MATLAB 的程式檔：以微分的計算爲例

last modified March 30, 2009

MATLAB作爲一個數學計算的工具, 不只是提供許多計算的指令而已, 它還是一個電腦語言 (接近 C 語言), 不但充分運用了一般電腦語言的邏輯能力, 還結合數學相關的指令成爲一個具高度計算能力的程式語言, 可以解決複雜程度較高的數學問題。本章以微分的計算初窺 MATLAB 解決數學問題的典型與能力, 也是從學習 MATLAB 的指令跳脫到 MATLAB 程式的第一步。

本章將學到關於程式設計

MATLAB的程式結構、程式的邏輯概念與程式檔案的管理。

⟨本章關於 MATLAB 的指令與語法⟩
操作元 (operators): .^ ./
指令:polyval, format long, semilogx, gtext, diff, polyder, log, fprintf, input
語法: 註解的使用時機與方式。

1 背景介紹

微分根據其定義:

$$f'(x) = \frac{df(x)}{dx} = \lim_{h \to 0}(\frac{f(x+h) - f(x)}{h}) \tag{1}$$

代表函數在某個位置 (x) 的變化情形 (速率), 是陡峭的還是平緩的, 在不同的應用領域上有其特殊的意義。這種瞬間的變化在函數上以極限 $(\lim)$ 的觀念來表達; 當函數從 x 移動到 $x+h$時, 函數值的變化從 $f(x)$ 到 $f(x+h)$, 這個變化速率表示爲

$$\frac{f(x+h) - f(x)}{(x+h) - x} \tag{2}$$

h 愈小代表愈短距離的速率, 當 h 小到幾乎等於 0 時, 稱爲函數在 x 的瞬間變化率 (一般也稱爲「斜率」), 代表函數在這個點上的動態表現。圖 1 表示出函數在兩個點上的變化情形。

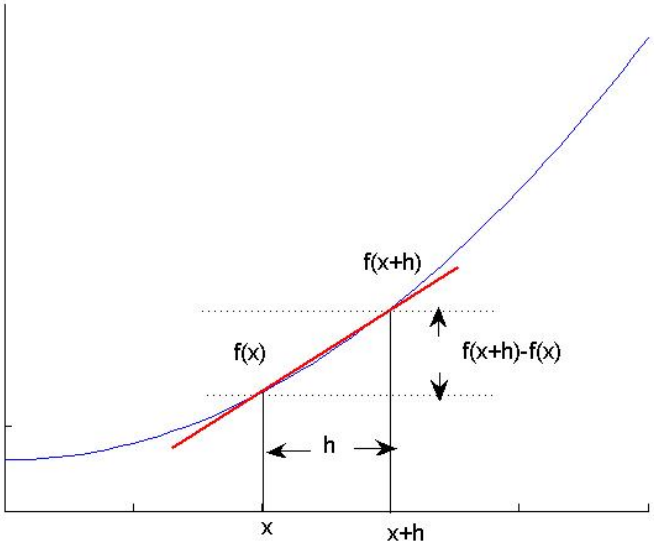

圖 1: 微分示意圖

配合函數的型態, 已推展出許多計算極限的微分技巧, 這些技巧爲人所用所熟悉, 但是電腦可沒這麼聰明, 雖有人工智慧的輔助, 還是不及人腦來得靈光。而且技巧總是有

限, 碰到一些「不友善」的函數時, 往往不夠用。這時電腦可派上用場了, 除了可以迅速並大量計算微分問題外, 函數的「長相」通常它是認不得的。撇開像 Mathematica 或 Maple 及 MATLAB Simulink(骨子裡也是 Maple) 這類可以根據數學符號就能計算微分的軟體, 以電腦程式來計算數值微分, 通常會從上述的定義來下手。以下列舉幾個範例來練習, 看看電腦如何計算函數 $f(x)$ 在任一點的導數 (或稱切線斜率)。[1]

2 練習

範例1: 假設$f(x) = x^2 + 3x + 1$, 求$f'(3)$。

- 定義上的極限lim 在數值計算上如何表示?
- h值該給多少才算趨近於0? 試試看當 $h = 0.1, 0.01, 0.001, 0.0001$時, 導數值分別爲多少? 與實際值相差多少? 哪一個選擇比較好? 導數值由下列公式計算:

$$f'(3) = \frac{f(3+h) - f(3)}{h}$$

上述的公式需要計算兩個多項函數值, 輸入時稍嫌麻煩, 此時可以使用 MATLAB 提供計算多項式函數的指令$polyval$, 可以讓整體指令「乾淨」許多, 譬如計算 $f(3)$ 可以依序輸入下列指令

```
p=[1 3 1] %函數的係數
f=polyval(p,3)
```

MATLAB的命令視窗將指令執行的結果顯示出來, 爲避免畫面的紊亂, 僅顯示到小數第四位, 若要觀察更多的小數位數, 可以執行以下指令延長到14位。

```
format long
```

[1]有一門領域叫做「數值分析,」專門研究如何利用電腦來解決數學計算上問題。關於「微分」當然也發展出許多的演算法(Algorithm), 克服各式各樣可能在電腦上產生的問題, 本文僅就最簡單的方式著手, 避免涉及太多細節, 以免干擾本文眞正的目的: 介紹 MATLAB 的程式功能。

如要恢復原來的四位數, 執行指令 format 即可。

範例2: 從上一個練習得知, 計算數值導數時,h 值的選擇會影響導數值。對於這個函數你能找出一個最好的 h 值嗎? 畫一張圖看看 h 值的選擇對計算導數的影響。畫圖時 x 軸的選擇很重要, 可以採用 semilogx 指令取代 plot 可以看得更清楚。

MATLAB的特點是矩陣的運算, 本練習要觀察不同的 h 值計算出的導數值, 便可以利用這項特質。譬如建立一個變數 h, 其內容涵蓋欲觀察的 h 值, 如

$$h = \left[\begin{array}{cccccccccc} 10^{-4} & 10^{-5} & 10^{-6} & 10^{-7} & 10^{-8} & 10^{-9} & 10^{-10} & 10^{-11} & 10^{-12} \end{array} \right]$$

建立的方式可以如上式一一輸入, 但最快也最能掌握 MATLAB 精髓的方式爲:

$$h = 10 \cdot\hat{}\, (-4 : -1 : -12) \text{ 或 } h = 1 \cdot /10 \cdot\hat{}\, (4 : 12)$$

利用 MATLAB 提供的運算元 $(\cdot)$, 可以同時針對矩陣或向量的每一個元素做計算, 依序執行下列指令, 並逐一觀察每個指令執行的結果, 將對 MATLAB 的使用有進一步的瞭解。值得注意的是, 第二個指令的分母不需刮號, 理由是指數運算元的計算順序優於其他。

```
p=[1 3 1];                    %函數的係數
h=10.^(-4:-1:-12); %列出所有 h 值
x=3;
f1=polyval(p,x);              %計算 f(3)
f2=polyval(p,x+h);            %計算不同 h 值下的函數值 f(3+h)
fp=(f2-f1)./h;                %計算不同 h 值下的導數值
semilogx(h,fp)
```

圖 2 展示上述指令的執行結果。透過同時計算比較大範圍的 h 選擇, 從圖形可以清楚的看出 h 的選擇對導數的影響, 也看出過小的 h 值產生電腦計算上困擾, 如圖形的

左邊, 可見在電腦上極限值的選擇仍有其侷限, 並非無限制的趨近 0。圖 2 也同時展示 MATLAB 在繪圖區強大的編輯功能, 幾乎所有顯示在上面的元件 (如座標軸、背景、刻度、線條顏色粗細樣式...) 都可以透過繪圖區的編輯工具進行調整。當然所有的編輯也都可以經由適當的 MATLAB 指令, 不過用滑鼠編輯還是比用鍵盤簡單許多, 也比較好玩。

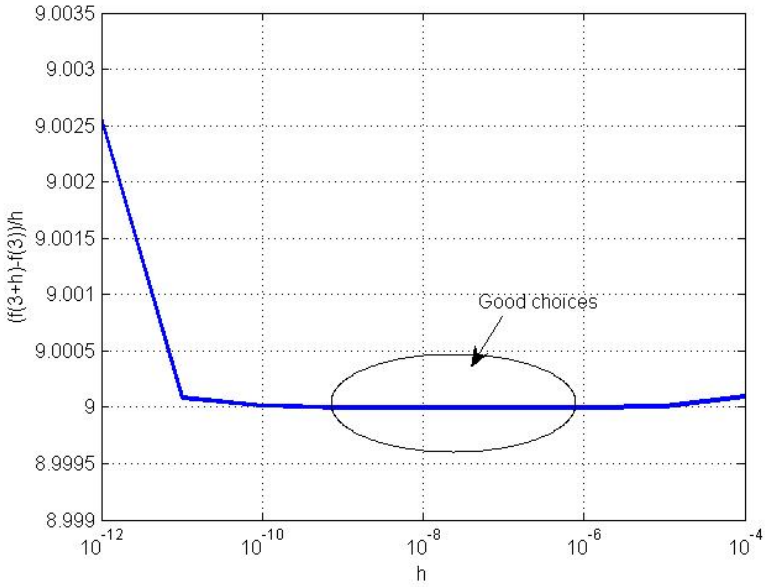

圖 2: h 值的選擇對導數計算的影響

範例3: 前述的練習需要幾個指令來完成導數的計算, 如果每計算一次就要重複執行相同的指令, 代入不同的數值, 那將會非常沒有效率, 更何況面對複雜的數學問題可能要花數十, 甚至數百條指令來處理。在這個情況下,MATLAB 允許將這些指令集中起來, 存放在一個檔案裡面, 藉由在命令視窗輸入檔名, MATLAB會循序執行檔案裡面的每一條指令。這個檔案被稱爲 MATLAB 的程式。請將上述範例計算導數的指令集中起來, 存成一個檔案 (副檔名自動設定爲 .m), [2] 譬如,derivative.m

---

請注意要 MATLAB 正確的執行程式檔, 必須符合下面兩個條件之一:

[2]請注意檔名的選擇, 不能與MATLAB 內建的指令名稱相同, 那將影響該指令的功能。

- 「命令視窗」所在的位置必須相同於程式檔。這可藉由變更命令視窗的「current directory」達到。
- 將程式檔所在的目錄設爲 MATLAB 可以自動參考的路徑。這可藉由「File」⟶「Set Path」⟶「Add Folder」達到。

將所有必要的指令放在一起, 不但方便執行, 也容易管理與維護, 更提供程式寫作許多可以運用的技巧。

範例4: 範例2針對特定函數 $f(x) = x^2 + 3x + 1$ 與特定的 x 值, 試圖找出計算數值導數 (2) 時適當的 h 值。在這個範例裡, 我們想知道對於特定的 h 值, 譬如 $h = 0.1$, 數值導數 (2) 與實際的導函數 $f'(x) = 2x + 3$ 相差多少?

圖 3 清楚的展示不當的 h 值 $(h = 1)$ 計算出來的導數與實際值的差距。當 $h = 0.1$ 時, 數值導數與實際導數非常貼近, 但 h 究竟該多小才是恰當需依實際情況而定。圖 3 選用了繪圖區的「Legend」工具鍵來區分不同線條的意義, 提供 gtext 或 text 指令外的另一種選擇。

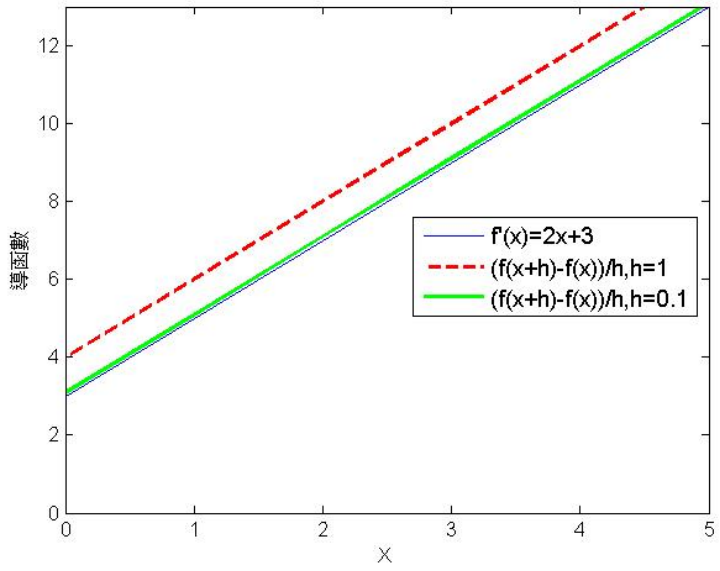

圖 3: 實際的導函數與數值導數在 $h = 1, 0.1$ 的比較

範例5:MATLAB 提供什麼指令計算數值導數呢?(hint: diff,polyder)。試著用 MATLAB 指令的方式來計算導數值。觀察看看, 與你前面所計算的值相差多少? 哪一個比較準?

像 MATLAB 這種等級的數學軟體當然會提供計算導數的功能指令, 究竟 MATLAB 計算導數的指令與自己根據定義所寫的程式有何不同? 這是令人好奇也值得一探的。如果發現 MATLAB 的指令執行結果令人滿意, 未來程式有需要自然可以直接引用, 不一定要使用自己寫的, 畢竟專業的程式設計還是比較縝密, 使用上也比較方便。

3 觀察

寫程式解決數學問題常會觀察到一些平常看不到的「內涵。」透過這些觀察與反覆思考查證, 往往會對理論有進一步的體會, 那可是平常人無法直接從書本獲得的。以下列舉些觀察點供參考。

- h值是影響數值計算的重要因素, 是否在所有的練習中可以歸納出一個適當的h值, 對所有的函式都適用? 對於h值的選擇, 你有哪些觀察呢?

- MATLAB 利用單一指令解決單純問題。當一個指令無法解決問題時, 可藉由指令的「接力」執行, 共同合作完成任務。多個指令組合 (或稱程式) 並且一起被執行, 方便解決較複雜的問題。但當指令被組合起來時, 必須注意其前後的合作關係, 如何安排變數、如何決定運算的次序、如何適當的採用 MATLAB 的內建功能, 讓程式執行順暢、有效率, 而且易懂、容易變更、擴充, 都是非常重要的程式寫作技巧。初學者應細心體會其中的奧妙。多看別人的程式, 比較自己的程式, 從中獲得眞正的寫作技巧。

- 程式的執行遵循從上而下的規則, 但程式的寫作可不一定要從第一行寫起。可以從做容易下手 (最直覺) 的地方著手, 再依程式的需要補足前後面的指令, 這樣反而比較順手, 速度比較快。因爲程式語言的邏輯與人腦還是不同的, 因此程式寫作的過程常常是上上下下穿插的寫, 並且不斷的執行部分過程, 以確定指令與

邏輯的正確, 千萬不可妄想一次寫完再執行, 往往會連錯在哪裡都不知道, 徒增除錯的困難。

- 本單元著重在多個指令的組合, 共同完成一件工作。不過在程式的寫作過程中或是最後完成的作品, 常需要能看得到程式執行的結果, 這時候一些文字輸出的指令顯得重要, 譬如 fprintf 的使用更爲常見, 以多項式函數的微分爲例, 假設我們選擇 $h = 10^{-8}$ 做爲 h 趨近於零的替代, 下列程式可以直接在命令視窗顯示 $f'(3)$ 的結果, 其中多項式 $f(x) = x^2 + 3x + 1$。。

```
p=[1 3 1];                    %函數的係數
h=10^-8;                      %適當的 h 值
x=3
f1=polyval(p,x);              %計算 f(3)
f2=polyval(p,x+h);            %計算不同 h 值下的函數值 f(3+h)
fp=(f2-f1)/h;                 %計算數值導數
fprintf('The derivative of f(x) at x=%7.2f is %10.4f \n',x,fp)
```

程式語言都有指令將文數字 print 到螢幕上 (console),MATLAB 的指令 fprintf 就是做這件工作。[3] 上述的 fprintf 指令將 x 值與 f'(x) 輸出到命令視窗, 且爲了方便察看, 在第一個格式參數 (format) 裡面放了必要的文字。格式參數是 fprintf 使用最主要的部分, 在單引號裡面包含兩個部分, 一是文字, 另一個是格式, 穿插放置, 將變數的結果與文字一同呈現出來。格式部分只要是將變數 (自第二個參數起, 如 x,fp) 以適當的格式呈現出來, 譬如格式 %7.2f,「f」代表 fixed-point, 用來呈現數值, 前方的數字 (如 7.2) 代表欲呈現出來的實數數字長度, 其中小數點後的數字 2 代表小數位數, 7 代表「整數位數 + 小數位數 + 1」, 換句話說, 整數位數是 4。最後的\n會造成換行的效果, 否則下一個 fprintf 會連續在後面。MATLAB 也提供除了 f 以外的特殊格式, 詳見線上手册。

[3] fprintf 不只可以將結果「寫」到命令視窗, 也可以寫到檔案裡面, 詳細的使用方式可以在線上手册上看到。

這支程式看以來還不夠好, 如果需要測試在不同 x 值的導數, 程式必須不斷修改與存檔, 略顯麻煩。有個指令 input 提供程式執行時, 從鍵盤直接輸入, 使用方式如 (取代上述程式第三行)

```
x=input('Enter x =');
```

4 作業

1. 將數值導數的定義改爲

$$f'(x) = \frac{df(x)}{dx} = \lim_{h\to 0}(\frac{f(x+h) - f(x-h)}{2h})$$

與之前的定義做比較, 哪一個定義比較準確? 舉範例1的函數, 取幾個不同的 h 值, 譬如: $h = 10^{-2}, 10^{-3}, \cdots, 10^{-9}$, 等, 畫一張圖紀錄在每一個 h 值, 不同導數公式的誤差 (誤差=|數值導數 - 實際導數|)。即畫一張圖顯示兩條來自不同的導數定義的誤差線。

2. 從上題的結果, 你可以判斷在數值計算上哪一個定義比較好嗎? 下面這個練習可以幫助釐清一些觀念:

> 令函數爲$f(x) = x^2 + 3x + 1$, 將此函數分別代入兩個定義中, 經過一番化簡之後, 你得到什麼? 這個結果可否幫助你判斷在數值計算上哪一個定義比較好?

3. 同範例4, 畫一張圖比較函數 $f(x) = -x^3 + 6x^2$ 的數值導數與理論導數的差別。所謂理論導數, 即 $f'(x)$; 而數值導數則是利用數值方法 (例如前述的練習) 以電腦程式計算的函數值。換句話說, 想比較下列兩個函數的差異:

$$\begin{aligned} f'(x) &= -3x^2 + 12x \\ f'_{app}(x) &= \frac{f(x+h) - f(x)}{h} \end{aligned}$$

請自行選擇適當的 h 值與 x 的間距範圍; 將理論值的圖與數值計算的圖畫在一起。請注意

- 你可以選擇不同的 h 值, 藉以凸顯不當的 h 值會影響數值導數的精確度。
- x 軸範圍的選擇頗爲關鍵。範圍的大小與區域的選擇不當往往會造成視覺上的錯覺, 導致錯估兩者間的差距。不妨多觀察幾個不同的區域與範圍, 甚至可以去計算誤差的大小。

4. 同上題, 但是函式爲:$f(x) = \frac{\ln x}{x^3}$, 其導函數 $f'(x) = \frac{1-3\ln x}{x^4}$ 。選擇 $x > 0$ 的適當範圍。[4]

5. 前面兩題在凸顯 h 值的角色, 如果要訂出一個準則來決定到底多小的 h 才是恰當, Root Mean Squared Error(RMSE) 倒是不錯的選擇,RMSE 定義如下:

$$RMSE = \sqrt{\frac{\sum_{k=1}^{N}(f_h(x_k) - f(x_k))^2}{N}}$$

其中 $f_h(x)$ 代表數值導數,x_k 是選取某個範圍內的 x 值, 共選取 N 個。我們可以指定當所選擇的 h 值使得 RMSE < 0.1 即可。請爲前面兩支程式加入 RMSE 的計算, 爲每一個所選擇的 h 值計算出相對的 RMSE, 並藉此決定適當的 h 值。

6. 利用上一節「觀察」提供的指令 input, fprintf 及 計算多項式函數一次微分後的係數 polyder, [5] 寫一支可以計算任何多項式函數導數的程式, 執行時要求輸入多項式係數(例如輸入 [1 3 1] 代表多項式 $f(x) = x^2 + 3x + 1$) 及 x 值, 程式執行後輸出 $f'(x)$ 的結果。下圖是執行的過程與結果:

[4]計算對數函數 $\ln x$ 的指令爲 log(x)

[5]接觸一個新的指令, 最好先看看線上手册的說明, 在自己試試幾個簡單的東西, 多靠自己摸索, 程式功力才能躍進

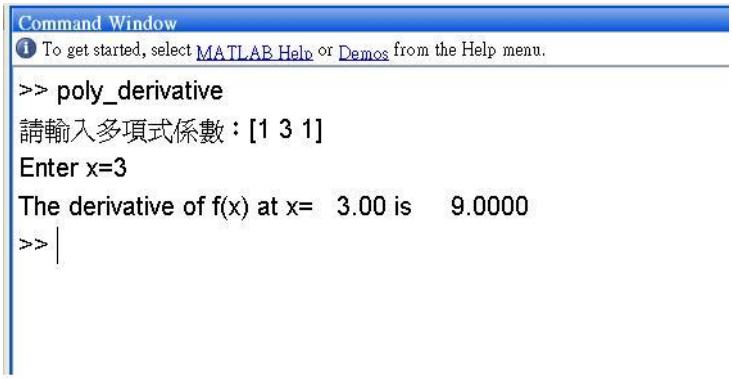
Command Window
To get started, select MATLAB Help or Demos from the Help menu.
>> poly_derivative
請輸入多項式係數：[1 3 1]
Enter x=3
The derivative of f(x) at x= 3.00 is 9.0000
>>

圖 4: 多項式導數計算