

# 非線性多變量方程式的根

last modified April 23, 2012

某些應用場合會遇到非線性的聯立方程式 (system of nonlinear equations) 求解的問題。通常無法得到解析解 (closed-form solution), 必須借助演算法才能取得近似解。本文直接訴諸現成的 MATLAB 指令, 只要妥善使用, 都可以解決問題。

**本章將學到關於程式設計**

Third-party 指令的使用。

〈[本章關於 MATLAB 的指令與語法](#)〉

指令:LMFsolve (MATLAB file-exchange), fsolve (optimization toolbox)

## 1 背景介紹

非線性多變量方程式表示如下:

$$F(\mathbf{x}) = \mathbf{0} \quad (1)$$

其中  $F(\mathbf{x})$  代表函數向量,  $\mathbf{0}$  則是零向量。譬如以下的非線性聯立方程式,

$$\begin{aligned} x_1^2 + x_2^2 - 2 &= 0 \\ e^{x_1-1} + x_2^3 - 2 &= 0 \end{aligned}$$

以 (1) 的方式表達, 則

$$F(\mathbf{x}) = \begin{bmatrix} x_1^2 + x_2^2 - 2 \\ e^{x_1-1} + x_2^3 - 2 \end{bmatrix}$$

## 2 練習

本節舉幾個簡單的範例, 示範 MATLAB 解決如 (1) 的非線性聯立方程式的指令 LMFsolve 的使用方式。<sup>1</sup>

1.

$$F = \begin{bmatrix} x_1^2 + x_2^2 - 2 \\ e^{x_1-1} + x_2^3 - 2 \end{bmatrix} \quad (2)$$

2.

$$F = \begin{bmatrix} 2(x_1 + x_2)^2 + (x_1 - x_2)^2 - 8 \\ 5x_1^2 + (x_2 - 3)^2 - 9 \end{bmatrix} \quad (3)$$

3.

$$F = \begin{bmatrix} 4(x_1 - 2)^3 + 2(x_1 - 2)x_2^2 \\ 2(x_1 - 2)^2x_2 + 2(x_2 + 1) \end{bmatrix} \quad (4)$$

---

<sup>1</sup>LMFsolve 指令可以從 MATLAB file-exchange 中下載, 網址 <http://www.mathworks.com/matlabcentral/fileexchange/>。

4.

$$F = \begin{bmatrix} -400x_1(x_2 - x_1^2) - 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{bmatrix} \quad (5)$$

典型的 LMFsolve 指令如:

```
[Xf, Ssq, CNT] = LMFsolve(FUN, Xo, Options)
```

其中輸入部分,FUN 指函數向量的設定,Xo 是初始值設定,Options 通常用來做停止條件的設定。輸出部分,Xf 指方程式根的估計值, Ssq 指 Sum of squares of residuals, 即  $\|F(Xf)\|^2$ , CNT 是執行演算法的迴圈數。以 (2) 為例, 程式可以寫成

```
FUN = @(x)[x(1)^2 + x(2)^2 - 2; exp(x(1) - 1) + x(2)^3 - 2];  
Xo = [2 2];  
[Xf, Ssq, CNT] = LMFsolve(FUN, Xo)
```

上述指令沒有加入 Options, 也就是使用預設的條件。LMFsolve 提供多少條件可供選擇, 可以由下面的指令取得。

```
options = LMFsolve('default')
```

進行設定時, 可以這麼做:

```
options = LMFsolve('default');  
options = LMFsolve...  
    (options,...  
    'XTol',1e-6,...% norm(x-xold,1)  
    'FTol',1e-12,...% norm(FUN(x),1)  
    'MaxIter',100... % Maximum iterations  
    'Scaled',[ ],...% scale control, [ ]=Identity matrix  
    'Display',1...% display result every iteration  
    );  
[Xf, Ssq, CNT] = LMFsolve(FUN, Xo, options)
```

習者可以嘗試不同的初始值設定, 看看 LMFsolve 是否給出不同的答案, 並藉由觀察輸出值  $Ssq$  與輸出的迴圈過程, 了解問題本身的複雜程度或困難度。

MATLAB optimization toolbox 提供了 fsolve 指令。使用方式與 LMFsolve 差不多, 不過 fsolve 使用不同的演算法, 選擇性多, 在 options 的設定上有更多選擇, 習者自行參酌手冊, 配合本節的題目, 應該可以駕馭這個指令。