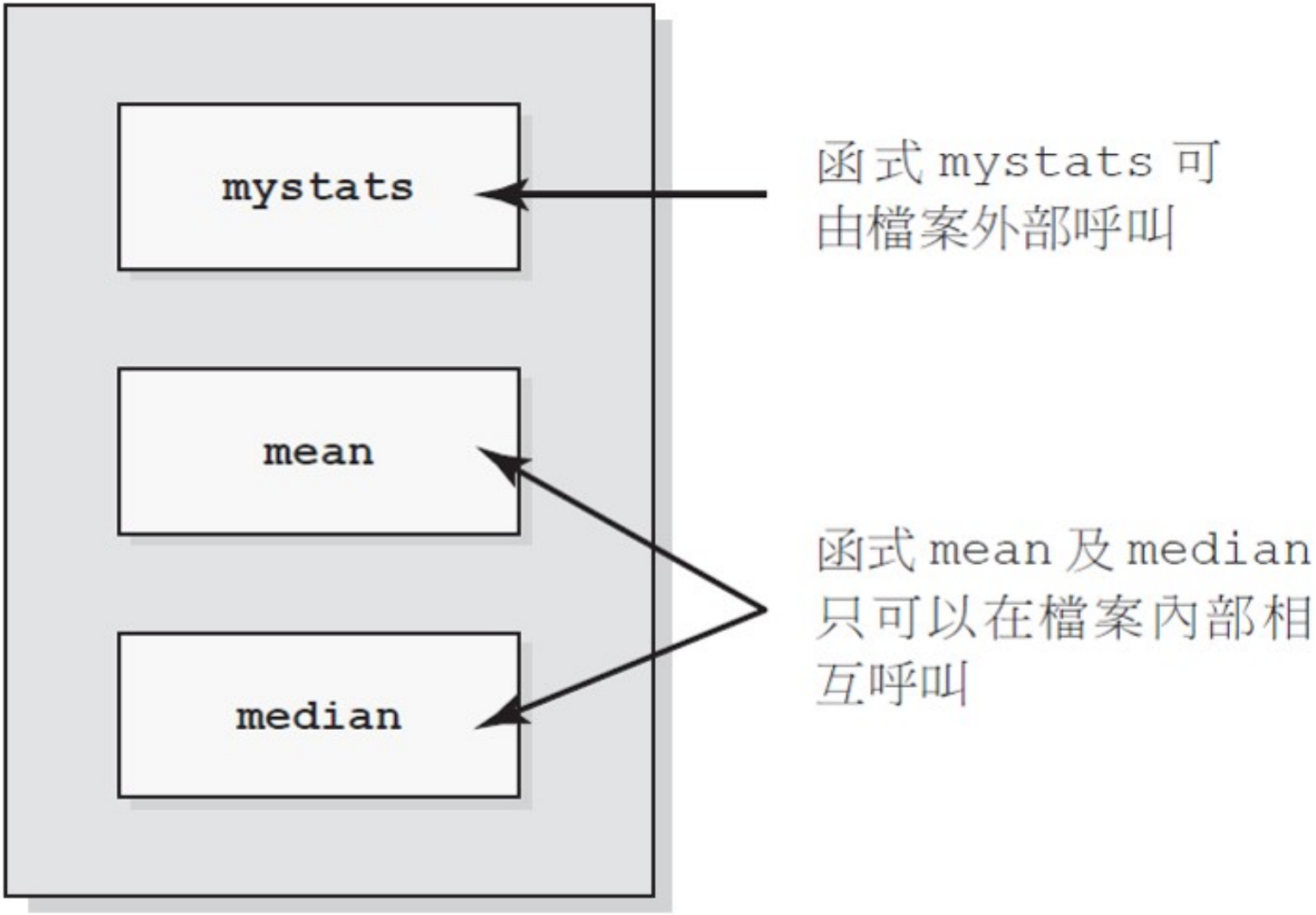
主要函式與子函式（檔名: mystats.m）

```
function [avg,med] = mystats(u)
% Primary function.
n = length(u);
avg = mean(u,n);
med = median(u,n);

function a = mean(v,n)
% Subfunction to calculate average.
a = sum(v) / n;

function m = median(v,n)
% Subfunction to calculate median.
w = sort(v);
if rem(n,2) == 1
  m = w((n+1)/2);
else
  m = (w(n/2) + w(n/2+1)) / 2;
end
```

函式間呼叫的示意圖 (承上頁)



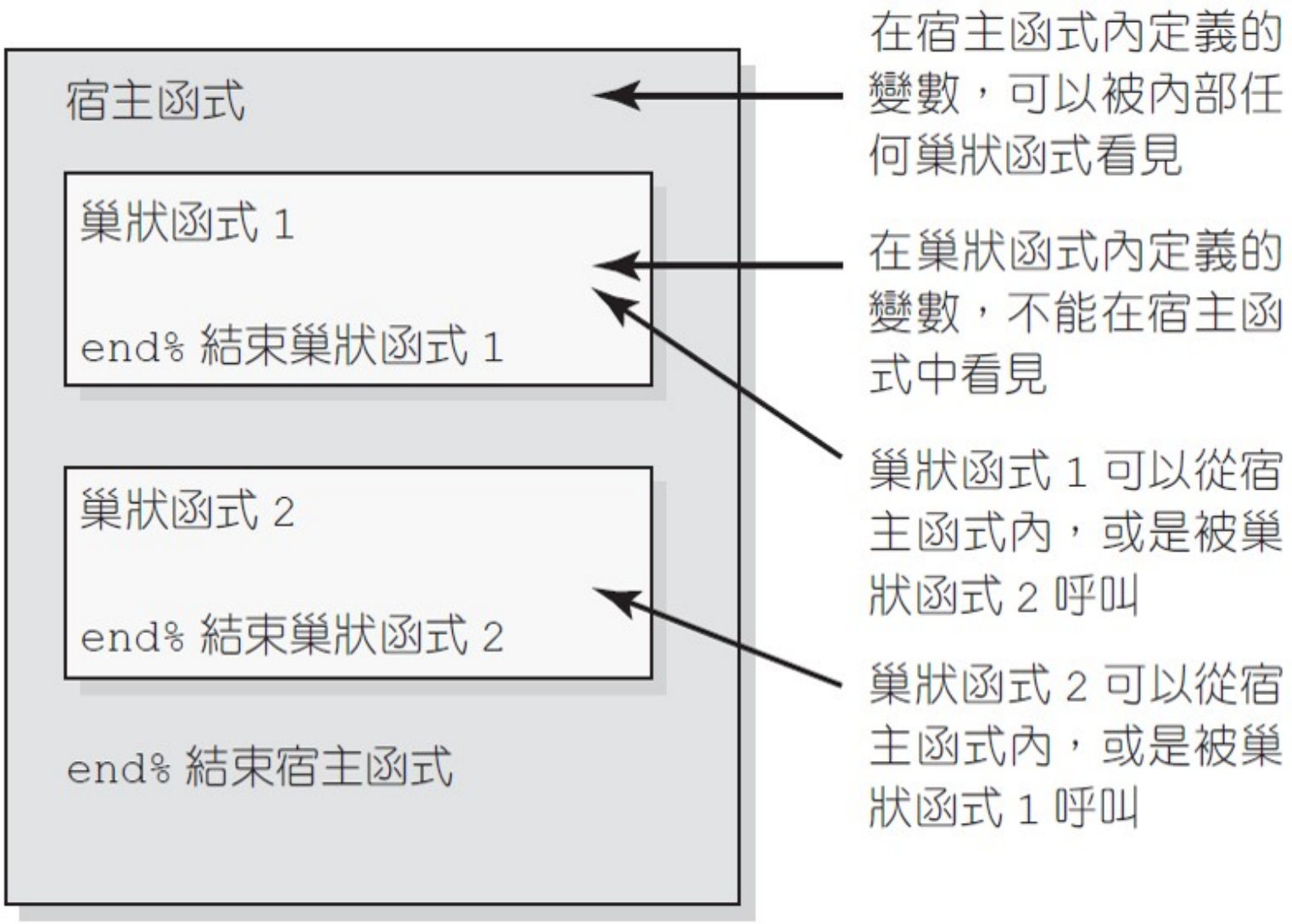
專用函式

- **專用函式或稱私有函式 (private functions)** 是儲存在於特定子目錄下的函式。
- 此子目錄 (或稱私有化目錄) 擁有特別名稱 `private`。
- 它們只能被 `private` 目錄，或是上層目錄 (或是資料夾) 的其他函式所看見。
- 通常被呼叫的 (子) 函式均放置於 `private` 子目錄內，而主程序檔或是主要函式存放於其上層目錄中。

巢狀函式

- **巢狀函式（nested function）**是一種完全存在於另一個函式〔稱為宿主函式（host function）〕本體內的函式。
- 它們只能被所嵌入的宿主函式，或是在同個宿主函式內的其他巢狀函式所看見。
- 一個巢狀函式可以存取在其函式內部的任何變數，以及在**宿主函式裡的任何變數**。
  - 當巢狀函式內的變數名稱，與宿主函式的變數名稱相同時，宿主函式裡的變數是不能被存取的。
  - 如果一個檔案擁有一個或多個巢狀函式時，每一個函式都**必須使用**一個 `end` 宣告來結束函式。
  - 這是**唯一**需要在函式結束時，加上宣告式 `end` 的時機。

函式間呼叫的示意圖 (承上頁)



巢狀函式的範例: test_nested_1.m

```
function res = test_nested_1

a = 1; b = 2; x = 0; y = 9;
fprintf('Before call to fun1: \n');
fprintf('a, b, x, y = %2d %2d %2d %2d \n',a,b,x,y);
x = fun1(x);
fprintf('\n After call to fun1: \n');
fprintf('a, b, x, y = %2d %2d %2d %2d \n',a,b,x,y);

  % Declare a nested function
  function res =fun1(y)
  fprintf('\n At start of call to fun1: \n');
  fprintf('a, b, x, y = %2d %2d %2d %2d \n',a b,x,y);
  y = y + 5; a = a + 1; res = y;
  fprintf('\n At end of call to fun1: \n');
  fprintf('a, b, x, y = %2d %2d %2d %2d \n',a,b,x,y);
  end % function fun1

end % function test_nested_1
```

程式測試結果 (承上例)

```
>> test_nested_1
Before call to fun1:
a, b, x, y = 1 2 0 9

At start of call to fun1:
a, b, x, y = 1 2 0 0

At end of call to fun1:
a, b, x, y = 2 2 0 5

After call to fun1:
a, b, x, y = 2 2 5 9
```

MATLAB 函式的計算順序

1. 檢查是否有這個名稱的巢狀函式。
2. 檢查是否有這個名稱的局部函式 (或是子函式)。
3. 檢查是否有這個名稱的專用函式。
4. 檢查在現行的工作資料夾下是否有這個名稱的函式。
5. 在 MATLAB 搜尋路徑上檢查是否有這個名稱的函式。

Section 7.3
函式握把

MATLAB 的資料型態

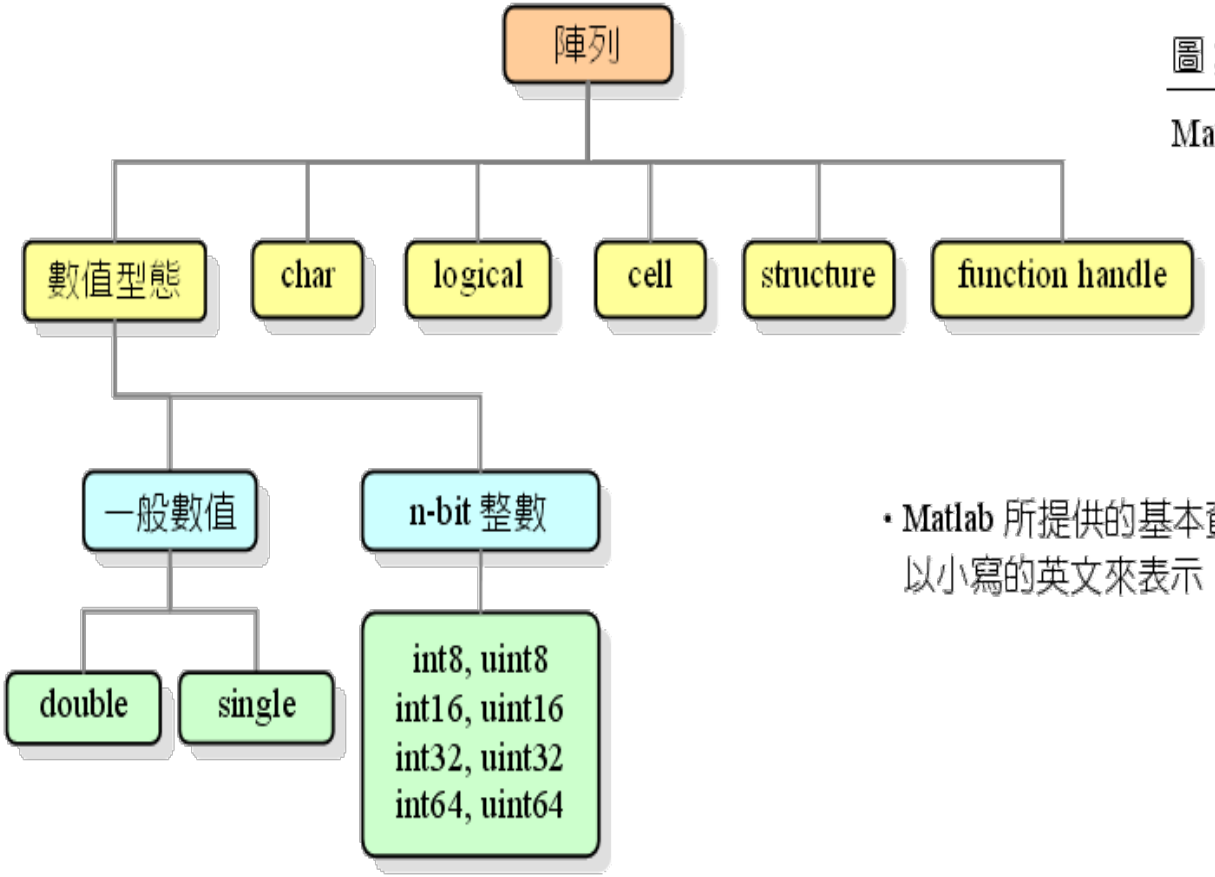


圖 3.1.1

Matlab 提供的資料型態

- Matlab 所提供的基本資料型態以小寫的英文來表示

何謂函式握把?

- **函式握把（function handle）**是 MATLAB 的一種資料型態，它保有呼叫一個函式所需的資訊。
- MATLAB 提供兩種方法來產生函式握把：
  1. 使用 @ 運算子來產生函式握把時，只需要將運算子放在函式名稱前面即可。

     ```
     fh = @函式名稱;
     ```

  2. 使用 str2func 產生函式握把時，必須將函式的名稱當成輸入引數的字串。

     ```
     fh = str2func('函式名稱');
     ```

函式握把的範例 (1/2)

使用者定義的函式: my_func.m

```
function y = my_func(x)

y = x.^2 - 2*x + 1;
```

```
>> fh1 = @my_func;  % 宣告 fh1 為一個函式握把
>> [fh1(4), my_func(4)]

ans =

   9    9

>> fh2 = @randn;
>> fh2() % 若沒有輸入引數時，必須加上圓括弧 ()

ans =

  -0.1241
```

函式握把的範例 (2/2)

```
>> whos
 Name   Size   Bytes   Class
 ans    1 × 1  8       double
 fh1    1 × 1  32      funciton_handle
 fh2    1 × 1  32      funciton_handle
>>> feval(fh1, 4) % 效果與 fh1(4) 相同
ans =
  9
>> func2str(fh1) % 找回函式握把 fh1 的原函式名稱
ans =
  my_func
```

處理函式握把的 MATLAB 函式

函式	描述
@	產生函式握把。
feval	使用函式握把來求取函式的數值。
func2str	找回一個函式握把之函式名稱。
functions	由函式握把找回其各種資訊，並以結構形式傳回資料。
str2func	從特定的字串產生一個函式握把。

使用函式握把的重要性

1. 可將函式的存取資訊傳遞給其他的函式。
2. 可增進須重複運算的程式執行效率。
3. 允許子函式與專用函式更廣泛的存取範圍。
4. 包含更多函式在同一 M 檔案裡以方便函式檔案的管理。

Section 7.4
匿名函式

- **匿名函式（anonymous function）**是一個「沒有名稱」的函式。
- 它是用單一行 MATLAB 敘述所宣告的函式，而且會回傳一個函式握把，然後可以用此握把來執行此函式。
- 可以在指令視窗裡直接定義一個匿名函式，而不用把函式寫在 M 檔案裡。
- 一般形式為

表 8.6.1　匿名函數的定義

指 令	說 明
fname=@(*arg_list*) *expr*	定義匿名函數，函數名稱為 *fname*，輸入引數為 *arg_list*，函數的內容則定義在 *expr* 的位置

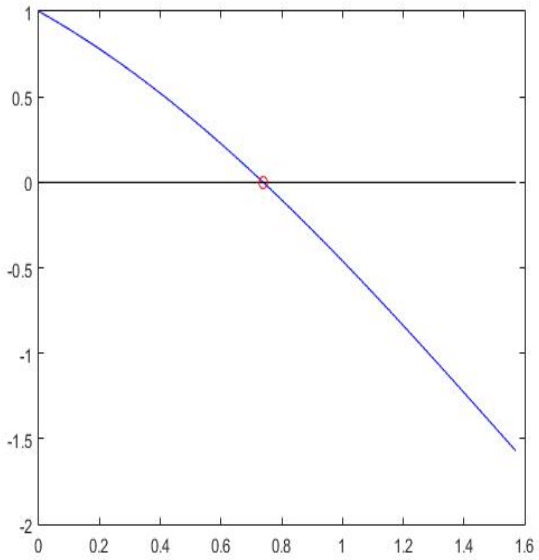
單自變量函數的範例

```
>> f = @(x) cos(x) - x; % 宣告 f 是一個函式握把
>> z = fzero(f,[0,pi/2]) % 求出函數 f 的零根

z =

  0.7391

>> x = linspace(0,pi/2); y = f(x); fz = f(z);
>> plot(x,y,'b-',x,zeros(size(x)),'k-',z,fz,'ro');
```



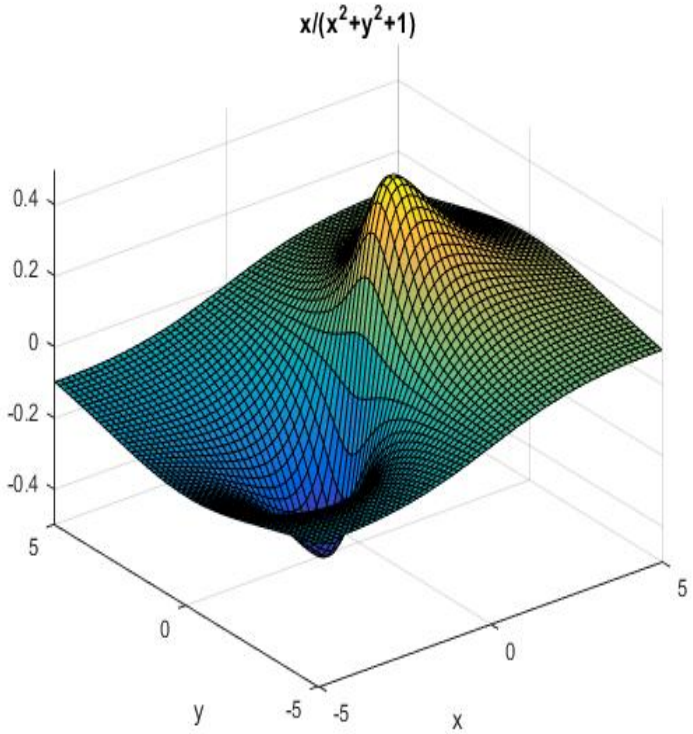
雙自變量函數的範例

```
>>  g = @(x,y) x./(x.^2+y.^2+1);
>> g(1,1)

ans =

  0.3333

>> ezsurf(g,[-5,5,-5,5]);
```



Section 7.5
遞迴函式

- 如果一個函式呼叫它自己的話，此函式稱之為**遞迴函式** (**recursive function**) 。
- 階乘函數是遞迴函式的一個好例子，其定義如下:

$$f(n) = n! = \begin{cases} 0! = 1, & n = 0 \\ n \cdot (n-1)! = n \cdot f(n-1), & n > 0 \end{cases}$$

遞迴函式的範例: fact.m

```
function result = fact(n)

if n == 0
  result = 1; % f(0) = 0! = 1
else
  result = n * fact(n-1); % f(n) = n * f(n-1)
end
```

Thank you for your attention!