

MATLAB 的副程式

last modified April 24, 2007

副程式 (function) 的應用在一般程式語言裡非常的普遍, 幾乎是必備的技巧, 只要稍具複雜度的程式多半會使用到副程式。當程式出現重複的程式片段, 或是該程式片段過長, 佔據過多篇幅, 妨礙了程式的流暢與可讀性時, 可以將該程式片段移出, 另置他處 (在同一程式檔案或其他檔案), 稱為副程式或子程式 (subroutine)。副程式除了可以減少主程式的程式碼, 提高主程式的可讀性外, 有時候也可以供其他程式呼叫, 提高程式寫作的效率與一致性。另外 MATLAB 在 7.0 版以後新增「匿名函數」功能, 減輕副程式的帶來的檔案管理問題, 對於小型的副程式提供更簡潔的作法。

本章將學到關於程式設計

MATLAB 的副程式與匿名函數的結構與應用。

[\(本章關於 MATLAB 的指令與語法\)](#)

指令: function

1 背景介紹

主程式與副程式間的「溝通」靠參數的傳遞來聯繫。主程式將一些既定的參數傳給副程式作為程式執行所需的變數，副程式執行完畢，將結果以一組參數傳回給主程式。副程式的角色相當於主程式的「協力廠商」，主程式將部分工作「outsource」出去給副程式做。而副程式扮演「協力廠商」的角色，通常只做些單純的工作，因此可以設計成更為「通用」、為更多的「主程式」服務。

副程式的結構隨程式語言的不同有些差異，請參考 MATLAB 對於副程式的定義，以下舉個簡單的範例說明 MATLAB 副程式的結構與使用方式：

```
副程式:stats.m
```

```
function [mu,sigma]=stats(x)
mu = mean(x);           %計算平均值並且放入輸出變數 mu
sigma = std(x);         %計算標準差並且放入輸出變數 sigma
```

```
主程式:main.m
```

```
x=chi2rnd(2,1,100);     %產生任意的資料
[mu, sigma] = stats(x); %呼叫副程式 stats, 並且傳入一個資料參數 x,
                        %傳回結果放在 mu 及 sigma 的變數裡面。
                        供後續程式使用。
```

通常副程式的函數名稱 (function name) 與檔名相同。一般而言一個副程式檔案內含一個副程式，但必要的時候，也可以超過一個副程式，副程式之間彼此呼叫。此外，副程式與主程式間溝通的管道靠輸入與輸出變數，其個數是需要都可以超過一個，當然個數太多時，寫起來很冗長不方便，可以考慮採結構性 (structure) 的變數，彈性也更大。關於結構性的變數如何設定與使用，請參考手冊。

另外，副程式的執行過程與主程式完全不相干，變數名稱也可以不同，輸入與輸出變數依出現的次序為對應關係，並非依變數名稱。在副程式內的任何變數也是獨立的，與主程式無關，並不影響主程式的變數或命令視窗的變數。

2 練習

範例1: 寫一支程式分別自「常態」、「卡方」與「二項」分配的母體中抽取100樣本, 並分別計算其「樣本均數」、「標準差」、「中位數」、「最大值」及「最小值」。其中計算的部分模仿上述的副程式, 加以擴充即可。

上述的範例是副程式使用的典型, 其優點讓主程式看起來比較「乾淨清爽,」也可以提供不同的程式呼叫。另一種副程式的使用時機如範例1的積分函數所示:

範例2: 使用MATLAB 提供計算數值積分的指令`quad`, 計算 $\int_0^2 \sin(x)dx$ 。

其使用方式有以下三種典型:

方式1	方式2	方式3
<code>quad('sin',0,2)</code>	<code>F = inline('sin(x)');</code> <code>quad(F,0,2);</code>	<code>quad(@my_int_fun,0,2);</code>

其中方式3,@之後的`my_int_fun`代表一個副程式的名稱。該副程式寫成:

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

上述三個方式中, 方式1與2適用於當欲積分的函數很單純時。若函數複雜, 以副程式的方式來描述積分函數是比較恰當, 甚至是唯一的方式。MATLAB 在7.0版以後, 提出「匿名函數」(anonymous function) 的方式, 免除副程式需要建立一個檔案的麻煩, 卻可以提供重複使用的方便性, 類似上述方式2的作法, 但更具彈性。以上述的積分為例, 寫成

```
f=@(x)sin(x);
quad(f,0,2);
```

f 代表 sin 函數, 也可以拿來計算函數值, 譬如, $f(\pi/2)$ 計算 $\sin(\pi/2)$ 。

試試看以下稍複雜些的積分式。

範例3: 模仿範例2的三種方式分別計算:

- $\int_0^5 100e^x dx$ (ans: 14741.32)
- $\int_0^7 x^2 e^{-x} dx$ (ans::1.94)
- $\int_{-1.96}^{1.96} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx$

匿名函數或 inline 式的副程式還可以提供額外的參數, 增加函數使用的方便性。舉多項式函數的計算為例:

```
f=inline('polyval(p,x)');  
p=[1 -3 2];  
f(p,3);  
pp=polyder(p);  
f(pp,3);
```

上述的範例分別計算 $f(3)$ 與 $f'(3)$, 其中 $f(x) = x^2 - 3x + 2$ 。inline 指令中放進了一個參數 p , 隨著 p 的不同, f 代表著不同的多項式函數。

3 觀察

1. 副程式的使用除了解決程式碼重複的問題外, 也提供一個可重複使用 (re-useable) 的程式, 提供給別的程式呼叫。但副程式的本身也製造出管理的問題, 太多的副程式往往讓程式設計師本身迷航了。所以什麼樣的程式碼適合編成獨立的副程式, 必須認真思考, 包括長度, 可重複使用的頻率。如果不合乎這些法則, 寧願還是以程式碼的方式存在原程式中。

2. MATLAB 自 6.5 版後加入「程式加速」的機制, 在合乎某些條件下, 即使使用多層的迴圈 (for loop), MATLAB 的加速機制依舊可以適當的轉換程式碼, 加速程式的執行。不過副程式的呼叫卻違背這些條件, 讓加速機制失效。因此, 當程式的執行時間很關鍵, 或是程式耗費太多時間, 都要考慮限制副程式的使用。

4 作業

1. 將之前寫過計算多項式微分的程式改成副程式的結構, 使其使用上像呼叫 MATLAB 的指令。
2. 將之前寫過多項式積分的程式改成副程式的結構, 使其使用上像呼叫 MATLAB 的指令。
3. 寫一個標準的副程式, 專用來計算多項式的函數值 (功能像 *polyval*)。輸入參數定義為多項式的係數。