Contents

[Getting Started – Bắt Đầu: 1](#_Toc159647752)

[Live Script: 2](#_Toc159647753)

[Data: 2](#_Toc159647754)

[Curve Fitting Tool: 3](#_Toc159647755)

[Command Window: 4](#_Toc159647756)

[Workspace: 5](#_Toc159647757)

[File – Tệp: 5](#_Toc159647758)

[Operator – Toán Tử: 6](#_Toc159647759)

[Function – Hàm: 7](#_Toc159647760)

[Condition – Điều Kiện: 8](#_Toc159647761)

[Loop – Lặp: 9](#_Toc159647762)

[Data Type – Kiểu Dữ Liệu: 9](#_Toc159647763)

[Matrix – Ma Trận: 9](#_Toc159647764)

[Tensor: 11](#_Toc159647765)

[Cell: 13](#_Toc159647766)

[Symbolic Expression: 13](#_Toc159647767)

[Plot: 19](#_Toc159647768)

[Simulink: 22](#_Toc159647769)

Getting Started – Bắt Đầu:

1. Cài Đặt Phần Mềm Matlab?

* Vào Link
* <https://phanmem123.com/download-matlab-2019/>
* Cuộn xuống Click “Download” màu đỏ bên dưới “MATLAB R2019b” + Click “Download” + chờ nó tải lâu vãi đái + giải nén File vừa tải với mật khẩu “Phanmem123.com” được 1 thư mục + vào thư mục này thấy chứa 1 File ISO và 1 File nén + giải nén File ISO + chạy File “setup” + Tick “Use a File Installation Key” + chọn “Next” + Tick “Yes” + chọn “Next” + Tick “I have the File Installation Key for my license:” + nhập “09806-07443-53955-64350-21751-41297” vào ô + chọn “Next” + tiến hành cài đặt như thường với các lựa chọn mặc định + chờ nó cài đặt xong + đóng cửa sổ + giải nén File nén kế File ISO lúc đầu với mật khẩu “Phanmem123.com” được thư mục, tạm gọi là A + chạy Matlab + Tick “Activate manually without the Internet” + chọn “Next” + Tick “Enter the full path …” + Click “Browse…” + chọn đường dẫn đến File “license\_standalone” nằm trong A + Click “Next” + Click “Finish” + Copy thư mục “bin” nằm trong thư mục “R2019b” trong A + Paste nó vào trong thư mục “C:\Program Files\Polyspace\R2019b” để thay thế cái cũ
* Khi này, trong thư mục “C:\Program Files\Polyspace\R2019b\bin” sẽ có File “matlab”, mở lên chạy

1. Cách Hoạt Động Của Phần Mềm Matlab?

* Tựa tựa VS Code, nó chỉ đơn thuần là 1 IDE

1. Cách Chọn Thư Mục Làm Việc Hiện Tại?

* Tại thanh đường dẫn, Click nút “Browse for folder” + chọn thư mục mong muốn

1. Phần Mở Rộng File Matlab?

* “.m”

1. Cấu Trúc Các Cửa Sổ?

* Giống Blender
* Để đóng 1 cửa sổ, Click nút “X” góc phải trên, khi đó cửa sổ bên cạnh sẽ mở rộng để lấp chỗ trống

1. Cửa Sổ Workspace?

* Nơi hiện thị các biến đã lưu
* Giống cách lưu biến trong Google Colab

1. Cửa Sổ Current Folder?

* Hiển thị các File và thư mục bên trong thư mục làm việc hiện tại
* Để thay đổi thư mục làm việc hiện tại mặc định khi mở Matlab
* Phải chuột vào biểu tượng Matlab ngoài Desktop + chọn “Properties” + vào Tab “Shortcut” + tại thuộc tính “Start in” + điền đường dẫn đến thư mục mong muốn + Click “OK”

1. Cửa Sổ Command Window?

* Giống Interactive Mode trong Python

1. Cách Chỉnh Sửa 1 File?

* Double Click vào File trong thư mục làm việc hiện tại + cửa sổ Editor hiện bên trên cửa sổ Workspace + tiến hành chỉnh sửa trong cửa sổ Editor + lưu + đóng cửa sổ

1. Cách Chạy 1 File?

* Cách 1
* Đầu tiên, thư mục làm việc hiện tại phải chứa File muốn chạy
* Tiếp theo, gõ tên của File vào cửa sổ Command Window, không có phần mở rộng + nhấn “Enter”
* Cách 2
* Mở File trong cửa sổ Editor + nhấn “F5” hoặc “Ctrl” + “Enter”, cách này y chang cách 1, cũng hiển thị lên Command Window, chỉ có điều nó sẽ lưu File rồi mới chạy

1. Tạo 1 File Matlab?

* Phải chuột vào cửa sổ Current Folder + chọn “New” + chọn “Script”

1. Chia File Thành Các Khối Code Để Chạy Riêng Lẻ?

* Bản chất giống bạn chạy từng khối Code riêng lẻ trong Google Colab
* Để tạo vạch ngăn cách giữa các khối Code trong 1 File, thêm kí tự “%%” vào đầu dòng ngăn cách, các kí tự tiếp theo cùng dòng coi như bị Comment
* Đặt con trỏ văn bản vào khối Code nào thì khi bạn nhấn nút chạy chỉ có khối Code đó chạy

1. Matlab Tìm Lệnh Ở Đâu Để Chạy?

* Các lệnh Matlab bản chất là 1 File Matlab, và bạn có thể gọi đến 1 File Matlab bất kì và chạy nó như 1 lệnh, nên bạn có thể tải các File lệnh Matlab trên mạng về rồi chạy nó
* Để xem danh sách các đường dẫn chứa các lệnh Matlab
* Vào Tab “HOME” + tại mục “ENVIRONMENT” + Click biểu tượng “Set Path” + khi này cửa sổ đường dẫn sẽ hiện lên liệt kê tất cả đường dẫn + thêm xóa đường dẫn rồi đóng lại
* Nó sẽ tìm lệnh trong thư mục làm việc hiện tại đầu tiên

Live Script:

1. Phần Mở Rộng File Live Script?

* “.mlx”

1. Tạo 1 File Live Script?

* Phải chuột vào cửa sổ Current Folder + chọn “New” + chọn “Live Script”
* Bản chất File Live Script y chang File Matlab thông thường, chỉ có điều thay vì dùng cửa sổ Editor nó dùng Live Editor
* Đồng thời, thay vì in kết quả ra Command Window, nó in ra nửa bên phải cửa sổ Live Editor
* Kết quả in ra sẽ được định dạng cực đẹp và dễ đọc, đặc biệt với Symbolic Expression
* Hơn nữa, mỗi lần chạy File là nó tự động xóa những gì đã in trước đó

1. Bật Tắt Chức Năng Gợi Ý Từ?

* Để tắt
* Vào Tab “HOME” + tại mục “ENVIRONMENT” + Click biểu tượng “Preferences” + vào mục “MATLAB” + vào mục “Keyboard” + bên phải tại mục “Suggestions and completions” + tại mục “Live Editor and App Designer” + chọn “Show on tab” + Click “OK”
* Để bật thì thay vì chọn “Show on tab” thì chọn “Show automatically”

Data:

1. Phần Mở Rộng File Dữ Liệu?

* “.mat”

1. Tạo 1 File Dữ Liệu?

* Để lưu tất cả các biến trong Workspace vào 1 File nhập lệnh sau, nếu File chưa tồn tại, tạo File mới, nếu đã tồn tại thì xóa dữ liệu cũ để ghi đè lên

save <Đường Dẫn Tới File Không Có Phần Mở Rộng>

* <Đường Dẫn Tới File Không Có Phần Mở Rộng> giống trong Batch
* Như trên, nhưng chỉ lưu 1 số biến trong Workspace

save <Đường Dẫn Tới File Không Có Phần Mở Rộng> <Các Biến>

* Ví dụ

save bob\foo a b c

* Để thêm 1 biến vào 1 File, nếu File đã có biến này thì ghi đè lên, nếu chưa có File thì báo lỗi

save(

<Đường Dẫn Tới File Không Có Phần Mở Rộng>,

<Tên Các Biến Dưới Dạng String>,

"-append"

)

* ví dụ

save("bob\foo", "d", "e", "-append")

* Để tải tất cả biến trong 1 File dữ liệu vào Workspace, nếu đã có biến thì ghi đè

load <Đường Dẫn Tới File Không Có Phần Mở Rộng>

* Như trên, nhưng chỉ tải 1 số biến

load <Đường Dẫn Tới File Không Có Phần Mở Rộng> <Các Biến>

Curve Fitting Tool:

1. Phần Mở Rộng File Curve Fitting?

* “.sfit”

1. Cấu Trúc?

* 1 File Curve Fitting bao gồm nhiều Tab ứng với nhiều đồ thị rời rạc

1. Cửa Sổ Curve Fitting?

* Để mở cửa sổ Curve Fitting
* Vào Tab “APPS” + tại mục “APPS” + Click biểu tượng “Curve Fitting”
* Để mở 1 File Curve Fitting
* Vào Tab “File” + chọn “Load Session…” + chọn File Curve Fitting mong muốn + Click “Open”
* Để lưu 1 File Curve Fitting
* Vào Tab “File” + chọn “Save Session As…” + chọn nơi lưu File + nhập tên File + Click “Save”
* Để tạo 1 Tab ứng với đồ thị rời rạc mới
* Vào Tab “Fit” + chọn “New Fit”
* Để đổi tên 1 Tab
* Vào Tab đó + ghi tên mới vào thuộc tính “Fit name” góc trái trên + nhấn “Enter”

1. Tìm Đường Cong Khít Nhất Với 1 Đồ Thị Rời Rạc 2D?

* Vào Tab muốn Plot đồ thị rời rạc và đường cong + tại thuộc tính “X data” chọn biến trong Workspace làm trục hoành + tại thuộc tính “Y data” chọn biến trong Workspace làm trục tung + tại ô trên cùng ở giữa chọn kiểu hàm để Fit
* Tại khu vực “Results” sẽ hiển thị các tham số khít nhất

Command Window:

1. Xóa Toàn Bộ Những Gì Trên Command Window?

clc

1. Cách Ẩn Không In Ra 1 Câu Lệnh In?

* Mỗi khi có 1 câu lệnh gán, hoặc in thông tin, trên Command Window lại in ra biến được gán, thông tin, …, để tránh điều này, thêm chấm phẩy vào sau câu lệnh gán
* Ví dụ

foo = bar + 2;

whos;

1. Cách Viết Nhiều Câu Lệnh Trên 1 Dòng?

* Thêm dấu phẩy giữa các câu lệnh, có thể thay bằng dấu chấm phẩy nếu không muốn in thông tin
* Ví dụ

clearvars, foo = 4 + 5; whos, bar = bar + 10

1. Comment 1 Dòng Code?

* Đặt dấu “%“ ở đầu
* Ví dụ

a = 1 + 5 %+ 7

* Ta có

|  |  |
| --- | --- |
| a | 6 |

* Để đặt dấu “%“ ở đầu tất cả các dòng Code được chọn
* Nhấn “Ctrl” + “R”
* Để loại bỏ dấu “%“ ở đầu tất cả các dòng Code được chọn
* Nhấn “Ctrl” + “T”

1. Comment 1 Khối Code?

* Đặt đấu “%{“ vào đầu dòng Code ngay trước khối Code và “%}” vào đầu dòng Code ngay sau khối Code
* Ví dụ

%{

foo = bar + 4

bar = foo

foobar = barfoo

%}

1. In Ra Cách Sử Dụng Hàm?

help <Tên Hàm>

* Ví dụ

help max

1. Cách Viết 1 Lệnh Trên Nhiều Dòng?

* Giống như trong Python, thay vì dùng dấu “\” thì dùng “…”

1. Mở Tài Liệu Trên Trình Duyệt Về 1 Hàm?

doc <Tên Hàm>

1. Chỉ In Giá Trị Ra Và Không In Tên Biến Rồi Xuống Dòng?

disp(<Giá Trị>)

* Ví dụ

disp("Hello Mom")

1. Chỉ In Ra Chuỗi Kí Tự Và Không Xuống Dòng?

fprintf(<String>)

1. In Ra Chuỗi Kí Tự Và Không Xuống Dòng Và Tạo Màu Các Kiểu Cho Nó?

cprintf(<Định Dạng>, <String>)

* Bảng định dạng

|  |  |
| --- | --- |
| "hyper" | Gạch chân và có màu xanh dương |

1. Đo Thời Gian Chạy Code?

* Chạy lệnh sau để bắt đầu đếm giây, hoặc đặt lại đồng hồ

tic

* Khi câu lệnh trên chạy, thì sẽ có 1 đồng hồ chạy ngầm trong phần mềm Matlab, bắt đầu từ 0 giây
* Để in giá trị từ đồng hồ tại thời điểm hiện tại

toc

1. Hỏi Người Dùng Nhập Input?

<Giá Trị Nhập> = input(<Câu Hỏi>)

* Khi đến dòng lệnh này, tại Command Window sẽ in ra <Câu Hỏi>, khi này bạn cần nhập 1 biểu thức sử dụng các biến tại Workspace, nó sẽ tự tính ra và trả về giá trị cho bạn, và lưu trong <Giá Trị Nhập>
* Ví dụ

foo = input("bar?")

* Khi này nếu bạn nhập x + y, mà x = 2 và y = 5, thì foo = 7
* Nếu bạn nhập 1 biểu thức không thể tính được hay nhập tào lao, thì <Câu Hỏi> sẽ lại được in ra và bạn phải nhập lại
* Nếu không muốn nó tính giá trị biểu thức nhập vào, mà chỉ muốn lưu String bạn nhập vào, thì viết

<String> = input(<Câu Hỏi>, 's')

1. Dừng Chương Trình Khi Nó Lặp Vô Tận?

* “Ctrl” + “C”

1. Tạm Dừng Chương Trình Trong Vài Giây?

pause(<Số Giây>)

Workspace:

1. Xóa Toàn Bộ Biến Hiện Tại Trong Workspace?

clearvars

1. Xóa Toàn Bộ Biến Hiện Tại Trong Workspace Ngoại Trừ 1 Số Biến?

clearvars -except <Các Biến Không Bị Xóa>

* Ví dụ

clearvars -except foo bar

1. In Ra Thông Tin Tất Cả Các Biến Hiện Tại Trong Workspace?

whos

File – Tệp:

1. Cách Mở 1 File Văn Bản?

* Khi chương trình Matlab mở 1 File nào đó, nó sẽ chạy ngầm, được đánh Index bắt đầu = 3, mở File tiếp theo, thì Index File này = 4, …, khi bạn đóng 1 File, thì do Index bị thiếu nên khi mở lại 1 File khác hoặc File cũ, thì Index File này sẽ Fill vào những chỗ bị thiếu, theo chiều tăng dần Index
* Để mở 1 File và trả về Index của nó

<File Index> = fopen(<Đường Dẫn Đến File Có Phần Mở Rộng>)

* <Đường Dẫn Đến File Có Phần Mở Rộng> giống trong Batch
* Ví dụ

foo = fopen('..\bar.txt')

* Để đóng 1 File

fclose(<File Index>)

* Để đóng toàn bộ File

fclose('all')

Operator – Toán Tử:

1. Lũy Thừa?

^

* Nếu dùng với ma trận thì = nhân ma trận nhiều lần hoặc lấy nghịch đảo nhiều lần
* Ví dụ

foo^-1

1. Số Phức?

<Thực> + <Ảo>i

* Ví dụ

2 + 4i

1. Trả Về Căn Bậc 2 Của 1 Giá Trị?

sqrt(<Giá Trị>)

1. Index?

* Bắt đầu từ 1 chứ đéo phải 0

1. Nhân Ma Trận?

\*

1. Tích Hadamard?

.\*

* Đặt dấu chấm đằng trước các toán tử để thực hiện phép tính trên từng phần tử tương ứng của 2 ma trận

1. Không Bằng?

~=

1. Toán Tử NOT?

~

1. Cách Gán 1 Giá Trị Cho Nhiều Biến 1 Lúc?

* Để gán từng giá trị khác nhau cho từng biến trên 1 dòng

<Các Biến> = deal(<Các Giá Trị>)

* Ví dụ

[a b c] = deal(1, 2, 3)

* Ta có

|  |  |
| --- | --- |
| a | 1 |
| b | 2 |
| c | 3 |

* Để gán 1 giá trị cho tất cả các biến

<Các Biến> = deal(<Giá Trị)

* Ví dụ

[a b c] = deal(5)

* Ta có

|  |  |
| --- | --- |
| a | 5 |
| b | 5 |
| c | 5 |

1. Chia Lấy Dư?

<Kết Quả> = mod(<Tensor>, <Số Chia>)

1. Cộng Char Với Char?

* Kết quả sẽ là tổng mã ASCII của 2 thằng, nghĩa là kết quả sẽ có kiểu Double

1. Cộng Char Với String Hoặc String Với String?

* Kết quả sẽ là 2 thằng nối lại với nhau tạo thành 1 String duy nhất

Function – Hàm:

1. Hàm Và Function Handle?

* 1 hàm phải khai báo ở cuối File
* Để sử dụng hàm, 1 là gọi trực tiếp tên hàm, 2 là thông qua Function Handle, tức là 1 biến tham chiếu đến hàm đó

1. Tạo Trực Tiếp 1 Function Handle?

* Giống hàm Lambda trong Python

<Function Handle> = @(<Các Tham Số>) <Biểu Thức>

* Ví dụ

foo = @(x, y) x + y

bar = foo(1, 2)

* Ta có

|  |  |
| --- | --- |
| bar | 3 |

1. Cấu Trúc 1 Hàm?

function <Các Biến Chứa Giá Trị Trả Về> = <Hàm>(<Các Tham Số>)

<Làm Gì Đó>

end

* Ví dụ

fool, boy = far(2, 3)

function [foo bar] = far(boo, fap)

foo = boo + fap

bar = boo .\* fap

end

* Ta có

|  |  |
| --- | --- |
| fool | 5 |
| boy | 6 |

* Nếu bạn viết kiểu này

fool = far(2, 3)

* Thì hàm sẽ trả về giá trị đầu tiên

1. Tạo 1 Function Handle Từ 1 Hàm Có Sẵn?

<Function Handle> = @<Hàm>

* Ví dụ

foo = @sin

bar = foo(pi/2)

* Ta có

|  |  |
| --- | --- |
| bar | 1 |

1. Tìm Điểm Cho Giá Trị Nhỏ Nhất Của 1 Hàm?

<X Min> = fminbnd(<Function Handle 1 Biến>, <Cận Dưới>, <Cận Trên>)

* Cơ chế là cho biến chạy từ <Cận Dưới> tới <Cận Trên> rồi tính từng giá trị hàm trả về để tìm <X Min> là giá trị biến cho giá trị nhỏ nhất
* Ví dụ

foo = fminbnd(@sin, 1, 2)

1. Tính Tích Phân Của 1 Hàm?

<Tích Phân> = integral(<Function Handle>, <Cận Dưới>, <Cận Trên>)

* <Function Handle> phải là hàm 1 biến
* Ví dụ

foo = integral(@(x) x.^2, 3, 6)

* Ta có

|  |  |
| --- | --- |
| foo | 63 |

1. Giải Hệ Phương Trình Phi Tuyến?

<Solution> = fsolve(<Function Handle>, <Giá Trị Khởi Đầu>)

* <Function Handle> phải là hàm 1 biến, trả về 1 Tensor
* <Giá Trị Khởi Đầu> sẽ được Pass vào <Function Handle>, làm vị trí xuất phát để tìm nghiệm
* <Solution> là 1 Tensor mà khi Pass vào <Function Handle> sẽ trả về Tensor Full 0
* Ví dụ

f = @(x) [x(1) + x(3), x(3) + x(1)]

nghiem = fsolve(f, randi(3, 4))

1. Pass Tham Số Từ Command Window Vào Chương Trình?

* Để làm điều này, bạn cần một hàm giống hàm main trong C++
* Hàm này phải có tên giống với tên File chứa nó, trong hàm này có thể sử dụng các hàm khác
* Các tham số Pass vào sẽ được chuyển thành chuỗi Char
* Để Pass tham số cho chương trình, dùng cú pháp

<Tên Chương Trình> <Danh Sách Tham Số>

* <Danh Sách Tham Số> cách nhau bởi khoảng trắng
* Ví dụ, trong File foo.m, ta có viết hàm sau

function foo(a, b)

<Làm Gì Đó>

end

* Thì khi này tại Command Window, ta có thể nhập lệnh

foo 123 abc

* Khi này tham số a sẽ = '123' và b = 'abc'

Condition – Điều Kiện:

1. Cấu Trúc Câu Điều Kiện?

if <Biểu Thức>

<Làm Gì Đó>

elseif <Biểu Thức>

<Làm Gì Đó>

else

<Làm Gì Đó>

end

* Ví dụ

if 2 > 1

foo = bar + 1

else

bar = foo

end

Loop – Lặp:

1. For Loop?

for <Biến Đếm> = <Tensor>

<Làm Gì Đó>

end

* Vòng lặp này sẽ lặp qua toàn bộ cột của <Tensor> và trả về cột đó cho

<Biến Đếm>, đọc theo thứ tự tăng dần Index, từ chiều thứ 2 tới chiều cuối cùng, ví dụ (1:3, 1, 1), (1:3, 2, 1), (1:3, 1, 2), …, ở ví dụ này <Biến Đếm> có Shape là (3, 1)

1. While Loop?

while <Tensor Kiểm Tra>

<Làm Gì Đó>

end

* Nếu <Tensor Kiểm Tra> khi quy đổi về Boolean mà toàn 1 thì <Làm Gì Đó> được thực thi và tiếp tục vòng lặp, nếu không thì kết thúc vòng lặp
* Ví dụ

Data Type – Kiểu Dữ Liệu:

1. Có Phải Tất Cả Các Biến Đều Là Ma Trận 2 Chiều?

* Đúng, đại lượng vô hướng là ma trận 1 x 1, …, các biến này có thể mở rộng lên Tensor bậc 3, 4, …

1. Logical?

* Là kiểu Boolean

1. String Và Char?

* String phải được đặt trong cặp dấu nháy kép
* Char phải được đặt trong cặp dấu nháy đơn
* Ví dụ
* "abcd" là 1 String Tensor với Shape (1, 1)
* 'abcd' là 1 Char Tensor với Shape (1, 4)
* Khi ta viết như này, thì các Char sẽ được hợp lại

[<Chuỗi Char 1>, <Chuỗi Char 2>, …]

* Ví dụ

['abc', 'd', 'ef']

* Tương đương với việc viết

'abcdef'

1. Trả Về Kiểu Dữ Liệu Của 1 Biến Hoặc 1 Phần Tử Trong Ma Trận?

<Kiểu Dữ Liệu> = class(<Biến>)

* Ví dụ

foo = class(bar)

* Ta có

|  |  |
| --- | --- |
| foo | "double" |

1. Chuyển String Thành Số?

* Để chuyển 1 Tensor, Tensor này chỉ chứa 1 phần tử duy nhất là 1 String, hoặc chuyển 1 Tensor với Shape (1, <Số Kí Tự>) chứa các phần tử thuộc kiểu Char, thành Tensor với Shape (1, 1), chứa phần tử duy nhất là 1 số ứng với String này hoặc hợp của các Char này

<Số> = str2num(<String>)

* Ví dụ

foo = str2num("123")

bar = str2num(['1', '24', '357'])

* Ta có

|  |  |
| --- | --- |
| foo | [[123]] |
| bar | [[124357]] |

1. Chuyển String Thành Char?

* Để chuyển 1 Tensor, Tensor này chỉ chứa 1 phần tử duy nhất là 1 String, thành Tensor với Shape (1, <Số Kí Tự>) chứa các phần tử thuộc kiểu Char

<Char> = char(<String>)

1. Chuyển Char Thành Double?

* Để chuyển 1 Tensor chứa các phần tử là Char sang Tensor chứa mã ASCII tương ứng

<Double Tensor> = double(<Char Tensor>)

1. Chuyển Double Thành Char?

* Để chuyển 1 Tensor chứa các phần tử là Double sang Tensor chứa các phần tử là Char, các phần tử Double đóng vai trò là mã ASCII của Char tương ứng, nếu phần tử Double này không nguyên, thì lấy phần nguyên của nó

<Char Tensor> = char(<Double Tensor>)

* Để chuyển 1 Tensor chứa các phần tử là Double sang Tensor chứa các phần tử là Char, là nối của các phần tử Double ở chiều cuối cùng, nghĩa là ví dụ các phần tử Double là 1, 45, 67 thì sẽ thành chuỗi Char '1 45 67', mỗi số cách nhau bởi 2 dấu cách

<Char Tensor> = num2str(<Double Tensor>)

1. Chuyển Char Từ Số Nhị Phân Sang Nguyên?

* Để chuyển 1 Tensor, Tensor này chỉ chứa các phần từ là '0' hoặc '1', thành số nguyên, nghĩa là các phần tử ở chiều cuối cùng của Tensor này hợp lại thành chuỗi nhị phân rồi chuyển thành số nguyên tương ứng

<Double Tensor> = bin2dec(<Char Tensor>)

1. Chuyển String Thành Mảng Các Từ?

* Để chuyển 1 Tensor, Tensor này chỉ chứa 1 phần tử duy nhất là 1 String, thành Tensor với Shape (<Số Từ>, 1), chứa các phần tử là các từ cách nhau bởi <Kí Tự Phân Chia> trong String này

<Các Từ> = split(<String>, <Kí Tự Phân Chia>)

* Mặc định

|  |  |
| --- | --- |
| <Kí Tự Phân Chia> | ' ' |

* Cơ chế như sau, cắt String con ra khỏi <String>, bắt đầu từ kí tự đầu tiên của <String>, đến hết <Kí Tự Phân Chia> đầu tiên, bỏ <Kí Tự Phân Chia> ra khỏi String con, ta được 1 từ, tiếp tục lặp lại với phần còn lại của <String>
* Ví dụ

foo = split(" toi la con di ")

bar = split(" aaanh muaaon a dit aa b", 'a')

* Ta có

|  |  |  |  |
| --- | --- | --- | --- |
| foo | [[""],  ["toi"],  ["la"],  ["con"],  ["di"],  [""]] | bar | [[" "],  [""],  [""],  ["nh mu"],  [""],  ["on "],  [" dit "],  [""],  [" b"] |

Matrix – Ma Trận:

1. Cách Gán Giá Trị Là 1 Ma Trận Cho 1 Biến?

<Ma trận> = [<Hàng 1> ; <Hàng 2> ; … ; <Hàng Cuối>]

* Các giá trị trong hàng ngăn cách nhau bởi dấu cách
* Ví dụ

M = [1 2 3 ; 4 5 6 ; 7 8 9]

* Nếu 1 phần tử là String thì toàn bộ phần tử sẽ chuyển sang String

1. Trả Về Bản Copy 1 Hàng Của 1 Ma Trận?

<Hàng> = <Ma Trận>(<Index Hàng>, :)

* <Hàng> là ma trận có 1 hàng

1. Trả Về Bản Copy Cột Của 1 Ma Trận?

<Cột> = <Ma Trận>(:, <Index Cột>)

* <Cột> là ma trận có 1 cột

1. Trả Về Bản Copy 1 Phần Tử Của 1 Ma Trận?

<Ma Trận>(<Index Hàng>, <Index Cột>)

* Để truy cập sử dụng <Số Thứ Tự> của phần tử, <Số Thứ Tự> được đánh từ trái qua phải từ trên xuống dưới, bắt đầu từ 1

<Ma Trận>(<Số Thứ Tự>)

* Để trả về phần tử có <Số Thứ Tự> lớn nhất, hay cuối cùng

<Ma Trận>(end)

1. Tạo 1 Ma Trận Với Giá Trị Ngẫu Nhiên Từ Phân Phối Đều Liên Tục Trong Khoảng Từ 0 Đến 1?

<Ma Trận> = rand(<Số Dòng>, <Số Cột>)

1. Trả Về Chuyển Vị Của Ma Trận?

<Ma Trận>.’

1. Trả Về Chuyển Vị Liên Hợp Phức Của Ma Trận?

<Ma Trận>’

1. Trả Về Dạng Hàng Bậc Thang Rút Gọn Của Ma Trận?

rref(<Ma Trận>)

1. Trả Về Eigen Vector Và Eigen Value Của Ma Trận?

[<Eigen Vector>, <Eigen Value>] = eig(<Ma Trận>)

* <Eigen Vector> là 1 ma trận mà các cột của nó là Eigen Vector của <Ma Trận>, các Eigen Vector này có độ dài = 1
* <Eigen Value> là 1 ma trận đường chéo mà giá trị trên đường chéo chính của nó là Eigen Value tương ứng với mỗi Eigen Vector

1. Tạo Ma Trận Chỉ Có 1 Hàng Và Là Cấp Số Cộng Với Công Sai = 1?

<Ma Trận> = <Start>:<End>

* Phần tử đầu tiên = <Start>, tăng dần sao cho không vượt quá <End>
* Ví dụ

foo = 4.3:9.1

* Ta có

|  |  |
| --- | --- |
| foo | [[4.3, 5.3, 6.3, 7.3, 8.3]] |

* Để dùng công sai khác

<Ma Trận> = <Start>:<Công Sai>:<End>

1. Tạo Ma Trận Chỉ Có 1 Hàng Và Là Cấp Số Cộng Cho Trước Số Phần Tử?

<Ma Trận> = linspace(<Start>, <End>, <Số Phần Tử>)

* Mặc định

|  |  |
| --- | --- |
| <Số Phần Tử> | 100 |

* Phần tử đầu tiên = <Start>, phần tử cuối = <Kết Thúc>, các phần tử cách đều nhau
* Ví dụ

foo = linspace(4, 9, 11)

* Ta có

|  |  |
| --- | --- |
| foo | [[4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9]] |

Tensor:

1. Tensor Trong Matlab?

* Ví dụ Tensor 3D, tưởng tượng xấp ảnh như trong Torch, nhưng mặt ảnh úp vào tường bên trái
* Mỗi ảnh là 1 ma trận riêng biệt
* Mỗi khi 1 Tensor được tạo, Shape của nó sẽ được rút gọn từ phải sang trái, quy tắc chiều cuối cùng = 1 thì rút gọn, không rút nữa nếu đạt đến Tensor bậc 2
* Ví dụ ban đầu Tensor có Shape là (4, 5, 1, 6, 1, 1) thì sẽ thành (4, 5, 1, 6)

1. Tạo 1 Tensor Với Giá Trị Ngẫu Nhiên Từ Phân Phối Đều Liên Tục Trong Khoảng Từ 0 Đến 1?

<Tensor> = rand(<Shape>)

* Ví dụ

foo = rand(4, 5, 6, 7)

1. Tạo 1 Tensor Với Giá Trị Nguyên Ngẫu Nhiên Từ Phân Phối Đều Trong Khoảng Từ 1 Đến Giá Trị Nào Đó?

<Tensor> = randi(<Max>, <Shape>)

* <Tensor> sẽ có giá trị nguyên từ 1 đến <Max>
* Ví dụ

foo = randi(10, 3, 4, 5, 6)

1. Tạo 1 Tensor Với Giá Trị Ngẫu Nhiên Từ Phân Phối Chuẩn?

<Tensor> = normrnd(<Mean>, <Độ Lệch Chuẩn>, <Shape>)

* Ví dụ

foo = normrnd(1, 2, 3, 4, 5)

1. Trả Về Giá Trị Lớn Nhất Trong Tensor?

<Max> = max(<Tensor>)

* Tính <Max> theo chiều đầu tiên có kích thước > 1
* Ví dụ nếu <Tensor> Shape là (1, 1, 4, 5) thì nó nó tính <Max> theo chiều thứ 3
* Ví dụ

foo = max(bar)

* Ta có

|  |  |
| --- | --- |
| bar | [[4, 5, 7],  [8, 9, 6]] |
| foo | [[8, 9, 7]] |

* Để trả về thêm <Index> của <Max>

[<Max>, <Index Tensor>] = max(<Tensor>)

* <Max> và <Index Tensor> cùng Shape, mỗi phần tử trong <Index Tensor> là <Index> của phần tử tương ứng trong <Max> theo chiều tính <Max>

1. Trả Về Chiều Lớn Nhất Của 1 Tensor?

length(<Tensor>)

* Ví dụ

foo = length(bar)

* Ta có

|  |  |
| --- | --- |
| bar Shape | (4, 9, 13, 6) |
| foo | 13 |

1. Trả Về Chiều Dài 1 Vector?

<Chiều Dài> = norm(<Vector>)

* <Vector> phải là Tensor sao cho có nhiều nhất 1 chiều có kích thước > 1

1. Trả Về Tensor Bị Lặp?

<Tensor Lặp> = repmat(<Tensor>, <Số Lần Lặp Từng Chiều>)

* Ví dụ

foo = repmat(bar, 2, 3)

* bar có Shape là (4, 5) thì foo có Shape là (8, 15)

1. Tích Vô Hướng?

<Tích Vô Hướng> = dot(<A>, <B>)

* Nếu <A> và <B> chỉ gồm 1 cột hoặc hàng và có số phần tử = nhau, thì

<Tích Vô Hướng> có Shape là (1, 1)

* Nếu <A> và <B> có chiều thứ nhất và thứ 2 đều > 1 và cùng Shape, dùng cú pháp

<Tích Vô Hướng> = dot(<A>, <B>, <Chiều Tính Tích>)

* Mặc định

|  |  |
| --- | --- |
| <Chiều Tính Tích> | 1 |

* Ví dụ

foo = dot(A, B, 2)

* Ta có

|  |  |
| --- | --- |
|  | Shape |
| A | (3, 4, 5) |
| B | (3, 4, 5) |
| foo | (3, 1, 5) |

1. Tích Có Hướng?

<Tích Có Hướng> = cross(<A>, <B>)

* Nếu <A> và <B> đều là 1 cột gồm 3 phần tử, thì <Tích Có Hướng> có Shape là (3, 1) còn nếu ít nhất 1 cái là hàng thì <Tích Có Hướng> có Shape là (1, 3)
* Nếu <A> và <B> có chiều thứ nhất và thứ 2 đều > 1 và cùng Shape, dùng cú pháp

<Tích Có Hướng> = cross(<A>, <B>, <Chiều Tính Tích>)

* Kích thước của <Chiều Tính Tích> phải = 3
* Tích có hướng này sử dụng liên hợp phức
* Mặc định

|  |  |
| --- | --- |
| <Chiều Tính Tích> | 1 |

* Ví dụ

foo = cross(A, B, 2)

* Ta có

|  |  |
| --- | --- |
|  | Shape |
| A | (4, 3, 5) |
| B | (4, 3, 5) |
| foo | (4, 3, 5) |

1. Trả Về Bản Copy 1 Phần Của Tensor?

<Tensor>(<Phần>)

* <Phần> tương tự như từ hàng thứ mấy tới hàng thứ mấy, cột thứ mấy đến cột thứ mấy, …, bắt đầu từ chiều đầu tiên, phải chỉ định tất cả các chiều
* end đại diện cho Index cuối cùng của chiều tương ứng
* Nếu lấy các hàng rời rạc thì liệt kê Index trong 1 mảng Double hoặc đánh dấu 1 ở hàng lấy và 0 ở hàng bỏ với mảng Logical
* Ví dụ 1, foo là Tensor với Shape (7, 4, 5, 6)

bar = foo([2, 4], 1, end-2:end, 2:end-1)

* bar sẽ có Shape (2, 1, 3, 4)
* Nếu không chỉ định tất cả các chiều, thì phần chỉ định cuối cùng, sẽ áp dụng cho toàn bộ các chiều còn lại, nghĩa là các chiều cuối sẽ bị hợp lại thành 1 chiều, thứ tự đánh Index sẽ từ chiều trước rồi mới tới chiều sau, end khi này đại diện cho phần tử cuối cùng của <Tensor>
* Tiếp tục ví dụ 1

alice = foo(1:5, [false false true false], 1:end-3)

* alice sẽ có Shape (5, 1, 27)

1. Đảo Chiều 1 Mảng?

* Cho 1 Tensor chỉ có nhiều nhất 1 chiều với kích thước > 1, để đảo chiều các phần tử ở chiều này, nghĩa là đọc từ phần tử cuối lên phần tử đầu

<Tensor Đảo> = flip(<Tensor>)

1. Check Xem Char Có Phải Dấu Cách?

<Logical Tensor> = isspace(<Char Tensor>)

* <Logical Tensor> = lấy mỗi phần tử trông <Char Tensor>, xem nó có phải dấu cách không, nếu phải thì True, không phải thì False

Cell:

1. Cell Là Cái Đéo Gì?

* Giống Tensor nhưng mỗi ô có thể lưu những giá trị thuộc kiểu dữ liệu khác nhau, kể cả Tensor, Symbol, String, … hay 1 Cell con
* Để tạo 1 Cell mặc định mà mỗi ô chứa Tensor với Shape (0, 0)

<Cell> = cell(<Shape>)

* Ví dụ

foo = cell(4, 5, 6)

* Để tạo 1 khởi tạo 1 Cell với giá trị do mình chỉ định
* <Cell> = {<Ma Trận Các Giá Trị>}
* Giống như khi khởi tạo giá trị cho Tensor
* Ví dụ

foo = {

1 2 3 4;

[4 7 8] 'a' 'ba' 7

}

Symbolic Expression:

1. Symbol?

* Mỗi Symbol bản chất là 1 Container, nó là Tensor mà mỗi ô là 1 biểu thức chữ của những đại lượng vô hướng
* Khi bạn in 1 Symbol, thì nó không in ra kết quả số mà là chữ

1. Tạo 1 Symbol 1 x 1?

<Symbol> = sym(<Kí Hiệu>)

* Ví dụ

foo = sym("x")

bar = 1/foo^2

* Ở đây, foo là Container 1 x 1, “x” là biểu thức dạng chữ chứa trong ô duy nhất của nó, và bar cũng là Container 1 x 1, “1 / x2” là biểu thức dạng chữ chứa trong ô duy nhất của nó
* Khi bạn Show bar ra, thì nó sẽ ra

* Để tạo 1 Symbol với Shape khác 1 x 1

<Symbol> = sym(<Kí Hiệu>, <Shape>)

* Khi này, mặc định mỗi ô trong <Symbol> sẽ chứa 1 kí tự có tên = <Kí Hiệu> + <Index>
* Ví dụ

foo = sym("M", [2 3 4])

* Thì ô đầu tiên ghi “M1\_1\_1”, ô thứ 2 ghi “M1\_1\_2”, …, ô cuối ghi “M2\_3\_4”
* Để thay thế các kí tự trong <Symbol> = 1 giá trị nào đó

<Kết Quả> = subs(<Symbol>, <Các Kí Tự Cũ>, <Các Giá Trị Mới>)

* Ví dụ

foo = subs(bar, [bob alice], [1 2])

1. Symbolic Function?

* Là 1 hàm nhận đối số là các kí tự và trả về Tensor mà các ô chứa biểu thức của các kí tự hoặc kí hiệu 1 hàm
* Để tạo 1 Symbolic Function

<Symbolic Function> = symfun(

<Biểu Thức Các Symbol>,

<Các Đối Số Symbol>

)

* <Các Đối Số Symbol> phải là các Symbol với Shape 2 chiều, chỉ gồm 1 dòng, khi này các Symbol sẽ được Concat với nhau
* Ví dụ

bar = sym("x", [1 2])

far = sym("y", [1 3])

fool = sym("z")

foo = symfun(bar \* bar.' + far(3) + fool, [bar far])

* Khi in foo ra màn hình

foo(x1, x2, y1, y2, y3) = x12 + x22 + y3 + z

* Ta có Pass giá trị cụ thể cho hàm, giá trị này có thể là Symbol, Tensor, … miễn là tính được giá trị, kết quả trả về là Symbol, lưu ý ở đây tính kiểu Bit Wise giữa các tham số
* Tiếp tục với ví dụ trên

alice1 = rand(4, 5, 6)

alice2 = rand(4, 5, 6)

…

bob = foo(alice1, alice2, alice3, alice4, alice5)

* bob là Symbol với Shape (4, 5, 6), = alice1.^2 + alice.^2 + alice5 + z

1. Tạo 1 Lúc Nhiều Symbol?

* Tạo nhiều Symbol 1 x 1

syms <Các Symbol>

* <Kí Hiệu> mặc định = tên <Symbol>
* Ví dụ

syms x y z foo bar

* Để tạo 1 lúc nhiều Symbol là Tensor với Shape y chang nhau

syms <Các Biến> <Shape>

* Ví dụ

syms x y z foo bar [3 4 5]

* Khi này các Symbol với tên “x1\_1\_1”, “x1\_1\_2”, … cũng sẽ được tạo, Shape 1 x 1 và chứa kí tự cùng tên
* Để tạo 1 Symbolic Function 1 x 1 mà trong ô duy nhất của nó chứa kí hiệu hàm, đồng thời tạo các Symbol đối số luôn, nghĩa là đây là hàm chưa biết biểu thức

syms <Symbolic Function>(<Các Đối Số>)

* Ví dụ

syms foo(bar, far)

* Khi in foo ra màn hình

foo(bar, far) = foo(bar, far)

* Cũng như trên, nhưng tạo Symbolic Function với Shape to hơn, các hàm trong ô cũng sẽ được tạo 1 Symbolic Function cho riêng nó

syms <Symbolic Function>(<Các Đối Số>) <Shape>

* Ví dụ

syms foo(bar, far) [3 4 5]

* Ô đầu tiên sẽ ghi “foo1\_1\_1(bar, far)”, ô thứ 2 ghi “foo1\_1\_2(bar, far)”, …

1. Tìm Giới Hạn Của 1 Symbolic Function Khi 1 Symbol Dần Tới 1 Giá Trị Nào Đó?

<Giới Hạn> = limit(<Biểu Thức>, <Symbol>, <Điểm>)

* <Biểu Thức> có thể là Symbol, Symbolic Expression hoặc Symbolic Function
* <Giới Hạn> là Symbol
* Ví dụ

foo = limit(bar + boo/boo, boo, 0)

* Khi in ra foo

foo = bar + 1

1. Đạo Hàm?

* Để tìm đạo hàm cấp N theo 1 Symbol cụ thể

<Đạo Hàm> = diff(<Biểu Thức>, <Symbol>, <N>)

* <Biểu Thức> có thể là Symbol, Symbolic Expression hoặc Symbolic Function
* <Đạo Hàm> là Symbol, nhưng sẽ là Symbolic Function nếu <Biểu Thức> cũng vậy
* Mặc định

|  |  |
| --- | --- |
| <N> | 1 |

* Ví dụ

foo = diff(X^3 + sin(X), X, 2)

* Khi in foo ra

foo = 6X – sin(X)

1. Giải Phương Trình?

<Các Nghiệm> = solve(<Biểu Thức>, <Symbol>)

* <Biểu Thức> có thể là Symbol, Symbolic Expression hoặc Symbolic Function
* <Các Nghiệm> là Symbol, là 1 ma trận chỉ có 1 cột, mỗi phần tử là 1 Symbolic Expression, sao cho khi <Symbol> = chúng thì <Biểu Thức> = 0
* Ví dụ

foo = solve(X^2 – 4, X)

* Ta có

|  |  |
| --- | --- |
| foo | [[2],  [–2]] |

* Giả sử ta muốn giải bất phương trình, thì đầu tiên Matlab sẽ tạo ra nghiệm với kí hiệu nào đó, sau đó thêm điều kiện giới hạn cho kí hiệu đó
* Ví dụ

foo = solve(X^2 > 4, X)

* foo là 1 Object, foo.X là Symbol chứa nghiệm kí hiệu, foo.conditions là Symbol chứa điều kiện

1. Thay Số Và Tính Giá Trị Cụ Thể 1 Symbolic Function?

<Kết Quả> = <Symbolic Function>(<Các Giá Trị>)

* <Kết Quả> là Symbol, là 1 biểu thức số cực chính xác không thể rút gọn hơn nữa
* Ví dụ

bar(x, y) = x + y

foo = bar(1, 2)

* Ta có

|  |  |
| --- | --- |
| foo | 3 |

1. Symbolic Equation?

* 1 Symbolic Equation có thể là Symbol, Symbolic Expression hoặc Symbolic Function, miễn là có toán tử so sánh bằng
* Ví dụ

foo = x^2 – 2 \* x + 1 == y + x

1. Giải Hệ Phương Trình Tuyến Tính?

* Để trả về ma trận bên trái dấu = và ma trận 1 cột bên phải dấu = của hệ phương trình

[<Ma Trận Trái> <Ma Trận Phải>] = equationsToMatrix(

<Các Symbolic Equation>, <Các Symbol>

)

* <Các Symbolic Equation> phải là các phương trình tuyến tính
* Thứ tự các biến được cho bởi <Các Symbol>
* Ví dụ

[foo bar] = equationsToMatrix([e1 e2 e3], [x y z])

* Ta có

|  |  |
| --- | --- |
| foo | [[1, 2, 3],  [5, 6, 7],  [9, 8, 7]] |
| bar | [[4],  [5],  [1]] |

* Để giải phương trình dạng AX = B, A, X, B là các ma trận, A, B đã biết, tìm X

<X> = linsolve(<A>, <B>)

1. Giải Phương Trình Vi Phân?

* Bước 1, tạo Symbolic Function A 1 x 1, trong ô duy nhất của nó chứa kí hiệu hàm chưa biết biểu thức, mình muốn tìm biểu thức của nó
* Bước 2, tạo Symbol B 1 x 1, trong ô duy nhất của nó chứa đẳng thức, mà trong đó chứa đạo hàm của A
* Bước 3, đưa B vào hàm giải phương trình vi phân, để trả về Symbol A 1 x 1 chứa lời giải

<Solution> = dsolve(<B>)

* <Solution> là Symbol
* Ví dụ
* Giải phương trình

* Ta có

syms y(x)

diffequa = diff(y, x) == y

solution = dsolve(diffequa)

* In ra Solution

* Để đặt ra điều kiện ban đầu

<Solution Cụ Thể> = dsolve(<Phương Trình Vi Phân>, <Các Điều Kiện>)

* Cách viết <Các Điều Kiện> giống như cách bạn viết trên giấy
* Ví dụ
* Giải phương trình

* Ta có

syms y(x)

diffequa = diff(y, x, 2) == y

dy = diff(y, x)

solution = dsolve(diffequa, [y(0) == 1 dy(1) == 12])

* In ra solution

1. Khai Triển Taylor?

<Chuỗi Taylor> = taylor(<Biểu Thức>, <Các Symbol>, <Vị Trí>, "Order", <Cấp>)

* <Biểu Thức> có thể là Symbol, Symbolic Expression hoặc Symbolic Function
* <Chuỗi Taylor> là Symbol, nhưng sẽ là Symbolic Function nếu <Biểu Thức> cũng vậy
* Ví dụ

foo = taylor(sin(a), a, 0, "Order", 5)

* In ra foo

1. Rút Gọn Biểu Thức?

<Biểu Thức Rút Gọn> = simplify(<Biểu Thức>)

* <Biểu Thức> phải là Symbol hoặc Symbolic Function, <Biểu Thức Rút gọn> sẽ cùng kiểu với <Biểu Thức>

1. Khai Triển Mọi Dấu Ngoặc?

<Biểu Thức Khai Triển> = expand(<Biểu Thức>)

* <Biểu Thức> phải là Symbol hoặc Symbolic Function, <Biểu Thức Rút gọn> sẽ cùng kiểu với <Biểu Thức>
* Ví dụ

foo = expand((x + y)\*2)

* In ra foo

2x + 2y

1. Tìm Nguyên Hàm?

<Nguyên Hàm> = int(<Biểu Thức>, <Symbol Muốn Tính Nguyên Hàm Theo>)

* <Biểu Thức> phải là Symbol hoặc Symbolic Function, <Nguyên Hàm> sẽ cùng kiểu với <Biểu Thức>
* <Nguyên Hàm> ở đây được tính giống như khi tính tay, và không có hằng số C
* Nếu không thể xác định được biểu thức của <Nguyên Hàm>, thì nó ở dạng tích phân luôn
* Ví dụ

foo = int(sin(x)/log(x), x)

* In ra foo

* Để tính tích phân trong khoảng nào đó

<Tích Phân> = int(

<Biểu Thức>, <Symbol Muốn Tính Tích Phân Theo>,

<Cận Dưới>, <Cận Trên>

)

* <Tích Phân> luôn là Symbol

1. Hàm Rẽ Nhánh?

<Hàm Rẽ Nhánh> = piecewise(

<Điều Kiện 1>, <Biểu Thức 1>,

<Điều Kiện 2>, <Biểu Thức 2>,

…,

<Điều Kiện N>, <Biểu Thức N>, <Biểu Thức Còn Lại>

)

* Các <Điều Kiện> và <Biểu Thức> phải là Symbol hoặc Symbolic Function,

<Hàm Rẽ Nhánh> sẽ là Symbolic Function chỉ cần 1 trong số chúng là Symbolic Function, còn không thì là Symbol

* Cách hoạt động của <Điều Kiện> giống If Else
* Ví dụ

foo = piecewise(x < 0, x^2, x < 5, x^4, x^3)

* In ra foo

1. Tính Tổng Dãy Số?

<Tổng> = symsum(

<Biểu Thức>, <Symbol Biến Đếm>, <Giá Trị Đầu>, <Giá Trị Cuối>

)

* <Biểu Thức> phải là Symbol hoặc Symbolic Function, <Giá Trị Đầu> và

<Giá Trị Cuối> có thể là Symbol, Symbolic Function hoặc số nguyên

* <Tổng> luôn là Symbol
* Ví dụ

foo = symsum(x^3, x, 1, 10)

* In ra foo được 3025

1. Cách Hiện Số Thập Phân Thay Vì Phân Số Khi In Symbol?

sympref("FloatingPointOutput", true)

* Để trở về trạng thái ban đầu

sympref("FloatingPointOutput", false)

1. Tính Divergence?

<Divergence> = divergence(<Trường Vector>, <Các Biến>)

* <Trường Vector> phải là Symbol hoặc Symbolic Function, số phần tử phải = số phần tử của <Các Biến>, vì nó sẽ lấy đạo hàm tương ứng với các biến này
* Ví dụ

foo = divergence([x^2 z^2+y t y^3], [x z y y])

* Khi in ra foo

1. Tính Curl?

<Curl> = curl(<Trường Vector>, <Các Biến>)

* Giống Divergence nhưng số phần tử của <Trường Vector> và <Các Biến> phải = 3

Plot:

1. Cách Hoạt Động?

* Y chang Matplotlib nhưng các Figure chạy song song và không làm tạm dừng chương trình, đồng thời có thể qua lại giữa các Figure và các Subplot

1. Tạo 1 Figure Mới Và Chuyển Sang Làm Việc Trên Figure Này?

<Figure> = figure(<Index>)

* Nếu đã có Figure với <Index> rồi thì chỉ chuyển sang làm việc trên Figure này
* Để trả Tensor có Shape (1, 4), lấy góc trái dưới của <Figure> làm mốc, 4 giá trị từ trái qua phải lần lượt có ý nghĩa là mốc ở bên phải cạnh trái màn hình bao nhiêu Pixel, mốc ở trên cạnh dưới màn hình bao nhiêu Pixel, chiều rộng và chiều cao <Figure> tính theo Pixel

<Figure>.Position

* Để chỉnh lại các thông số trên

<Figure>.Position = [<4 Thông Số>]

* Ví dụ

foo.Position = [0 0 500 500]

* Để trả về <Index> của <Figure>

<Figure>.Number

* Để kiểm tra xem <Figure> có bị đóng

<Check> = ishandle(<Figure>)

* <Check> = 1 khi có <Figure>, = 0 khi <Figure> bị đóng

1. Tạo 1 Subplot Trong Figure Hiện Tại Và Chuyển Sang Làm Việc Trên Đó?

subplot(<Số Hàng>, <Số Cột>, <Index>)

* Y chang khi tạo Subplot trong Matplotlib
* Nếu đã có Subplot vừa khít với Subplot này thì chuyển ngay sang làm việc trên Subplot đó
* Nếu Subplot này đè lên 1 tí Subplot khác là Subplot khác bay màu

1. Chỉ Định Trạng Thái Của Subplot Hiện Tại?

* Trạng thái mặc định của Subplot là nếu có Plot mới thì xóa toàn bộ Plot cũ trên nó, đồng thời không có lưới tọa độ, lưới tọa độ y chang Matplotlib
* Để chuyển sang trạng thái không xóa Plot cũ

hold on

* Để chuyển sang trạng thái mặc định sẽ xóa Plot cũ

hold off

* Để chuyển sang trạng thái có lưới tọa độ

grid on

* Để chuyển sang trạng thái mặc định không có lưới tọa độ

grid off

1. Plot Đồ Thị 2D Vào Subplot Hiện Tại?

<Line> = plot(<X>, <Y>, <Style>)

* Trường hợp 1, <X> và <Y> là ma trận chỉ có 1 hàng hoặc cột, nhưng có số phần tử = nhau
* Giống Plot 1 Line trong Matplotlib
* Trường hợp 2, <X> và <Y> là 2 ma trận cùng Shape, hoặc 1 trong 2 là cột hoặc hàng, khi đó số phần tử phải = số dòng của ma trận còn lại để Broadcast
* Giống Plot nhiều Line trong Matplotlib, mỗi cột trong ma trận là 1 Line
* <Style> là 1 String chỉ định màu, kiểu đường, … của tất cả các Line
* Ta có thể cập nhật lại <X> và <Y> của <Line> và ngay lập tức đồ thị của nó được cập nhật

<Line>.XData = <X Mới>

<Line>.YData = <Y Mới>

1. Plot 1 Đống Vector Vào Subplot Hiện Tại?

<Quiver> = quiver(

<X>, <Y>, <U>, <V>, 0,

'linewidth", <Độ Rộng Vector>,

'MaxHeadSize', <Độ To Đầu Mũi Tên>

)

* <X>, <Y>, <U>, <V> phải là các Tensor cùng Shape, gốc của các Vector sẽ có hoành độ là <X>, tung độ là <Y>, hướng phải là <U>, hướng trên là <V>
* Ta có thể cập nhật lại <X>, <Y>, <U>, <V> của <Quiver> và ngay lập tức đồ thị của nó được cập nhật

<Quiver>.XData = <X Mới>

<Quiver>.YData = <Y Mới>

<Quiver>.UData = <U Mới>

<Quiver>.VData = <V Mới>

1. Plot 1 Đống Điểm Rời Rạc Vào Subplot Hiện Tại?

<Scatter> = scatter(<X>, <Y>)

* <X> và <Y> phải là các Tensor sao cho số phần tử = nhau, đồng thời chỉ có tối đa 1 chiều có kích thước > 1
* Các điểm sẽ có hoành độ là <X> và tung độ là <Y>

1. Plot 1 Đống Chữ Vào Subplot Hiện Tại?

<Các Text> = text(<X>, <Y>, <Chữ>)

* <X>, <Y> phải là các Tensor sao cho số phần tử = nhau, đồng thời chỉ có tối đa 1 chiều có kích thước > 1
* <Chữ> phải là Tensor kiểu String sao cho Broadcast được với <X> và <Y>
* <Các Text> là 1 mảng gồm các <Text>
* Để thay đổi vị trí 1 <Text>

<Text>.Position = [<X> <Y>]

* Để thay đổi chữ của 1 <Text>

set(<Text>, "String", <Chữ>)

1. Xóa 1 Đối Tượng Line, Quiver, Scatter Hoặc Text Ra Khỏi Subplot?

delete(<Line Hoặc Quive Hoặc Scatter Hoặc Text>)

1. Plot Đồ Thị 1 Symbolic Function Là Hàm 1 Biến?

fplot(<Symbolic Function>, [<Cận Dưới> <Cận Trên>])

* Mặc định

|  |  |
| --- | --- |
| <Cận Dưới> | –5 |
| <Cận Trên> | 5 |

* Giống y chang plot đồ thị 2D, đồ thị sẽ bị cắt ở 2 <Cận>
* Ví dụ

foo = symfun(sin(X), [X])

fplot(foo, [5 10])

* Để Plot sử dụng phương trình tham số

fplot(<X Là Hàm Theo T>, <Y Là Hàm Theo T>, [<Cận Trên T> <Cận Dưới T>])

* Mặc định

|  |  |
| --- | --- |
| <Cận Dưới T> | –5 |
| <Cận Trên T> | 5 |

* Ví dụ

syms foo bar t

foo = sin(t)

bar = cos(t)

fplot(foo, bar, [3 9])

1. Chỉ Định Tên Của Trục Của Subplot Hiện Tại?

* Giống y chang trong Matplotlib
* Chỉ định tên trục hoành

xlabel(<Tên Trục Hoành>)

* Tương tự với các trục còn lại
* Nếu muốn kiểu mũ mà không phải kiểu “^2”, … mà là kiểu “2” thì viết

<String Đẹp> = sprintf(<String>)

* Ví dụ

foo = sprintf("m/s^2")

* Khi này, khi dùng foo để làm tên trục, nó sẽ hiện m/s2

1. Chỉ Định Tên Của Subplot Hiện Tại?

title(<Tên Subplot>)

1. Chú Thích Các Plot Trong Subplot Hiện Tại?

legend(<Các Chú Thích>)

* <Các Chú Thích> phải có số lượng = số Plot hiện tại
* Chú thích đầu tiên sẽ áp dụng cho cái Plot được Plot sớm nhất, chú thích thứ 2 sẽ áp dụng cho cái Plot được Plot thứ 2, …
* Ví dụ

legend("foo", "bar", "far")

* Lưu ý bảng chú thích có thể di chuyển được = cách kéo thả chuột

1. Chỉ Định Cận Của Subplot Hiện Tại?

* Giống y chang trong Matplotlib
* Chỉ định cận trục hoành

xlim([<Cận Trái> <Cận Phải>])

* Tương tự với các trục còn lại

1. Đóng Figure Hiện Tại?

close

* Để đóng toàn bộ Figure

close all

Simulink:

1. Phần Mở Rộng File Simulink?

* “.slx”

1. Mở 1 File Simulink?

* Mỗi 1 File Simulink mở ra sẽ cần 1 cửa sổ dành riêng cho nó
* Vào Tab “HOME” + tại mục “SIMULINK” + Click biểu tượng “Simulink” + Click biểu tượng “Open…” góc trái trên + chọn File Simulink + Click “Open” = mở File trong cửa sổ “Current Folder”

1. Thêm 1 Phần Tử Vào Mạch?

* Vào Tab “SIMULATION” + tại mục “LIBRARY” + Click biểu tượng “Library Browser” để mở cửa sổ thư viện + kéo thả phần tử vào cửa sổ chính hoặc chọn phần tử rồi nhấn “Ctrl” + “I” = Double Click vào chỗ bất kì trong cửa sổ chính + gõ tên phần tử + chọn phần tử
* Để nối cổng này của phần tử này với cổng kia của phần tử kia, kéo thả cổng này vào cổng kia, nếu xuất hiện mũi tên đen thì kết nối được, đỏ thì không được, 2 phần tử nối được với nhau chỉ khi chúng thuộc cùng 1 thẻ trong cửa sổ thư viện
* Để rẽ nhánh dây điện
* Nhấn giữ “Ctrl” + kéo thả 1 điểm trên dây
* Để chọn 1 phần tử
* Click chuột vào phần tử đó
* Để hiện hoặc ẩn tên bên dưới phần tử được chọn
* Di chuột lên dấu 3 chấm phía trên phần tử + Click biểu tượng thứ 2 từ trái sang
* Để kích hoạt hoặc vô hiệu hóa phần tử được chọn
* Di chuột lên dấu 3 chấm phía trên phần tử + Click biểu tượng thứ 1 từ trái sang
* Để hiện cửa sổ tham số của phần tử
* Phải chuột vào phần tử + chọn “Block Parameters” + chỉnh tham số + Click “OK” = Double Click vào phần tử + chỉnh tham số + Click “OK”
* Để nhân đôi nhanh 1 phần tử
* Nhấn giữ phải chuột vào phần tử rồi kéo thả
* Để quay 1 phần tử được chọn 90 độ sang phải
* Nhấn “Ctrl” + “R”

1. Chạy Mô Phỏng Mạch?

* Để chỉnh thời gian kết thúc mô phỏng
* Vào Tab “SIMULATION” + tại mục “SIMULATE” + điền số giây vào thuộc tính “Stop Time”, lưu ý đây không phải là thời gian để máy bạn chạy mô phỏng mà là thời gian ảo nên trôi rất nhanh
* Để chỉ mô phỏng sang giây tiếp theo rồi tạm dừng
* Vào Tab “SIMULATION” + tại mục “SIMULATE” + Click biểu tượng “Step Forward”
* Để tiếp tục mô phỏng hết thời gian còn lại
* Nhấn “Ctrl” + “T”

1. Thẻ Simulink?

* Nguồn điện thế hình Sin
* Vào “Sources” + kéo thả “Sine Wave”
* Oscilloscope
* Vào “Sinks” + kéo thả “Scope”
* Cửa sổ tham số của Oscilloscope sẽ hiển thị đồ thị tín hiệu chân Input của nó khi chạy mô phỏng
* Display
* Vào “Sinks” + kéo thả “Display”
* Phần tử này hiện điện thế của chân Input của nó khi chạy mô phỏng

1. Thẻ Simscape?

* Nguồn AC
* Vào “Foundation Library” + vào “Electrical” + vào “Electrical Sources” + kéo thả “AC Voltage Source”
* Cảm biến hiệu điện thế
* Vào “Foundation Library” + vào “Electrical” + vào “Electrical Sensors” + kéo thả “Voltage Sensor”
* Chân V = chân dương – chân âm, tín hiệu thuộc loại Simscape
* Cấu hình Solver
* Vào “Utilities” + kéo thả “Solver Configuration”
* Mỗi 1 mạch Simscape riêng biệt, nghĩa là giữa 2 điểm bất kì trên mạch luôn tồn tại 1 đường đi, đều phải được nối với duy nhất 1 cấu hình Solver, nối cấu hình Solver vào điểm nào trên mạch cũng được
* Chuyển đổi tín hiệu Simscape sang Simulink
* Vào “Utilites” + kéo thả “PS Simulink Converter”
* Điện trở
* Vào “Foundation Library” + vào “Electrical” + vào “Electrical Elements” + kéo thả “Resistor”
* Tụ điện
* Vào “Foundation Library” + vào “Electrical” + vào “Electrical Elements” + kéo thả “Capacitor”
* Mặc định = tụ điện thuần + chút điện trở
* Cuộn cảm
* Vào “Foundation Library” + vào “Electrical” + vào “Electrical Elements” + kéo thả “Inductor”
* Mặc định = cuộn cảm thuần + chút điện trở
* GND
* Vào “Foundation Library” + vào “Electrical” + vào “Electrical Elements” + kéo thả “Electrical Reference”

1. Thẻ RF Blocks?

* Nguồn điện thế hình Sin
* Vào “Circuit Envelope” + vào “Sources” + kéo thả “Sinusoid”
* Điện trở
* Vào “Circuit Envelope” + vào “Elements” + kéo thả “R”
* Tụ điện
* Vào “Circuit Envelope” + vào “Elements” + kéo thả “C”
* Cuộn cảm
* Vào “Circuit Envelope” + vào “Elements” + kéo thả “L”
* GND
* Vào “Circuit Envelope” + vào “Elements” + kéo thả “Gnd”