

Phần mềm tính toán khoa học

MATLAB và Octave

Lê V. Hải¹

Người trình bày: Vinh Khang²

¹Department of Mathematics, Penn State University, USA.

²Department of Computer Science, Liberty University, USA.

PiMA Summer Camp, August 2016

Mục chính

- 1** Giới thiệu
 - MATLAB
 - Octave
- 2** Cơ bản
 - Giao diện
 - Variables (Biến)
 - Phép tính cơ bản
- 3** Chức năng
 - Plots (Đồ thị)
 - Built in Functions (Hàm có sẵn)
- 4** Lập trình
 - Scripts
 - Functions (Viết hàm)
 - Flow Control (Vòng lặp)

Tải Octave

Do MATLAB cần phải trả phí, trong trại hè chúng ta sẽ dùng Octave trên máy tính cá nhân, và nếu được, có thể sẽ dùng MATLAB ở phòng máy của KHTN.

Tải Octave: <https://ftp.gnu.org/gnu/octave/windows/> (chọn octave-4.0.0_0-installer.exe)

Nếu trực trặc không download được, có thể dùng Octave trực tiếp ở đây, các chức năng tương tự, chỉ có giao diện là hơi khác: <http://octave-online.net/>

Nguồn gốc và ứng dụng

MATLAB là một phần mềm tính toán khoa học, phát triển bởi công ty MathWorks và được sáng lập bởi nhà toán học, lập trình gia Cleve Moler. Ông Moler viết phần mềm này nhằm giúp đỡ các học sinh ở đại học University of New Mexico, Hoa Kỳ lập trình dễ dàng hơn mà không phải dùng một ngôn ngữ phổ biến lúc bấy giờ, Fortran. Năm 1984, MATLAB chính thức được thương mại hóa và được phổ biến ở nhiều nơi trên thế giới.

Nhược điểm của MATLAB đó là phải trả phí khá cao để được dùng.

Hình: Cleve Moler, người sáng tạo MATLAB

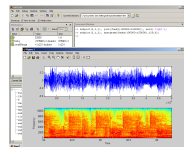
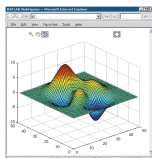
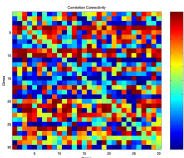


Hình: Logo của MATLAB



Nguồn gốc và ứng dụng

MATLAB là viết tắt của từ "**MAT**rix **LAB**oratory." Đúng như tên gọi, ưu điểm lớn nhất của MATLAB là có một thư viện hỗ trợ tính toán và xử lý ma trận (matrix) cực kì hiệu quả. Chính vì lợi thế này, MATLAB hiện đang được dùng ở nhiều viện nghiên cứu, trường đại học cũng như trong công nghiệp ở các lĩnh vực như đại số tuyến tính (linear algebra), xử lý hình ảnh (image processing), học máy (machine learning).



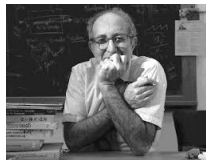
Hình: Ứng dụng của MATLAB (Nguồn: Google Images)

Nguồn gốc và ứng dụng

Octave là một ngôn ngữ lập trình dùng trong việc tính toán khoa học như MATLAB. Vì là một ngôn ngữ mã nguồn mở thuộc **GNU Project**, Octave hoàn toàn miễn phí. Ngoài ưu điểm này, điểm đặc biệt của Octave là có thể chạy các chương trình và có giao diện giống như MATLAB. Nhược điểm là thư viện tính toán không có đồ sộ như MATLAB, nhưng các thuật toán cơ bản Octave đều có thể hỗ trợ.

Octave sơ khai chỉ là một dự án cho một lớp hóa học ở Đại học bang Oregon, Hoa Kỳ. Năm 1992, John W. Eaton phát triển thành một phiên bản hoàn chỉnh. Octave lấy tên từ một giáo sư hóa học ở đại học Oregon, Octave Levenspiel, do Eaton khá là ngưỡng mộ khả năng tính nhẩm của Octave.

Hình: Octave Levenspiel

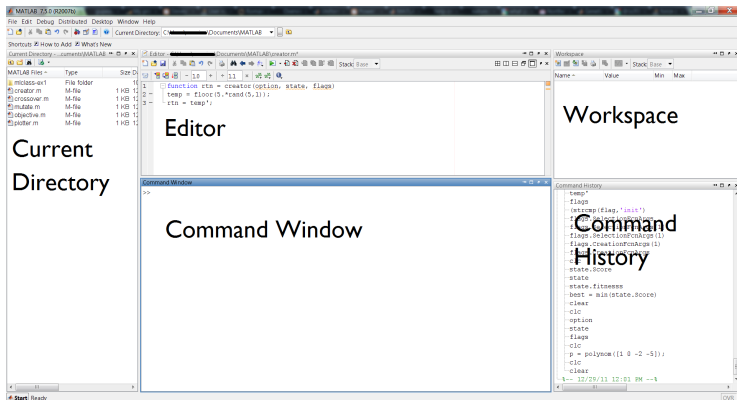


Hình: Logo của Octave



Cửa sổ MATLAB

Hình: Giao diện MATLAB



Thành phần chính của MATLAB (Octave tương tự)

MATLAB hoặc Octave

- Editor (Nơi viết code)
- Command Window (Nơi chạy câu lệnh)
- Workspace (Nơi lưu trữ giá trị các biến)
- Command History (Lịch sử các dòng lệnh)
- Current Directory (Địa chỉ thư mục đang sử dụng)

Hai thành phần chính cần chú ý đó là **Editor** và **Command Window**. Command Window có thể được sử dụng một cái máy tính. Làm thử một ví dụ, nhập $2+3$ trong Command Window, nhấn `Enter`, sẽ được kết là 5. Hãy thử $2^2 + 3!$

Gán biến

MATLAB hỗ trợ các biến sau

- Số: `> > alpha = 3`
- Dãy/Vector-hàng: `> > b = [1 2 3]` hoặc `b = [1,2,3]`
- Ma trận: `> > M = [1 1 1; 2 3 4; 3 1 4]` (xuống hàng bằng dấu ";")

Làm thử một ví dụ, trong Command Window nhập hai câu lệnh `> > a = 3` và `> > b = 3`, sau đó hãy thử nhập `> > a+b` rồi Enter để thấy kết quả.

Cho dấu ";" ở cuối câu lệnh để không phải hiện output.

`> > A = [1 2; 3 4];` (để ý có dấu ";" ở cuối câu, hãy thử bỏ để thấy sự khác biệt)

Câu lệnh `> > whos` để hiện ra tất cả các biến đã được đặt.

Tính toán trong MATLAB

Các phép tính cộng, trừ, nhân, chia được biểu diễn trong MATLAB tương ứng là $+$, $-$, $*$, $/$. Lưu ý khi nhân biến và số, phải dùng dấu " $*$ ", ví dụ như $2*x$; nếu viết $2x$ sẽ gây ra lỗi.

Điểm mạnh của MATLAB là có thể cộng, trừ, nhân và chia (khi được) ma trận như tính toán với số. Lấy nghịch đảo của ma trận là " A^{-1} ," chuyển vị là " A' ".



Truy cập thành phần của ma trận

Giả sử ta có một ma trận A như sau >> A = [1 2 3; 4 5 6; 7 8 9]

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Truy cập thành phần của A, ví dụ như số 2 ở vị trí hàng thứ nhất, cột thứ hai: >> A(1,2).

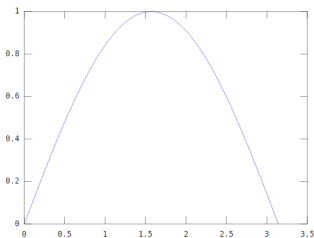
Truy cập nguyên một hàng hay cột, ví dụ như A(1,:) (hàng thứ nhất), A(:,2) (cột thứ hai). Lưu ý là dãy trong MATLAB luôn bắt đầu bằng số thứ tự "1", do đó A(0,1) sẽ không có nghĩa.

Vẽ hình trong MATLAB

Giả sử muốn có hai vectors x và y , coi như $y=f(x)$ với một hàm f nào đó. Đồ thị của hàm này được vẽ bằng câu lệnh `plot(x,y)`. Hãy thử những câu lệnh sau để vẽ đồ thị của hàm `sin(x)`:

```
> > x = linspace(0,pi,100); (chia đoạn [0, 1] thành 100 điểm)
> > y = sin(x); (tính giá trị của hàm sin(.) tại các điểm của x)
> > plot(x,y)
```

Hình: Đồ thị hàm `sin(.)`



Nếu muốn vẽ nhiều đồ thị trong một hình, gõ `hold on` trước khi vẽ.

Hàm quen thuộc

Điểm mạnh nữa của MATLAB là hỗ trợ nhiều hàm số quen thuộc. Thư viện của MATLAB hỗ trợ các hàm như `sin`, `cos`, `exp`, `log`. Ngoài ra còn nhiều hàm khác dùng cho mục đích riêng.

Có thể tham khảo <http://www.mathworks.com/help/matlab/functionlist.html> để tìm hiểu về các hàm hữu dụng và cách sử dụng của chúng. Có một số hàm hay dùng như là `size()` (kiểm tra độ dài của dãy số/ma trận), `norm()` (độ lớn của vector/ma trận).



Tệp tin của MATLAB

Tổng quát

- Với công việc lớn hơn thì ta muốn dùng lại một số câu lệnh hoặc hàm tự viết
- Tệp tin của MATLAB có kết thúc là ".m"
- Có hai loại tệp chính: script-file và function-file
- Chạy tệp tin bằng việc gõ trong Command Window tên của tệp tin (không cần ".m")

Script-file chẳng qua chỉ là một file lưu trữ các câu lệnh. Thay vì phải viết lại, ví dụ như 10 câu lệnh, ta có thể lưu trữ chúng vào một tệp script-file và chỉ cần chạy lại như trên thông qua cửa sổ Command Window.



Sáng tạo ra hàm mới

Function-file giúp chúng ta tự viết nên một hàm mới, ví dụ như ta muốn tạo ra một hàm $f(x)$ xuất ra hai chữ số cuối cùng của x .

Cấu tạo của một Function-file

- Bắt đầu bằng `function`:
`function [out1, out2, ...] = function_name(in1, in2, ...)`
trong đó `out1`, `out2` là output parameters (biến xuất), `in1`, `in2` là input parameters (biến nhập), có thể thay đổi tùy sở thích
- Tiếp theo là những câu lệnh nhằm tính toán các biến
- Nếu muốn ghi chú (không phải câu lệnh) để người khác đọc dễ hiểu hơn thì viết sau dấu `"%"`
- Kết thúc bằng `end`



Ví dụ

Ta viết thử hàm $f(x)$ xuất ra hai chữ số cuối cùng của x . Trong cửa sổ Editor, điền các dòng sau:

```
function last = haisocuoix(x)
    last = mod(x,100); % lay so du cua x khi chia cho 100
end
```

Lưu file với cái tên "haisocuoix.m." Rồi chạy thử trong Command Window `haisocuoix(2016)` sẽ nhận được kết quả là 16.

Các vòng lặp thông dụng

Cũng như các ngôn ngữ lập trình khác, MATLAB hỗ trợ `for`, `if`, `while`. Cách sử dụng có thể tham khảo ở `http`:

`//www.mathworks.com/help/matlab/matlab_prog/loop-control-statements.html`

Tài liệu

Tài liệu chi tiết hơn có thể được đọc qua những links sau:

<http://www.math.mtu.edu/~msgocken/intro/intro.html>

<https://www.mccormick.northwestern.edu/documents/students/undergraduate/introduction-to-matlab.pdf>

<http://www.maths.dundee.ac.uk/software/MatlabNotes.pdf>

Các bài tập cụ thể có thể được tìm thấy ở:

<http://www.facstaff.bucknell.edu/maneval/help211/exercises.html>

<http://www.eng.ox.ac.uk/~labejp/Seminar/Matlab/Exercises.pdf>

<https://matlabacademy.mathworks.com/> (đòi hỏi tạo account)

Chúc các bạn học tập vui vẻ!

