

Octave 操作簡介

Hung-Jui Chang

CYCU Applied Mathematics

April 20, 2022

- General programming language:
 - C/C++, Java, Python, ...
- Statistical language/software:
 - R, SAS, SPSS, ...
- Mathematical language/software
 - MATLAB, Maple, Mathematics, Octave

- MATLAB is a commercial software
 - Developed by “The MathWorks”
 - Using C-like syntax
 - Can interface with programs written in other language.
 - C, C++, Java, Fortran, Python.
- GNU Octave is a open-source software
 - Implemented with C++ standard library
 - Compatible with MATLAB
- In this course, we will use GNU Octave.

- GNU Octave Homepage: <https://octave.org/>
- GNU Octave Manual: <https://octave.org/octave.pdf>
- Install:
 - Windows: Downloads installer exe file and executes it
 - BSD: See the document
 - MacOS: See the documents
 - GNU/Linux:
 - `sudo apt-get install octave`
- Execution:
 - Double clicks the octave icon
 - CLI (Command Line Interface) version
 - GUI (Graphic User Interface) version
 - Or opens a terminal and types octave (in GNU/Linux)
 - Or Alt+F2 octave (in Ubuntu)

- Octave的變數不用宣告就可直接使用
 - 實數(real): $x = 1.0$, $y = 1$
 - 複數(complex number) $a = 1 + 2i$, $b = 1 + 2 * i$, $c = 2 * j$
 - i 和 j 都可以當作虛數的符號
 - 因為 i 通常會被拿去當迴圈的index，所以用 j 代替
 - 矩陣(matrix): $A = [1, 2; 3, 4]$, $B = \text{zeros}(5)$, $C = \text{magic}(5)$
 - 字串(string): $s = \text{"Thisisastring"}$
 - 結構(structure): $z.a = 1$, $z.b = [1, 2; 3, 4]$, $z.c = \text{"string"}$
- 和裝上述內容的陣列(array)
- 可以用clear把指定的變數清空。
 - 否則這個變數會一直存在
- 保存宣告的變數留到下次使用
 - `save file_name` %保留目前所有的變數
 - `load file_name` %讀取上次儲存的變數

象徵變數(Symbolic Variable)

- `syms x`，宣告 x 是一個數學上的變數
 - $x + 3$ ，會得到多項式 $x + 3$
 - $(x + 3) * (x + 2)$ 會得到兩個多項式相乘
 - 可以用`expand`把它展開
 - 也可以用`factor`進行因式分解
- $f(x) = (x+2)*(x+3)$
 - f 是一個以 x 為變數的函數
 - $f(10)$ 可以取當 $x = 10$ 時的 $f(x)$ 的值。
- `syms` 屬於套件 `symbolic`
- 使用前需要先安裝/讀取套件：
 - `pkg install -forge symbolic`
 - `pkg load symbolic`
 - 意義上近似於C/C++的`#include`和Python的`import`
- Ubuntu 16.04有套件支援問題，需要升級到18.04以上。

關於矩陣

- Matlab/Octave的矩陣index是從1開始算。
- 存取一個矩陣內的元素是使用小括號()而非中括號[]
 - 中括號用於定義矩陣
- 可以用函數size知道一個矩陣的Row和Col數
 - $A=[1, 2, 3; 4, 5, 6];$
 - $\text{size}(A)$ 會得到 $[2, 3]$
 - 在Octave中可以有多個回傳值。
 - 可以用 $[M, N] = \text{size}(A)$ 來取得回傳值。
 - 或是用 $\text{size}(A, 1)$ 和 $\text{size}(A, 2)$ 分別得到row和column的數量
- 第一個column是 $A(:,1)$
- 第一個row是 $A(1,:)$
- $:$ 表示全部的範圍
- $1:10$ 表示從數字1到10的範圍
- $x = [1:10]$

新增矩陣

- 矩陣的初始化分成兩種:
 - 已知未來的大小，先開一個矩陣給它。
 - `A = zeros(3);` %一個3x3的0矩陣，也可以是`zrtos(3,3)`
 - `A = ones(3);` %一個3x3的1矩陣，也可以是`onese(3,3)`
 - `A = eye(3);` %一個3x3的單位矩陣 I_3
 - 不知道未來的大小，一個一個新增。
 - `clear B;`
 - `for i = 1:5`
 - `B(i) = i;`
 - `endfor`
 - B 在執行完之後，會是一個1x5的矩陣。
- 當指定矩陣的一個元素(i, j)是 x 時
 - 如果矩陣的size (M, N)滿足 $1 \leq i \leq M, 1 \leq j \leq N$ ，會修改元素(i, j)的值。
 - 如果矩陣的size (M, N)不滿足 $1 \leq i \leq M, 1 \leq j \leq N$ ，會擴展矩陣的範圍得到新的滿足條件的 M 和 N ，新增的元素預設會是0。

矩陣操作

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

- 當我們用 $A=[1, 2, 3; 4, 5, 6]$ 可以得到如上的矩陣
- $B = \text{reshape}(A, 1, 6)$ 可以得到矩陣
 $B = [1 \quad 4 \quad 2 \quad 5 \quad 3 \quad 6]$
- 可以用 $B(x:y)$ 存取 B 的第 x 到第 y 個元素
 - $B(2:4)$ 會得到 $[4 \quad 2 \quad 5]$
 - $B(1:2:5)$ 會得到 $[1 \quad 2 \quad 3]$ % 從1開始每2個一數取至第5個
 - $B(6:-1:1)$ 會得到 $[6 \quad 3 \quad 5 \quad 2 \quad 4 \quad 1]$ % 從6開始每1個一數取至第1個，也就是取反序
 - $\text{fliplr}(B)$ 也可以得到一樣的結果。

矩陣翻轉

- `fliplr(A)` %矩陣左右翻轉
- `flipud(A)` %矩陣上下翻轉
- `flip(A, dim)` %矩陣翻轉，dim不加或1是上下，2是左右。
- `rot90(A, K)` %矩陣轉90度，K是轉幾次，不加是1次。

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

- `sum(A)` or `sum(A, 1)`
 - `ans = [5 7 9]`
- `sum(A, 2)`
 - `ans = $\begin{bmatrix} 6 \\ 15 \end{bmatrix}$`
- `[ans1, ans2] = max(A)` or `max(A, [], 1)`
 - `ans1 = [4 5 6]`, `ans2 = [2 2 2]`
- `[ans1, ans2] = max(A, [], 2)`
 - `ans1 = $\begin{bmatrix} 3 \\ 6 \end{bmatrix}$` `ans2 = $\begin{bmatrix} 3 \\ 3 \end{bmatrix}$`

- `A = [1:6] % [1, 2, 3, 4, 5, 6]`
- `int2str(A)` 會生成一個長度16的字串。
 - "1 2 3 4 5 6" 每個數字間有兩個空白。
- 用ASCII的觀念轉換。
- `strvcat(X) %` 把X裡面的數字用ASCII轉換成文字。
 - 97: a, 65:A, 48:0
 - `strvcat(A+48)` 會得到 "123456"

- 加減乘除: $+$, $-$, $*$, $/$
 - Element-wise calculator
 - $+.+, .-, .*, ./$
 - $A * B \Rightarrow$ 矩陣乘法
 - $A .* B \Rightarrow$ 同位置的元素相乘
 - $A .* B = C, C_{i,j} = A_{i,j} * B_{i,j}$
- 取餘數是用 `mod()`
 - `mod(5, 3)` 會得到 2
 - `%` 在 Matlab/Octave 中是註解符號
- 次方符號 \wedge : $a \wedge b$, a 的 b 次方, 也可以用 $a ** b$

常用數學函數(1/2)

- 三角函數: $\sin()$, $\cos()$, $\tan()$
- 反三角函數: $\text{asin}()$, $\text{acos}()$, $\text{atan}()$
- 對數: $\log()$, $\ln()$, $\log_{10}()$, $\log_2()$
- 自然數: $\exp()$, e
- 圓周率: π , $\pi(N, M)$
- 取絕對值: $\text{abs}()$
- 取下高斯: $\text{floor}()$ %無條件捨去取整數部分
- 取上高斯: $\text{ceil}()$ %無條件進位取整數部分

常用數學函數(2/2)

- 相加一個向量X, `sum(X)`
- 乘一個向量X中的所有元素: `prod(X)`
 - 5的階乘: `factorial(5)`, `prod(1:5)`
- 最小公倍數 `lcm`, 最大公因數 `gcd`
- 確認是否是質數 `isprime()`, 質因數分解`factor()`, N以下的所有質數`primes(N)`
- 可以用`help command`查詢一個指令或函數的用法

- if statement
 - if (condition)
 - statements;
 - endif
- if-else statement
 - if (condition)
 - statement1;
 - else
 - statement2;
 - endif
- if-elseif statement
 - if (condition1)
 - statement1;
 - elseif (condition2)
 - statement2;
 - endif
- while statements
 - while (condition)
 - statements
 - endwhile
- do statements
 - do
 - statements;
 - until(condition)
- for statements
 - for var = expression
 - statements;
 - endfor

邏輯判斷符號

- 相等 $x == y$, $eq(x, y)$
- 不相等 $x \neq y$, $x \sim= y$
- 大於等於: $x \geq y$, $ge(x, y)$
- 大於: $x > y$, $gt(x, y)$
- 小於等於: $x \leq y$, $le(x, y)$
- 小於: $x < y$, $lt(x, y)$

自訂義函式(function)

- `function return_values = function_name(parameters)`
- %函式的內容
- `endfunction`
- 一個算出輸入的兩個數的加減乘的函數
 - `function [A, B, C] = cal(X, Y)`
 - `A = X + Y;`
 - `B = X - Y;`
 - `C = X * Y;`
 - `endfunction`

- 可以把寫好的函式放在.m檔中
- 放在和目前的工作目錄(working directory)同一個地方
 - 可以用pwd檢查目前的工作目錄
- 把函數的名字和檔名取成同一個名字
 - e.g. myfun.m
 - 和java的命名法相似。
- 之後就可以呼叫自己寫的函數

繪圖: `plot(x, y)`

- 一般數學軟體中是以描點內差的方式畫圖。
- Example:
 - `X = -1 : 0.1 : 1;` %X是從-1到1每隔0.1取一點的向量
 - `Y = sin(X);` %Y是每一個取樣點X對應的sin函數值
 - `plot(X, Y);` %畫出X向量和Y向量表示的圖，其中每個點是由 $(X(i), Y(i))$ 所組成。

- `plot3(X,Y,Z)`和 `plot`的使用法相似，可以把X,Y,Z向量中的點描繪出來。
- `surf(X,Y,Z)`: 根據X,Y,Z向量，畫出描繪的點形成的曲面
- 決定X, Y的取樣點:
 - `X = linspace(-1, 1, 25);` %從-1到1取樣25點。
 - `Y = linspace(-1, 1, 25);` %從-1到1取樣25點。
 - `[XX, YY] = meshgrid(X, Y)` % XX和YY是 25 by 25的矩陣，相當於:
 - `XX = [X;X;X;X;X;X;X;...;X]` % 把X複製25次
 - `YY = [Y;Y;Y;Y;Y;Y;Y;...;Y]'` % 把Y複製25次然後轉置
 - `ZZ = sqrt(3 - XX.**2 - YY.**2)` % 假設要畫的是球面
 $z^2 = 3 - x^2 - y^2$
 - `surf(XX, YY, ZZ);`

- C-style input/output function

- From stdin: `printf("...",)`
- To stdout: `scanf("...",)`
- open file:
 - `file_name = "Test File.txt"`
 - `fid = fopen(file_name, option)`
 - option 可能是寫入 "w" 或讀取 "r" 等。
 - `input_file = fopen("TestIn.txt", "r")`
 - 注意檔案開啓需要有正確的路徑。
 - Windows 和 Linux 系的目錄分隔符號一個是 "\" 一個是 "/"
- From file: `fscanf(fid, "...",)`
- To file: `fprintf(fid, "...",);`
- close file: `fclose(fid)` % 檔案用完後記得關起來。
- 三個預設可以直接用的 fid: `stdin`, `stdout`, `stderr`