**GIỚI THIỆU HƯỚNG DẪN**

**Hướng dẫn MatLab**

MATLAB là một ngôn ngữ lập trình được phát triển bởi MathWorks. Nó bắt đầu như một ngôn ngữ lập trình ma trận trong đó việc lập trình đại số tuyến tính rất đơn giản. Nó có thể được chạy cả trong các phiên tương tác và dưới dạng một công việc hàng loạt.

Hướng dẫn này cung cấp cho bạn phần giới thiệu nhẹ nhàng về ngôn ngữ lập trình MATLAB. Nó được thiết kế để giúp sinh víên sử dụng thành thạo ngôn ngữ lập trình MATLAB. Các ví dụ MATLAB dựa trên vấn đề đã được đưa ra theo cách đơn giản và dễ dàng giúp cho việc học của bạn trở nên nhanh chóng và hiệu quả.

**Khán giả**

Hướng dẫn này đã được chuẩn bị cho người mới bắt đầu để giúp họ hiểu chức năng cơ bản đến nâng cao của MATLAB. Sau khi hoàn thành hướng dẫn này, bạn sẽ thấy mình có trình độ chuyên môn vừa phải trong việc sử dụng MATLAB, từ đó bạn có thể chuyển sang bước tiếp theo. cấp độ.

**Điều kiện tiên quyết**

Chúng tôi cho rằng bạn có một chút kiến thức về bất kỳ chương trình máy tính nào và hiểu các khái niệm như biến, hằng, biểu thức, câu lệnh, v.v. Nếu bạn đã lập trình bằng bất kỳ ngôn ngữ lập trình cấp cao nào khác như C, C++ hoặc Java, thì sẽ rất rất nhiều lợi ích và việc học MATLAB sẽ giống như một niềm vui đối với bạn.

**Bản quyền & Thông báo từ chối trách nhiệm**

Tất cả nội dung và hình ảnh trong hướng dẫn này là tài sản của tutuspoint.com. Bất kỳ nội dung nào từ tutorialspoint.com hoặc hướng dẫn này không được phân phối lại hoặc sao chép dưới bất kỳ hình thức, hình thức hoặc hình thức nào mà không có sự cho phép bằng văn bản của tutorialspoint.com. Không làm như vậy là ví phạm luật bản quyền.

Hướng dẫn này có thể chứa những điểm không chính xác hoặc sai sót và tutorialspoint không đảm bảo về tính chính xác của trang web hoặc nội dung của nó bao gồm cả hướng dẫn này. Nếu bạn phát hiện ra rằng trang web tutorialspoint.com hoặc nội dung hướng dẫn này có một số lỗi, vui lòng liên hệ với chúng tôi tại [webmaster@tutorialspoint.com](file://localhost/C:/Users/ZARA/Desktop/webmaster%40tutorialspoint.com)

Mục lục

## TỔNG QUAN

MATLAB (phòng thí nghiệm ma trận) là ngôn ngữ lập trình cấp cao thế hệ thứ tư và tương tác môi trường cho tính toán số, trực quan hóa và lập trình.

MATLAB được phát triển bởi MathWorks.

Nó cho phép thao tác ma trận; vẽ sơ đồ các chức năng và dữ liệu; thực hiện các thuật toán; tạo người dùng giao diện; giao tiếp với các chương trình được viết bằng các ngôn ngữ khác, bao gồm C, C++, Java và Fortran; phân tích dữ liệu; phát triển thuật toán; và tạo ra các mô hình và ứng dụng.

Nó có nhiều lệnh tích hợp sẵn và các hàm toán học giúp bạn thực hiện các phép tính toán học, tạo vẽ đồ thị và thực hiện các phương pháp số

### Sức mạnh toán học tính toán của MATLAB

MATLAB được sử dụng trong mọi khía cạnh của toán học tính toán. Sau đây là một số phép tính toán học thường được sử dụng trong đó nó được sử dụng phổ biến nhất:

* Xử lý ma trận và Mảng
* Vẽ đồ thị 2-D và 3-D và đồ họa
* Đại số học tuyến tính.
* phương trình đại số
* Hàm phi tuyến tính.
* Số liệu thống kê
* Dữ liệu Phân tích
* Giải tích và ví phân phương trình
* số phép tính
* Hội nhập
* Biến đổi
* Đường cong lắp
* Nhiều đặc biệt khác chức năng

### Các tính năng của MATLAB

Sau đây là các tính năng cơ bản của MATLAB:

* Nó là một ngôn ngữ cấp cao để tính toán số, trực quan hóa và ứng dụng phát triển.
* Nó cũng cung cấp một môi trường tương tác để khám phá, thiết kế và giải quyết vấn đề lặp đi lặp lại. giải quyết.
* Nó cung cấp thư víện rộng lớn các hàm toán học cho đại số tuyến tính, thống kê, phân tích Fourier, lọc, tối ưu hóa, tích phân số và giải ví phân thường phương trình.
* Nó cung cấp đồ họa tích hợp để trực quan hóa dữ liệu và các công cụ để tạo tùy chỉnh âm mưu.
* Giao diện lập trình của MATLAB cung cấp các công cụ phát triển để cải thiện chất lượng mã, khả năng bảo trì và tối đa hóa hiệu suất.
* Nó cung cấp các công cụ để xây dựng các ứng dụng với đồ họa tùy chỉnh giao diện.
* Nó cung cấp các chức năng để tích hợp các thuật toán dựa trên MATLAB với các ứng dụng và ngôn ngữ bên ngoài như C, Java, .NET và Microsoft Excel.

### Công dụng của MATLAB

MATLAB được sử dụng rộng rãi như một công cụ tính toán trong khoa học và kỹ thuật bao gồm các lĩnh vực vật lý, hóa học, toán học và tất cả các luồng kỹ thuật. Nó được sử dụng trong một loạt các ứng dụng bao gồm:

* Xử lý tín hiệu và truyền thông
* Hình ảnh và Xử lý vídeo
* Điều khiển hệ thống
* Kiểm tra và Đo đạc
* Tính toán Tài chính
* Tính toán Sinh học

## HƯỚNG DẪN SỬ DỤNG MATLAB

### M File

M file là tập hợp các lệnh được lưu vào trong một file.

Trong MATLAB, tập tin m-file (hay còn gọi là script file) là một tập tin văn bản chứa các lệnh MATLAB được lưu trữ để thực thi các tính toán và các tác vụ khác trong môi trường MATLAB. Các tập tin m-file có phần mở rộng là ".m" và chứa các lệnh MATLAB, các biến, hàm, kịch bản hoặc các mã chương trình để thực thi trong MATLAB. Bạn có thể chạy tập tin m-file bằng cách sử dụng lệnh "run" hoặc nhấn nút "Run" trên giao diện MATLAB.

## TỆP - M

Cho đến nay, chúng tôi đã sử dụng môi trường MATLAB như một máy tính. Tuy nhiên, MATLAB cũng là một công cụ mạnh mẽ ngôn ngữ lập trình, cũng như một môi trường tính toán tương tác.

Trong các chương trước, bạn đã học cách nhập lệnh từ dấu nhắc lệnh của MATLAB. MATLAB cũng cho phép bạn viết một loạt lệnh vào một tệp và thực thi tệp dưới dạng một đơn vị hoàn chỉnh, giống như viết một hàm và gọi nó.

### Tập tin M

MATLAB cho phép viết hai loại tệp chương trình:

* **Scripts** - Tập tin **Scripts** là tập tin chương trình có phần mở rộng .m . Trong các tệp này, bạn viết một loạt lệnh mà bạn muốn thực hiện cùng nhau. Tập lệnh không chấp nhận đầu vào và không trả về bất kỳ đầu ra nào. Họ hoạt động trên dữ liệu trong không gian làm việc.
* **Functions** - Tệp hàm cũng là tệp chương trình có phần mở rộng .m . Các chức năng có thể chấp nhận đầu vào và trả về đầu ra. Các biến nội bộ là cục bộ của chức năng. Bạn có thể sử dụng MATLAB Editor hoặc bất kỳ trình soạn thảo văn bản nào khác để tạo các tệp .m của mình. Trong phần này, chúng ta sẽ thảo luận về các tệp script. Tệp tập lệnh chứa nhiều dòng lệnh MATLAB và lệnh gọi hàm tuần tự. Bạn có thể chạy tập lệnh bằng cách nhập tên của tập lệnh tại lệnh dòng.

### Tạo và chạy tập lệnh tập tin

Để tạo tệp script, bạn cần sử dụng trình soạn thảo văn bản. Bạn có thể mở trình soạn thảo MATLAB theo hai cách:

* Sử dụng lệnh lời nhắc
* Sử dụng IDE

Nếu bạn đang sử dụng dấu nhắc lệnh, hãy nhập **edit** trong dấu nhắc lệnh. Thao tác này sẽ mở trình chỉnh sửa. Bạn có thể nhập trực tiếp **edit** và sau đó nhập tên tệp (với phần mở rộng .m)

|  |
| --- |
| edit  or  edit <filename> |

Lệnh trên sẽ tạo tệp trong thư mục MATLAB mặc định. Nếu bạn muốn lưu trữ tất cả các tệp chương trình trong một thư mục cụ thể, thì bạn sẽ phải cung cấp toàn bộ đường dẫn.

Hãy để chúng tôi tạo một thư mục có tên progs. Nhập các lệnh sau tại dấu nhắc lệnh (>>):

|  |
| --- |
| mkdir progs % create directory progs under default directory  chdir progs % changing the current directory to progs  edit prog1.m % creating an m file named prog1.m |

Nếu bạn tạo tệp lần đầu tiên, MATLAB sẽ nhắc bạn xác nhận nó. Nhấp vào Có.

Hình:

Ngoài ra, nếu bạn đang sử dụng IDE, hãy chọn **New** **Script**. Thao tác này cũng mở trình chỉnh sửa và tạo một tệp có tên Untitled. Bạn có thể đặt tên và lưu tệp sau khi nhập mã.

Nhập mã sau vào trình chỉnh sửa:

|  |
| --- |
| NoOfStudents = 6000;  TeachingStaff = 150;  NonTeachingStaff = 20;  Total = NoOfStudents + TeachingStaff...  +NonTeachingStaff;  disp(Total); |

Sau khi tạo và lưu tệp, bạn có thể chạy tệp theo hai cách:

* Nhấp vào nút Run trên cửa sổ soạn thảo hoặc
* Chỉ cần gõ tên tệp (không có phần mở rộng) trong dấu nhắc lệnh: >> prog1

Dấu nhắc cửa sổ lệnh hiển thị kết quả:

|  |
| --- |
| 6170 |

**Ví dụ**: Tạo một tệp tập lệnh và nhập mã sau:

|  |
| --- |
| a = 5; b = 7;  c = a + b  d = c + sin(b)  e =5 \* d  f = exp(-d) |

Khi đoạn mã trên được biên dịch và thực thi, nó sẽ tạo ra kết quả như sau:

|  |
| --- |
| c =  12  d =  2177/172  e =  11771/186  f =  1/313949 |

## MÔI TRƯỜNG

Bạn thực sự không cần thiết lập môi trường của riêng mình để bắt đầu học lập trình ngôn ngữ MATLAB/Octave. Lý do rất đơn giản, chúng tôi đã thiết lập môi trường Octave trực tuyến để bạn có thể thực hiện tất cả các ví dụ có sẵn trực tuyến cùng một lúc khi bạn đang làm bài tập lý thuyết của mình. Điều này giúp bạn tự tin trong những gì bạn đang đọc và để kiểm tra kết quả với các tùy chọn khác nhau.

Vui lòng sửa đổi bất kỳ ví dụ nào và thực hiện nó trực tuyến bằng cách sử dụng tùy chọn **DÙNG THỬ** tại trang web: https://www.mathworks.com/products/matlab-online.html

|  |
| --- |
| x = [1 2 3 4 5 6 7 8 9 10];  y1 = [.16 .08 .04 .02 .013 .007 .004 .002 .001 .0008];  y2 = [.16 .07 .03 .01 .008 .003 .0008 .0003 .00007 .00002];  semilogy(x,y1,'-bo',x,y2,'-kx');  title('Plot title');  xlabel('X Axis');  ylabel('Y Axis');  print('graph.eps', '-depsc'); |

**Sau khi chạy** Câu lệnh sẽ tạo ra biểu đồ đường đúng và tệp EPS "graph.eps".

### Cài đặt môi trường cục bộ

Nếu bạn vẫn sẵn sàng thiết lập môi trường của mình, hãy để tôi nói cho bạn một bí mật, việc thiết lập môi trường MATLAB chỉ cần vài cú nhấp chuột. Tuy nhiên, bạn cần tải xuống trình cài đặt từ [đây :](https://www.mathworks.com/downloads/web_downloads/)

MathWorks cung cấp sản phẩm được cấp phép, phiên bản dùng thử và cả phiên bản dành cho sinh víên. Bạn cần đăng nhập vào trang web và đợi một chút sự chấp thuận.

Khi bạn nhận được liên kết tải xuống, như tôi đã nói, chỉ cần vài cú nhấp chuột:

### Hiểu môi trường MATLAB

Bạn có thể khởi chạy IDE phát triển MATLAB từ biểu tượng được tạo trên màn hình của bạn. Cửa sổ làm việc chính trong MATLAB được gọi là màn hình nền. Khi bạn khởi động MATLAB, màn hình nền sẽ xuất hiện trong bố cục mặc định của nó:

Máy tính để bàn có các bảng sau:

* Thư mục hiện tại - Bảng này cho phép bạn truy cập các thư mục dự án của mình và các tập tin.
* Cửa sổ lệnh - Đây là khu vực chính nơi bạn nhập lệnh tại dòng lệnh, được biểu thị bằng dấu nhắc lệnh (>>).
* Không gian làm việc - Không gian làm việc hiển thị tất cả các biến bạn tạo và/hoặc nhập từ đó các tập tin.
* Lịch sử lệnh - Bảng này hiển thị hoặc chạy lại các lệnh mà bạn đã nhập tại lệnh dòng.

### Thiết lập GNU Octave

Nếu bạn sẵn sàng sử dụng Octave trên máy của mình (Linux, BSD, OS X hoặc Windows), vui lòng tải xuống phiên bản mới nhất từ [Tải xuống GNU Octave .](http://www.gnu.org/software/octave/download.html) Bạn có thể kiểm tra hướng dẫn cài đặt đã cho cho máy của mình.

## CÚ PHÁP CƠ BẢN

Môi trường MATLAB hoạt động giống như một máy tính siêu phức tạp. Bạn có thể nhập lệnh tại dấu nhắc lệnh >>. MATLAB là một môi trường thông dịch. Nói cách khác, bạn đưa ra một lệnh và MATLAB thực hiện nó ngay lập tức.

### Thực hành

**Ví dụ:** Nhập một biểu thức toán học đơn giản

|  |
| --- |
| 5 + 5 |

Khi bạn nhấn **Enter,** MATLAB sẽ thực thi ngay lập tức và kết quả trả về là:

|  |
| --- |
| ans =   10 |

**Ví dụ:** Thực hiện phép toán số mũ

|  |
| --- |
| 3 ^ 2 |

Khi bạn nhấn **Enter,** MATLAB sẽ thực thi ngay lập tức và kết quả trả về là:

|  |
| --- |
| ans =  9 |

**Ví dụ:**

|  |
| --- |
| sin(pi/2) |

Khi bạn nhấn **Enter,** MATLAB sẽ thực thi ngay lập tức và kết quả trả về là:

|  |
| --- |
| ans =  1 |

**Ví dụ:** Thực hiện phép toán chia cho 0

|  |
| --- |
| 7 / 0 |

Khi bạn nhấn **Enter,** MATLAB sẽ thực thi ngay lập tức và kết quả trả về là:

|  |
| --- |
| ans =  Inf  warning: division by zero |

**Ví dụ:**

|  |
| --- |
| 732\*20.3 |

Khi bạn nhấn **Enter,** MATLAB sẽ thực thi ngay lập tức và kết quả trả về là:

|  |
| --- |
| ans =  1.4860e+04 |

MATLAB cung cấp một số biểu thức đặc biệt cho một số ký hiệu toán học như:

* pi cho π
* Inf cho ∞
* i (và j) cho √-1
* …
* Nan là viết tắt của 'not a number'.

### Sử dụng dấu chấm phẩy (;)

Dấu chấm phẩy (;) chỉ kết thúc câu lệnh. Tuy nhiên, nếu bạn muốn chặn và ẩn đầu ra MATLAB cho một biểu thức, hãy thêm dấu chấm phẩy sau biểu thức.

**Ví dụ:**

|  |
| --- |
| x = 3;  y = x + 5 |

Khi bạn nhấn **Enter,** MATLAB sẽ thực thi ngay lập tức và kết quả trả về là:

|  |
| --- |
| y =  8 |

### Thêm chú thích

Chú thích trong MATLAB được sử dụng để cung cấp thông tin về các dòng mã của bạn. Các chú thích có thể được sử dụng để giải thích mục đích của từng dòng mã, cung cấp thông tin về đơn vị đo, giải thích các công thức và phương trình, và đặc biệt là giúp cho mã của bạn dễ đọc hơn và dễ hiểu hơn.

Các chú thích trong MATLAB có hai loại chính:

1. Chú thích trên một dòng: Sử dụng ký hiệu % để bắt đầu chú thích. Bất kỳ dòng nào bắt đầu bằng % sẽ được xem như là một chú thích và sẽ không được thực thi bởi MATLAB.

**Ví dụ:**

|  |
| --- |
| % Tính diện tích hình chữ nhật  width = 5; % chiều rộng  height = 10; % chiều dài  area = width \* height; % tính diện tích |

1. Chú thích trên nhiều dòng: Sử dụng ký hiệu %{ và %} để bắt đầu và kết thúc chú thích. Bất kỳ dòng nào nằm giữa %{ và %} sẽ được xem như là một chú thích và sẽ không được thực thi bởi MATLAB.

**Ví dụ:**

|  |
| --- |
| %{  Đây là một chú thích trên nhiều dòng.  Nó có thể được sử dụng để giải thích các phương trình hoặc các thuật toán phức tạp.  %} |

Bên cạnh đó, còn có cách sử dụng hàm help để tạo ra một tài liệu mô tả cho một hàm MATLAB hoặc tệp kịch bản, hoặc sử dụng doc để mở tài liệu MATLAB chính thức trực tiếp trong trình duyệt.

Biểu tượng phần trăm (%) được sử dụng để biểu thị một dòng chú thích.

**Ví dụ:**

|  |
| --- |
| x = 9 % Value x is 9 |

Bạn cũng có thể nhận xét bằng cách sử dụng các toán tử nhận xét khối % { và % }.

Trình soạn thảo MATLAB bao gồm các công cụ và mục trình đơn ngữ cảnh để giúp bạn thêm, xóa hoặc thay đổi định dạng của nhận xét.

### Toán tử thường được sử dụng và Đặc biệt Nhân vật

MATLAB hỗ trợ các toán tử và ký tự đặc biệt thông dụng sau:

|  |  |
| --- | --- |
| **Toán tử** | **Mục đích** |
| + | Thêm; Toán tử cộng. |
| - | Dấu trừ; Toán tử trừ. |
| \* | Toán tử nhân vô hướng và ma trận. |
| .\* | Toán tử nhân mảng. |
| ^ | Toán tử lũy thừa vô hướng và ma trận. |
| .^ | Toán tử lũy thừa mảng. |
| \ | Toán tử chia trái. |
| / | Toán tử chia phải. |
| .\ | Toán tử chia trái mảng. |
| ./ | Toán tử chia phải mảng. |
| : | Tạo các phần tử cách đều nhau và đại diện cho toàn bộ hàng hoặc cột. |
| ( ) | Dấu ngoặc đơn; bao gồm các đối số hàm và chỉ số mảng; ghi đè quyền ưu tiên. |
| [ ] | Chân đế; bao vây các phần tử mảng. |
| . | Dấu thập phân. |
| … | Dấu chấm lửng; Toán tử nối tiếp dòng |
| , | Dấu phẩy; tách các câu lệnh và các phần tử trong một hàng |
| ; | Dấu chấm phẩy; tách các cột và chặn hiển thị. |
| % | Ký hiệu phần trăm; chỉ định một nhận xét và chỉ định định dạng. |
| \_ | Ký hiệu trích dẫn và toán tử chuyển vị. |
| .\_ | Toán tử chuyển vị không liên hợp. |
| = | Toán tử gán. |

### Các biến và hằng số đặc biệt

MATLAB hỗ trợ các biến và hằng số đặc biệt sau:

|  |  |
| --- | --- |
| **Tên** | **Nghĩa** |
| Ans | Câu trả lời gần đây nhất. |
| Eps | Độ chính xác của độ chính xác dấu phẩy động. |
| i,j | Đơn vị ảo √-1. |
| Inf | Vô cực. |
| NaN | Kết quả số không xác định (không phải là số). |
| Pi | Hằng số π |

### Đặt tên biến

Tên biến bao gồm một chữ cái theo sau bởi bất kỳ số lượng chữ cái, chữ số hoặc dấu gạch dưới nào. MatLab phân biệt chữ hoa chữ thường .

Tên biến có thể có độ dài bất kỳ, tuy nhiên, MATLAB chỉ sử dụng N ký tự đầu tiên, trong đó N được cung cấp bởi hàm **namelengthmax**.

### Lưu công việc của bạn

Lệnh **save** được sử dụng để lưu tất cả các biến trong không gian làm việc, dưới dạng tệp có phần mở rộng .mat, trong thư mục hiện tại.

**Ví dụ:**

|  |
| --- |
| save myfile |

Bạn có thể tải lại tệp bất cứ lúc nào sau đó bằng cách sử dụng lệnh **load**.

|  |
| --- |
| load myfile |

## 

## BIẾN

Trong môi trường MATLAB, mọi biến đều là mảng hoặc ma trận.

**Ví dụ:** Bạn có thể gán các biến một cách đơn giản.

|  |
| --- |
| x = 3 % Định nghĩa x và khởi tạo giá trị |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| x =  3 |

Nó tạo ra một ma trận 1-by-1 có tên là x và lưu trữ giá trị 3 trong phần tử của nó.

**Ví dụ:**

|  |
| --- |
| x = sqrt(16) % Định nghĩa x và khởi tạo giá trị |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| x =  4 |

**Lưu ý:**

* Khi một biến được nhập vào hệ thống, bạn có thể tham khảo nó sau.
* Các biến phải có giá trị trước khi chúng được được sử dụng.
* Khi một biểu thức trả về một kết quả không được gán cho bất kỳ biến nào, hệ thống sẽ gán nó cho một biến có tên là ans, có thể được sử dụng sau.

**Ví dụ:**

|  |
| --- |
| sqrt(78) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans =  8.8318 |

Bạn có thể sử dụng biến này ans:

|  |
| --- |
| 9876 / ans |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans =  1.1182e+03 |

**Ví dụ:** Hãy xem xét một ví dụ khác:

|  |
| --- |
| x =7\*8;  y = x \*7.89 |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| y =   441.8400 |

### Nhiều bài tập

Bạn có thể có nhiều câu lệnh trên cùng một dòng.

**Ví dụ:**

|  |
| --- |
| a = 2; b = 7; c = a \* b |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| c =   14 |

### Hiển thị các biến đã sử dụng

**Ví dụ:** Lệnh who hiển thị tất cả các tên biến mà bạn đã sử dụng.

|  |
| --- |
| who |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| Your variables are:  a b c x y |

Lệnh whos hiển thị thêm một chút về các biến:

* Các biến hiện tại kỉ niệm
* Loại của mỗi biến
* Bộ nhớ được phân bổ cho mỗi Biến đổi
* Cho dù chúng là các biến phức tạp hay không phải

|  |
| --- |
| whos |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

**Ví dụ:** Lệnh **clear** xóa tất cả các biến hoặc xóa biến đã chỉ định ra khỏi bộ nhớ.

|  |
| --- |
| clear x % nó sẽ xóa biến x, không hiển thị gì cả  clear all % nó sẽ xóa tất cả các biến trong không gian làm việc |

### Bài tập dài

Câu lệnh dài có thể được viết xuống dòng bên dưới bằng cách sử dụng dấu chấm lửng (...)

**Ví dụ:**

|  |
| --- |
| a = 100;  b = 200;  c = 500;  result = a ...  + b \* c |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| result =   100100 |

### Lệnh định dạng (The format Command)

Theo mặc định, MATLAB hiển thị các số có bốn giá trị vị trí thập phân. Điều này được gọi là **short format**. Tuy nhiên, nếu muốn chính xác hơn, bạn cần sử dụng lệnh **format**.

Lệnh **format long** hiển thị 16 chữ số sau số thập phân.

**Ví dụ:**

|  |
| --- |
| format long  x = 7+10/3+5^1.2 |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| x =  17.231981640639408 |

**Ví dụ:** Một ví dụ khác:

|  |
| --- |
| format short  x = 7+10/3+5^1.2 |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| x =  17.2320 |

Lệnh định dạng **bank** làm tròn số đến hai chữ số thập phân.

**Ví dụ:**

|  |
| --- |
| format bank  a = 2.2555;  result = a \* 3 |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| result =  6.77 |

MATLAB hiển thị các số lớn sử dụng ký hiệu hàm mũ.

Lệnh **format short e** cho phép hiển thị ở dạng số mũ với bốn chữ số thập phân cộng với số mũ.

**Ví dụ:**

|  |
| --- |
| format short e  4.6789 \* 4.987654 |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans =  2.3337e+01 |

Lệnh **format long e** cho phép hiển thị ở dạng số mũ với bốn chữ số thập phân cộng với số mũ.

**Ví dụ:**

|  |
| --- |
| format long e  x = pi |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| x =  3.141592653589793e+00 |

Lệnh định dạng **rat** đưa ra biểu thức hữu tỷ gần nhất do một phép tính.

**Ví dụ:**

|  |
| --- |
| format rat  4.678\*4.9 |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans =  2063/90 |

### Tạo vectơ

Một vectơ là một mảng một chiều của các số. MATLAB cho phép tạo hai loại vectơ:

* Vectơ hàng
* Vectơ cột

#### Vectơ hàng

**Vectơ hàng:** được tạo bằng cách đặt tập hợp các phần tử trong dấu ngoặc vuông, sử dụng dấu cách hoặc dấu phẩy để phân tách các phần tử.

**Ví dụ:**

|  |
| --- |
| r = [7 8 9 10 11] |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| r =  7 8 9 10 11 |

**Ví dụ:** Một ví dụ khác,

|  |
| --- |
| r =[7 8 9 10 11];  t =[2,3,4,5,6];  result = r + t |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| result =  9 11 13 15 17 |

#### Vectơ cột

**Vectơ cột**: được tạo bằng cách đặt tập hợp các phần tử trong dấu ngoặc vuông, sử dụng dấu chấm phẩy (;) để phân tách các phần tử.

|  |
| --- |
| c =[7;8;9;10;11] |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| c =  7   8   9   10   11 |

### Tạo ma trận

Ma trận là một mảng hai chiều các số. Trong MATLAB, một ma trận được tạo bằng cách nhập mỗi hàng dưới dạng một chuỗi các phần tử được phân tách bằng dấu cách hoặc dấu phẩy và phần cuối của hàng được phân định bằng dấu chấm phẩy.

**Ví dụ:** Tạo ma trận 3 x 3 như sau:

|  |
| --- |
| matrix = [1 2 3; 4 5 6; 7 8 9] |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| matrix =  1 2 3   4 5 6   7 8 9 |

## LỆNH

MATLAB là một chương trình tương tác để tính toán số và trực quan hóa dữ liệu. Bạn có thể nhập một lệnh bằng cách gõ nó tại dấu nhắc MATLAB '>>' trên Cửa sổ Lệnh.

Trong phần này, chúng tôi sẽ cung cấp danh sách các lệnh MATLAB chung thường được sử dụng.

### Các lệnh để quản lý một phiên bản

MATLAB cung cấp nhiều lệnh khác nhau để quản lý phiên làm việc. Bảng sau đây cung cấp tất cả các lệnh như vậy:

|  |  |
| --- | --- |
| **Lệnh** | **Mục đích** |
| clc | Xóa cửa sổ lệnh. |
| clear | Xóa các biến khỏi bộ nhớ. |
| exis | Kiểm tra sự tồn tại của tệp hoặc biến. |
| global | Khai báo các biến là toàn cầu. |
| help | Tìm kiếm một chủ đề trợ giúp. |
| lookfor | Tìm kiếm các mục trợ giúp cho một từ khóa. |
| quit | Dừng MATLAB. |
| who | Liệt kê các biến hiện tại. |
| whos | Liệt kê các biến hiện tại (hiển thị dài). |

### Các lệnh thao tác với hệ thống

MATLAB cung cấp nhiều lệnh hữu ích khác nhau để làm việc với hệ thống, chẳng hạn như lưu công việc hiện tại trong không gian làm việc dưới dạng tệp và tải tệp sau.

Nó cũng cung cấp các lệnh khác nhau cho các hoạt động liên quan đến hệ thống khác như: hiển thị ngày, liệt kê các tệp trong thư mục, hiển thị thư mục hiện tại, …

Bảng sau hiển thị một số lệnh liên quan đến hệ thống thường được sử dụng:

|  |  |
| --- | --- |
| **Lệnh** | **Mục đích** |
| Cd | Thay đổi thư mục hiện tại. |
| Date | Hiển thị ngày hiện tại. |
| Delete | Xóa một tập tin. |
| Diary | Bật/tắt ghi tập tin nhật ký |
| Dir | Liệt kê tất cả các tệp trong thư mục hiện tại. |
| Load | Tải các biến không gian làm việc từ một tệp |
| Path | Hiển thị đường dẫn tìm kiếm. |
| Pwd | Hiển thị thư mục hiện tại. |
| Save | Lưu các biến không gian làm việc trong một tệp. |
| Type | Hiển thị nội dung của một tập tin. |
| What | Liệt kê tất cả các tệp MATLAB trong thư mục hiện tại. |
| wklread | Đọc tệp bảng tính .wk1 |

### Lệnh đầu vào và đầu ra

MATLAB cung cấp các lệnh liên quan đến đầu vào và đầu ra sau:

|  |  |
| --- | --- |
| **Lệnh** | **Mục đích** |
| Disp | Hiển thị nội dung của một mảng hoặc chuỗi. |
| Fscanf | Đọc dữ liệu được định dạng từ một tập tin. |
| Format | Kiểm soát định dạng hiển thị màn hình. |
| Fprintf | Thực hiện ghi định dạng vào màn hình hoặc tập tin. |
| Input | Kiểm soát định dạng hiển thị màn hình. |
| ; | Ngăn chặn in ấn màn hình. |

Các lệnh **fscanf** và **fprintf** hoạt động giống như các hàm scanf và printf của C. Chúng hỗ trợ các mã định dạng sau:

|  |  |
| --- | --- |
| **Lệnh** | **Mục đích** |
| %s | Định dạng dưới dạng chuỗi. |
| %d | Định dạng dưới dạng số nguyên. |
| %f | Định dạng dưới dạng giá trị dấu phẩy động. |
| %e | Định dạng dưới dạng giá trị dấu phẩy động trong ký hiệu khoa học. |
| %g | Định dạng ở dạng nhỏ gọn nhất: %f hoặc %e. |
| \n | Chèn một dòng mới vào chuỗi đầu ra. |
| \t | Chèn một tab vào chuỗi đầu ra. |

Hàm định dạng có các dạng sau được sử dụng để hiển thị số:

|  |  |
| --- | --- |
| **Lệnh** | **Mục đích** |
| format short | Bốn chữ số thập phân (mặc định). |
| format long | 16 chữ số thập phân. |
| format short e | Năm chữ số cộng với số mũ. |
| format long e | 16 chữ số cộng với số mũ. |
| format bank | Hai chữ số thập phân. |
| format + | Tích cực, tiêu cực, hoặc bằng không. |
| format rat | xấp xỉ hợp lý. |
| format compact | Loại bỏ một số nguồn cấp dữ liệu dòng. |
| format loose | Đặt lại về chế độ hiển thị nhỏ gọn hơn. |

### Các lệnh vectơ, ma trận và mảng

Bảng sau đây hiển thị các lệnh khác nhau được sử dụng để làm việc với mảng, ma trận và vectơ:

|  |  |
| --- | --- |
| **Lệnh** | **Mục đích** |
| Cat | Nối các mảng. |
| Find | Tìm chỉ số của các phần tử khác không. |
| Length | Tính số phần tử. |
| linspace | Tạo vector cách đều nhau. |
| logspace | Tạo véc tơ cách đều logarit. |
| Max | Trả về phần tử lớn nhất. |
| Min | Trả về phần tử nhỏ nhất. |
| Prod | Sản phẩm của mỗi cột. |
| reshape | Thay đổi kích thước. |
| Size | Tính toán kích thước mảng. |
| Sort | Sắp xếp từng cột. |
| Sum | Tính tổng từng cột. |
| Eye | Tạo một ma trận nhận dạng. |
| Ones | Tạo một mảng của những người. |
| Zeros | Tạo một mảng các số không. |
| Cross | Tính toán các sản phẩm chéo của ma trận. |
| Dot | Tính toán các sản phẩm chấm ma trận. |
| Det | Tính định thức của một mảng. |
| Inv | Tính nghịch đảo của ma trận. |
| Pinv | Tính nghịch đảo giả của ma trận. |
| Rank | Tính hạng của một ma trận. |
| Rref | Tính toán hình thức bậc thang hàng giảm. |
| Cell | Tạo mảng ô. |
| celldisp | Hiển thị mảng ô. |
| cellplot | Hiển thị biểu diễn đồ họa của mảng ô. |
| num2cell | Chuyển đổi mảng số thành mảng ô. |
| Deal | So khớp danh sách đầu vào và đầu ra. |
| Iscell | Xác định mảng ô. |

### Vẽ các lệnh

MATLAB cung cấp nhiều lệnh để vẽ đồ thị. Bảng sau đây cho thấy một số lệnh thường được sử dụng cho âm mưu:

|  |  |
| --- | --- |
| **Lệnh** | **Mục đích** |
| axis | Đặt giới hạn trục. |
| fplot | Vẽ đồ thị thông minh của các chức năng. |
| grid | Hiển thị đường lưới. |
| plot | Tạo biểu đồ xy. |
| print | In đồ thị hoặc lưu đồ thị vào một tập tin. |
| title | Đặt văn bản ở trên cùng của cốt truyện. |
| xlabel | Thêm nhãn văn bản vào trục x. |
| ylabel | Thêm nhãn văn bản vào trục y. |
| axes | Tạo các đối tượng trục. |
| close | Đóng cốt truyện hiện tại. |
| close all | Đóng tất cả các lô. |
| figure | Mở một cửa sổ hình mới. |
| gtext | Cho phép đặt nhãn bằng chuột. |
| hold | Đóng băng cốt truyện hiện tại. |
| legend | Vị trí huyền thoại bằng chuột. |
| refresh | Vẽ lại cửa sổ hình hiện tại. |
| set | Chỉ định các thuộc tính của các đối tượng như trục. |
| subplot | Tạo các ô trong các cửa sổ con. |
| text | Đặt chuỗi trong hình. |
| bar | Tạo biểu đồ thanh. |
| loglog | Tạo cốt truyện log-log. |
| polar | Tạo âm mưu cực. |
| semilogy | Tạo biểu đồ semilog. (tọa độ logarit). |
| stairs | Tạo cốt truyện cầu thang. |
| stem | Tạo biểu đồ gốc. |

## CÁC LOẠI DỮ LIỆU

MATLAB không yêu cầu bất kỳ khai báo kiểu hoặc báo cáo kích thước nào. Bất cứ khi nào MATLAB gặp một tên biến mới, nó sẽ tạo biến đó và phân bổ không gian bộ nhớ thích hợp.

Nếu biến đã tồn tại, thì MATLAB sẽ thay thế nội dung ban đầu bằng nội dung mới và phân bổ biến mới không gian lưu trữ, khi cần thiết.

**Ví dụ:**

|  |
| --- |
| Total = 42 |

Câu lệnh trên tạo ra một ma trận 1-by-1 có tên 'Total' và lưu trữ giá trị 42 trong đó.

### Các kiểu dữ liệu có sẵn trong MATLAB

MATLAB cung cấp 15 kiểu dữ liệu cơ bản. Mọi kiểu dữ liệu đều lưu trữ dữ liệu ở dạng ma trận hoặc mảng. Kích thước của ma trận hoặc mảng này tối thiểu là 0 x 0 và điều này có thể phát triển thành ma trận hoặc mảng có kích thước bất kỳ.

Bảng sau đây cho thấy các kiểu dữ liệu được sử dụng phổ biến nhất trong MATLAB:

|  |  |
| --- | --- |
| **Lệnh** | **Mục đích** |
| int8 | Số nguyên có dấu 8 bit |
| uint8 | Số nguyên không dấu 8 bit |
| int16 | Số nguyên có dấu 16 bit |
| uint16 | Số nguyên không dấu 16 bit |
| int32 | Số nguyên có dấu 32 bit |
| uint32 | Số nguyên không dấu 32 bit |
| int64 | Số nguyên có dấu 64 bit |
| uint64 | Số nguyên không dấu 64-bit |
| single | Dữ liệu số chính xác duy nhất |
| double | Dữ liệu số chính xác kép |
| logical | Các giá trị logic là 1 hoặc 0, tương ứng là đúng và sai |
| char | Dữ liệu ký tự (chuỗi được lưu trữ dưới dạng vectơ ký tự) |
| cell array | Mảng các ô, mỗi ô có khả năng lưu trữ một mảng có kích thước và kiểu dữ liệu khác nhau |
| structure | Các cấu trúc giống như C |
| function handle | Con trỏ đến một chức năng |
| user classes | Các đối tượng được xây dựng từ một lớp do người dùng định nghĩa |
| java classes | Các đối tượng được xây dựng từ một lớp Java |

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| str = 'Hello World!'  n = 2345  d = double(n)  un = uint32(789.50)  rn =5678.92347  c = int32(rn) |

Khi đoạn mã trên được biên dịch và thực thi, nó sẽ tạo ra kết quả như sau:

|  |
| --- |
| str =  'Hello World!'  n =  2345  d =  2345  un =  [**uint32**](matlab:helpPopup%20uint32)  790  rn =  5.6789e+03  c =  [**uint32**](matlab:helpPopup%20uint32)  5679 |

### Chuyển đổi kiểu dữ liệu

MATLAB cung cấp nhiều hàm khác nhau để chuyển đổi từ kiểu dữ liệu này sang kiểu dữ liệu khác. Bảng sau hiển thị các hàm chuyển đổi kiểu dữ liệu:

|  |  |
| --- | --- |
| **Hàm** | **Mục đích** |
| char | Chuyển thành mảng ký tự (chuỗi) |
| int2str | Chuyển đổi dữ liệu số nguyên thành chuỗi |
| mat2str | Chuyển đổi ma trận thành chuỗi |
| num2str | Chuyển đổi số thành chuỗi |
| str2double | Chuyển đổi chuỗi thành giá trị chính xác kép |
| str2num | Chuyển đổi chuỗi thành số |
| native2unicode | Chuyển đổi byte số thành ký tự Unicode |
| unicode2native | Chuyển đổi các ký tự Unicode thành byte số |
| base2dec | Chuyển đổi chuỗi số cơ số N thành số thập phân |
| bin2dec | Chuyển đổi chuỗi số nhị phân thành số thập phân |
| dec2base | Chuyển đổi số thập phân thành cơ số N trong chuỗi |
| dec2bin | Chuyển đổi số thập phân sang số nhị phân trong chuỗi |
| dec2hex | Chuyển đổi số thập phân thành số thập lục phân trong chuỗi |
| hex2dec | Chuyển đổi chuỗi số thập lục phân thành số thập phân |
| hex2num | Chuyển đổi chuỗi số thập lục phân thành số có độ chính xác kép |
| num2hex | Chuyển đổi đơn và đôi thành chuỗi thập lục phân IEEE |
| cell2mat | Chuyển đổi mảng ô thành mảng số |
| cell2struct | Chuyển đổi mảng ô thành mảng cấu trúc |
| cellstr | Tạo mảng ô của chuỗi từ mảng ký tự |
| mat2cell | Chuyển đổi mảng thành mảng ô với các ô có kích thước khác nhau |
| num2cell | Chuyển đổi mảng thành mảng ô với các ô có kích thước nhất quán |
| struct2cell | Chuyển đổi cấu trúc thành mảng ô |

### Xác định kiểu dữ liệu

MATLAB cung cấp các chức năng khác nhau để xác định kiểu dữ liệu của một biến. Bảng sau cung cấp các hàm để xác định kiểu dữ liệu của một biến:

|  |  |
| --- | --- |
| **Hàm** | **Mục đích** |
| is | phát hiện trạng thái |
| isa | Xác định xem đầu vào có phải là đối tượng của lớp được chỉ định không |
| iscell | Xác định xem đầu vào có phải là mảng ô không |
| iscellstr | Xác định xem đầu vào có phải là mảng ô của chuỗi không |
| ischar | Xác định xem mục có phải là mảng ký tự không |
| isfield | Xác định xem đầu vào có phải là trường mảng cấu trúc không |
| isfloat | Xác định xem đầu vào có phải là mảng dấu phẩy động không |
| ishghandle | Đúng cho Xử lý xử lý đối tượng đồ họa |
| isinteger | Xác định xem đầu vào có phải là mảng số nguyên không |
| isjava | Xác định xem đầu vào có phải là đối tượng Java không |
| islogical | Xác định xem đầu vào có phải là mảng logic không |
| isnumeric | Xác định xem đầu vào có phải là mảng số không |
| isobject | Xác định xem đầu vào có phải là đối tượng MATLAB không |
| isreal | Kiểm tra xem đầu vào có phải là mảng thực không |
| isscalar | Xác định xem đầu vào có vô hướng hay không |
| isstr | Xác định xem đầu vào có phải là mảng ký tự không |
| isstruct | Xác định xem đầu vào có phải là mảng cấu trúc không |
| isvector | Xác định xem đầu vào có phải là vectơ không |
| class | Xác định lớp đối tượng |
| validateattributes | Kiểm tra tính hợp lệ của mảng |
| whos | Liệt kê các biến trong không gian làm việc, với kích thước và loại |

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| x = 3  isinteger(x)  isfloat(x)  isvector(x)  isscalar(x)  isnumeric(x)  x = 23.54  isinteger(x)  isfloat(x)  isvector(x)  isscalar(x)  isnumeric(x)  x = [123]  isinteger(x)  isfloat(x)  isvector(x)  isscalar(x)  x = 'Hello'  isinteger(x)  isfloat(x)  isvector(x)  isscalar(x)  isnumeric(x) |

Khi bạn chạy tệp, nó sẽ tạo ra kết quả sau:

|  |
| --- |
| x =  3  ans =  [**logical**](matlab:helpPopup%20logical)  0  ans =  [**logical**](matlab:helpPopup%20logical)  1  ans =  [**logical**](matlab:helpPopup%20logical)  1  ans =  [**logical**](matlab:helpPopup%20logical)  1  ans =  [**logical**](matlab:helpPopup%20logical)  1  x =  23.5400  ans =  [**logical**](matlab:helpPopup%20logical)  0  ans =  [**logical**](matlab:helpPopup%20logical)  1  ans =  [**logical**](matlab:helpPopup%20logical)  1  ans =  [**logical**](matlab:helpPopup%20logical)  1  ans =  [**logical**](matlab:helpPopup%20logical)  1  x =  123  ans =  [**logical**](matlab:helpPopup%20logical)  0  ans =  [**logical**](matlab:helpPopup%20logical)  1  ans =  [**logical**](matlab:helpPopup%20logical)  1  ans =  [**logical**](matlab:helpPopup%20logical)  1  x =  'Hello'  ans =  [**logical**](matlab:helpPopup%20logical)  0  ans =  [**logical**](matlab:helpPopup%20logical)  0  ans =  [**logical**](matlab:helpPopup%20logical)  1  ans =  [**logical**](matlab:helpPopup%20logical)  0  ans =  [**logical**](matlab:helpPopup%20logical)  0 |

## TOÁN TỬ

Toán tử là một ký hiệu cho biết trình biên dịch thực hiện các thao tác logic hoặc toán học cụ thể. MATLAB được thiết kế để hoạt động chủ yếu trên toàn bộ ma trận và mảng. Do đó, các toán tử trong MATLAB hoạt động cả trên dữ liệu vô hướng và không vô hướng. MATLAB cho phép các loại hoạt động cơ bản sau:

* Toán tử số học
* Toán tử quan hệ
* Toán tử logic
* Toán tử Bitwise
* Toán tử Set

### Toán tử số học

MATLAB cho phép hai loại phép tính số học khác nhau:

* Phép toán số học ma trận
* Các phép toán số học mảng

Các phép toán số học ma trận giống như được định nghĩa trong đại số tuyến tính. Các hoạt động của mảng được thực hiện theo từng phần tử , cả trên mảng một chiều và mảng nhiều chiều.

Toán tử ma trận và toán tử mảng được phân biệt bằng ký hiệu dấu chấm (.). Tuy nhiên, vì phép cộng và phép trừ giống nhau đối với ma trận và mảng, nên toán tử giống nhau cho cả hai trường hợp. Bảng sau đây mô tả ngắn gọn về các toán tử:

|  |  |
| --- | --- |
| **Toán tử** | **Sự miêu tả** |
| + | Phép cộng hoặc phép cộng đơn phương. A+B thêm A và B. A và B phải có cùng kích thước, trừ khi một là vô hướng. Một vô hướng có thể được thêm vào một ma trận có kích thước bất kỳ. |
| - | Phép trừ hoặc phép trừ đơn phương. AB trừ B từ A. A và B phải có cùng kích thước, trừ khi một là vô hướng. Một vô hướng có thể được trừ từ một ma trận có kích thước bất kỳ. |
| \* | Phép nhân ma trận. C = A\*B là tích đại số tuyến tính của ma trận A và B. Chính xác hơn,  Matrix Multiplication  Đối với A và B không vô hướng, số cột của A phải bằng số hàng của B. Một đại lượng vô hướng có thể nhân một ma trận có kích thước bất kỳ. |
| .\* | Phép nhân mảng. A.\*B là tích theo từng phần tử của mảng A và B. A và B phải có cùng kích thước, trừ khi một trong số chúng là vô hướng. |
| / | Dấu gạch chéo hoặc phép chia phải của ma trận. B/A gần giống như B\*inv(A). Chính xác hơn, B/A = (A'\B')'. |
| ./ | Mảng chia phải. A./B là ma trận có các phần tử A(i,j)/B(i,j). A và B phải có cùng kích thước, trừ khi một trong số chúng là vô hướng. |
| \ | Dấu gạch chéo ngược hoặc phép chia trái ma trận. Nếu A là một ma trận vuông, thì A\B gần giống như inv(A)\*B, ngoại trừ nó được tính theo một cách khác. Nếu A là ma trận n-by-n và B là vectơ cột có n thành phần hoặc ma trận có nhiều cột như vậy, thì X = A\B là nghiệm của phương trình AX = B . Một thông báo cảnh báo được hiển thị nếu A được chia tỷ lệ kém hoặc gần như số ít. |
| .\ | Mảng chia trái. A.\B là ma trận có các phần tử B(i,j)/A(i,j). A và B phải có cùng kích thước, trừ khi một trong số chúng là vô hướng. |
| ^ | Sức mạnh ma trận. X^p là X lũy thừa p, nếu p là một đại lượng vô hướng. Nếu p là một số nguyên, thì lũy thừa được tính bằng cách bình phương lặp lại. Nếu số nguyên âm, X được đảo ngược trước. Đối với các giá trị khác của p, phép tính bao gồm các giá trị riêng và vectơ riêng, sao cho nếu [V,D] = eig(X), thì X^p = V\*D.^p/V. |
| .^ | Công suất mảng. A.^B là ma trận có các phần tử A(i,j) lũy thừa B(i,j). A và B phải có cùng kích thước, trừ khi một trong số chúng là vô hướng. |
| ' | Chuyển vị ma trận. A' là phép chuyển vị đại số tuyến tính của A. Đối với ma trận phức, đây là phép chuyển vị liên hợp phức. |
| .' | Chuyển vị mảng. MỘT.' là phép hoán vị mảng của A. Đối với ma trận phức tạp, điều này không liên quan đến phép chia. |

**Ví dụ:** Các ví dụ sau đây cho thấy việc sử dụng các toán tử số học trên dữ liệu vô hướng.

Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| a = 10;  b = 20;  c = a + b  d = a - b  e = a \* b  f = a / b  g = a \ b  x = 7;  y = 3;  z = x ^ y |

Khi bạn chạy tệp, nó sẽ tạo ra kết quả sau:

|  |
| --- |
| c =  30 d =  -10 e =  200 f =  0.5000 g =  2 z =  343 |

#### Hàm cho phép toán số học

Ngoài các toán tử số học đã đề cập ở trên, MATLAB cung cấp các lệnh/hàm sau được sử dụng cho mục đích tương tự:

|  |  |
| --- | --- |
| **Hàm** | **Sự miêu tả** |
| uplus(a) | cộng một ngôi; tăng thêm một lượng a |
| plus (a,b) | Thêm; trả về a + b |
| uminus(a) | Phép trừ đơn phương; giảm đi một lượng a |
| minus(a, b) | Dấu trừ; trả về a - b |
| times(a, b) | Nhân mảng; trả về a.\*b |
| mtimes(a, b) | Phép nhân ma trận; trả về a\* b |
| rdivide(a, b) | Chia mảng bên phải; trả về a ./ b |
| ldivide(a, b) | Chia mảng trái; trả về a.\ b |
| mrdivide(A, B) | Giải hệ phương trình tuyến tính xA = B cho x |
| mldivide(A, B) | Giải hệ phương trình tuyến tính Ax = B với x |
| power(a, b) | Công suất mảng; trả về a.^b |
| mpower(a, b) | Ma trận quyền lực; trả về a^b |
| cumprod(A) | Tích lũy sản phẩm; trả về một mảng có cùng kích thước với mảng A chứa sản phẩm tích lũy.  Nếu A là một vectơ, thì cumprod(A) trả về một vectơ chứa tích lũy của các phần tử của A.  Nếu A là một ma trận, thì cumprod(A) trả về một ma trận chứa các tích lũy cho mỗi cột của A.  Nếu A là một mảng nhiều chiều, thì cumprod(A) hoạt động dọc theo chiều không phải đơn thứ nhất. |
| cumprod(A, dim) | Trả về tích lũy dọc theo thứ nguyên *dim* . |
| cumsum(A) | Tổng tích lũy; trả về một mảng A chứa tổng tích lũy.  Nếu A là một vectơ, thì cumsum(A) trả về một vectơ chứa tích lũy tổng các phần tử của A.  Nếu A là một ma trận, thì cumsum(A) trả về một ma trận chứa các tổng tích lũy cho mỗi cột của A.  Nếu A là một mảng nhiều chiều, thì cumsum(A) hoạt động dọc theo chiều không phải đơn thứ nhất. |
| cumsum(A, dim) | trả về tổng tích lũy của các phần tử dọc theo thứ nguyên dim . |
| diff(X) | Ví phân và đạo hàm gần đúng; tính toán sự khác biệt giữa các phần tử liền kề của X.  Nếu X là một vectơ, thì diff(X) trả về một vectơ, một phần tử ngắn hơn X, về sự khác biệt giữa các phần tử liền kề: [X(2)-X(1) X(3)-X(2) ... X (n)-X(n-1)]  Nếu X là một ma trận, thì diff(X) trả về một ma trận chênh lệch hàng: [X(2:m,:)-X(1:m- 1,:)] |
| diff(X,n) | Áp dụng đệ quy diff n lần, dẫn đến sự khác biệt thứ n. |
| diff(X,n,dim) | Đây là hàm khác biệt thứ n được tính dọc theo thứ nguyên được chỉ định bởi độ mờ vô hướng. Nếu thứ tự n bằng hoặc vượt quá độ dài của thứ nguyên dim, diff trả về một mảng trống. |
| prod(A) | Tích các phần tử của mảng; trả về tích của các phần tử mảng của A. Nếu A là một vectơ, thì prod(A) trả về tích của các phần tử.  Nếu A là một ma trận khác rỗng, thì prod(A) coi các cột của A là các vectơ và trả về một vectơ hàng chứa các tích của mỗi cột.  Nếu A là một ma trận 0-by-0 trống, prod(A) trả về 1.  Nếu A là một mảng nhiều chiều, thì prod(A) hoạt động dọc theo chiều không phải đơn thứ nhất và trả về một mảng các tích. Kích thước của kích thước này giảm xuống còn 1 trong khi kích thước của tất cả các kích thước khác vẫn giữ nguyên.  Hàm prod tính toán và trả về B là đơn nếu đầu vào A là đơn. Đối với tất cả các loại dữ liệu số và logic khác, prod tính toán và trả về B gấp đôi |
| prod(A,dim) | Trả về các sản phẩm dọc theo chiều mờ. Ví dụ: nếu A là ma trận, prod(A,2) là vectơ cột chứa tích của mỗi hàng. |
| prod(\_\_\_,datatype) | nhân vào và trả về một mảng trong lớp được chỉ định bởi kiểu dữ liệu. |
| sum(A) | Tổng các phần tử của mảng; trả về tổng dọc theo các kích thước khác nhau của một mảng. Nếu A là dấu phẩy động, nghĩa là dấu phẩy động kép hoặc đơn, thì B được tích lũy nguyên bản, nghĩa là cùng lớp với A và B có cùng lớp với A. Nếu A không phải là dấu phẩy động, B được tích lũy kép và B có lớp gấp đôi.  Nếu A là một vectơ, sum(A) trả về tổng của các phần tử.  Nếu A là ma trận, tổng(A) coi các cột của A là vectơ, trả về một hàng  véc tơ tổng của mỗi cột.  Nếu A là một mảng nhiều chiều, tổng(A) coi các giá trị dọc theo chiều không phải đơn lẻ đầu tiên là các vectơ, trả về một mảng các vectơ hàng. |
| sum(A,dim) | Tính tổng theo chiều của A được chỉ định bởi scalar dim . |
| sum(..., 'double') sum(..., dim,'double') | Thực hiện phép cộng với độ chính xác kép và trả về kết quả kiểu double, ngay cả khi A có kiểu dữ liệu đơn hoặc kiểu dữ liệu số nguyên. Đây là giá trị mặc định cho kiểu dữ liệu số nguyên. |
| sum(..., 'native') sum(..., dim,'native') | Thực hiện các phép cộng trong kiểu dữ liệu gốc của A và trả về câu trả lời có cùng kiểu dữ liệu. Đây là mặc định cho đơn và đôi. |
| ceil(A) | Tròn về hướng dương vô cùng; làm tròn các phần tử của A thành số nguyên gần nhất lớn hơn hoặc bằng A. |
| fix(A) | Làm tròn về không |
| floor(A) | Làm tròn về phía âm vô cực; làm tròn các phần tử của A thành số nguyên gần nhất nhỏ hơn hoặc bằng A. |
| idivide(a, b)  idivide(a, b,'fix') | Phép chia số nguyên với tùy chọn làm tròn; giống như a./b ngoại trừ các thương số phân số được làm tròn về 0 đến các số nguyên gần nhất. |
| idivide(a, b, 'round') | Các thương phân số được làm tròn đến số nguyên gần nhất. |
| idivide(A, B, 'floor') | Các thương phân số được làm tròn về phía âm vô cực đến các số nguyên gần nhất. |
| idivide(A, B, 'ceil') | Các thương phân số được làm tròn về phía vô cực đến các số nguyên gần nhất. |
| mod (X,Y) | Mô đun sau khi chia; trả về X - n.\*Y trong đó n = sàn(X./Y). Nếu Y không phải là số nguyên và thương số X./Y nằm trong lỗi làm tròn của một số nguyên, thì n là số nguyên đó. Các đầu vào X và Y phải là các mảng thực có cùng kích thước hoặc vô hướng thực (với điều kiện là Y ~=0).  Xin lưu ý:  mod(X,0) là X mod(X,X) là 0  mod(X,Y) cho X~=Y và Y~=0 có cùng dấu với Y |
| rem (X,Y) | Phần còn lại sau khi phân chia; trả về X - n.\*Y trong đó n = fix(X./Y). Nếu Y không phải là số nguyên và thương số X./Y nằm trong lỗi làm tròn của một số nguyên, thì n là số nguyên đó. Các đầu vào X và Y phải là các mảng thực có cùng kích thước hoặc vô hướng thực (với điều kiện là Y ~=0).  Xin lưu ý rằng:  rem(X,0) là NaN rem(X,X) cho X~=0 là 0  rem(X,Y) cho X~=Y và Y~=0 có cùng dấu với X. |
| vòng(X) | Làm tròn đến số nguyên gần nhất; làm tròn các phần tử của X đến số nguyên gần nhất. Các phần tử dương có phần phân số là 0,5 làm tròn đến số nguyên dương gần nhất. Các phần tử âm có phần phân số là -0,5 làm tròn xuống số nguyên âm gần nhất. |

### Toán tử quan hệ

Các toán tử quan hệ cũng có thể hoạt động trên cả dữ liệu vô hướng và không vô hướng. Các toán tử quan hệ cho mảng thực hiện phép so sánh từng phần tử giữa hai mảng và trả về một mảng logic có cùng kích thước, với các phần tử được đặt thành logic 1 (true) trong đó quan hệ là đúng và các phần tử được đặt thành logic 0 (false) trong đó không phải.

Bảng sau đây cho thấy các toán tử quan hệ có sẵn trong MATLAB:

|  |  |
| --- | --- |
| **Toán tử** | **Miêu tả** |
| < | Ít hơn |
| <= | Ít hơn hoặc bằng |
| > | Lớn hơn |
| >= | Lớn hơn hoặc bằng |
| = = | Tương đương với |
| ~= | không bằng |

**Ví dụ:** Tạo một tệp script và nhập mã sau:

|  |
| --- |
| a = 100;  b = 200;  if(a >= b)  max = a  else  max = b  end |

Khi bạn chạy tệp, nó sẽ tạo ra kết quả sau:

|  |
| --- |
| max =  200 |

Ngoài các toán tử quan hệ nêu trên, MATLAB còn cung cấp các lệnh/hàm sau được sử dụng cho cùng mục đích:

|  |  |
| --- | --- |
| **Hàm** | **Miêu tả** |
| eq(a, b) | Kiểm tra xem a có bằng b không |
| ge(a, b) | Kiểm tra xem a có lớn hơn hoặc bằng b không |
| gt(a, b) | Kiểm tra xem a có lớn hơn b không |
| le(a, b) | Kiểm tra xem a có nhỏ hơn hoặc bằng b không |
| lt(a, b) | Kiểm tra xem a có nhỏ hơn b không |
| ne(a, b) | Kiểm tra xem a có bằng b không |
| isequal | Kiểm tra mảng cho bình đẳng |
| isequaln | Kiểm tra sự bằng nhau của các mảng, coi các giá trị NaN là bằng nhau |

**Ví dụ:** Tạo một tệp script và nhập mã sau:

|  |
| --- |
| a = 100;  b = 200;  if(ge(a,b))  max = a  else  max = b  end  % Thay đổi 2 giá trị a và b  a = 340;  b = 520;  if(le(a, b))  disp(' a is either less than or equal to b')  else  disp(' a is greater than b')  end |

Khi bạn chạy tệp, nó sẽ tạo ra kết quả sau:

|  |
| --- |
| max =  200  a is either less than or equal to b |

### Toán tử logic

MATLAB cung cấp hai loại toán tử logic và hàm:

* Phần tử khôn ngoan - các toán tử này hoạt động trên các phần tử logic tương ứng mảng.
* Đoản mạch - các toán tử này hoạt động trên vô hướng, logic biểu thức.

Các toán tử logic theo từng phần tử vận hành từng phần tử trên các mảng logic. Các ký hiệu &, |, và ~ là các toán tử mảng logic AND, OR và NOT.

Toán tử logic ngắn mạch cho phép đoản mạch trên các phép toán logic. Các ký hiệu && và || là các toán tử logic ngắn mạch AND và OR.

**Ví dụ:** Tạo một tệp script và nhập mã sau:

|  |
| --- |
| a =5;  b =20;  if( a && b )  disp('Line 1 - Condition is true');  end  if( a || b )  disp('Line 2 - Condition is true');  end  % lets change the value of a and b a =0;  b =10;  if( a && b )  disp('Line 3 - Condition is true');  else  disp('Line 3 - Condition is not true');  end  if(~(a && b))  disp('Line 4 - Condition is true');  end |

Khi bạn chạy tệp, nó sẽ tạo ra kết quả sau:

|  |
| --- |
| Line 1 - Condition is true  Line 2 - Condition is true  Line 3 - Condition is not true  Line 4 - Condition is true |

#### Hàm cho phép toán logic

Ngoài các toán tử logic đã đề cập ở trên, MATLAB còn cung cấp các lệnh hoặc hàm sau được sử dụng cho cùng một mục đích:

|  |  |
| --- | --- |
| **Hàm** | **Sự miêu tả** |
| and(A, B) | Tìm AND logic của đầu vào mảng hoặc vô hướng; thực hiện logic AND của tất cả các mảng đầu vào A, B, v.v. và trả về một mảng chứa các phần tử được đặt thành logic 1 (đúng) hoặc logic 0 (sai). Một phần tử của mảng đầu ra được đặt thành 1 nếu tất cả các mảng đầu vào chứa một phần tử khác 0 tại cùng một vị trí mảng đó. Mặt khác, phần tử đó được đặt thành 0. |
| not(A) | Tìm logic KHÔNG của mảng hoặc đầu vào vô hướng; thực hiện logic NOT của mảng đầu vào A và trả về một mảng chứa các phần tử được đặt thành logic 1 (true) hoặc logic 0 (false). Một phần tử của mảng đầu ra được đặt thành 1 nếu mảng đầu vào chứa phần tử có giá trị bằng 0 tại cùng vị trí mảng đó. Mặt khác, phần tử đó được đặt thành 0. |
| or(A, B) | Tìm logic OR của đầu vào mảng hoặc vô hướng; thực hiện OR logic của tất cả các mảng đầu vào A, B, vân vân. và lợi nhuận một mảng có chứa yếu tố bộ đến một trong hai hợp lý 1 (thật) hoặc hợp lý 0 (sai). Một phần tử của mảng đầu ra được đặt thành 1 nếu bất kỳ mảng đầu vào nào chứa phần tử khác 0 tại cùng vị trí mảng đó. Mặt khác, phần tử đó được đặt thành 0. |
| xor(A, B) | Loại trừ logic-HOẶC; thực hiện một phép toán OR loại trừ trên các phần tử tương ứng của mảng A và B. Phần tử kết quả C(i,j,...) là logic đúng (1) nếu A(i,j,...) hoặc B(i,j,...), nhưng không phải cả hai, là khác không. |
| all(A) | Xác định xem tất cả các phần tử mảng của mảng A là khác không hay đúng.  Nếu A là một vectơ, all(A) trả về logic 1 (true) nếu tất cả các phần tử khác không và trả về logic 0 (false) nếu một hoặc nhiều phần tử bằng 0.  Nếu A là một ma trận khác rỗng, all(A) coi các cột của A là vectơ, trả về một vectơ hàng của logic 1 và 0.  Nếu A là một ma trận 0-by-0 trống, all(A) trả về logic 1 (true).  Nếu A là một mảng nhiều chiều, thì all(A) hoạt động dọc theo chiều không phải đơn thứ nhất và trả về một mảng các giá trị logic. Kích thước của kích thước này giảm xuống còn 1 trong khi kích thước của tất cả các kích thước khác vẫn giữ nguyên. |
| all(A, dim) | Kiểm tra dọc theo kích thước của A được chỉ định bởi scalar dim . |
| any(A) | Xác định xem có bất kỳ phần tử mảng nào khác không; kiểm tra xem bất kỳ phần tử nào dọc theo các chiều khác nhau của một mảng là một số khác 0 hay là logic 1 (đúng). Hàm bất kỳ bỏ qua các mục là NaN (Không phải là Số).  Nếu A là một vectơ, any(A) trả về logic 1 (true) nếu bất kỳ phần tử nào của A là số khác 0 hoặc logic 1 (true) và trả về logic 0 (false) nếu tất cả các phần tử đều bằng không.  Nếu A là một ma trận khác rỗng, bất kỳ(A) coi các cột của A là vectơ, trả về một vectơ hàng của logic 1 và 0.  Nếu A là một ma trận 0-by-0 trống, any(A) trả về logic 0 (sai).  Nếu A là một mảng nhiều chiều, thì any(A) hoạt động dọc theo chiều không phải đơn thứ nhất và trả về một mảng các giá trị logic. Kích thước của kích thước này giảm xuống còn 1 trong khi kích thước của tất cả các kích thước khác vẫn giữ nguyên. |
| any(A,dim) | Kiểm tra dọc theo kích thước của A được chỉ định bởi scalar dim . |
| False | Logic 0 (sai) |
| false(n) | là ma trận n-by-n của các số 0 logic |
| false(m, n) | là một ma trận m-by-n của các số 0 logic. |
| false(m, n, p, ...) | là một mảng m-by-n-by-p-by-... các số 0 logic. |
| false(size(A)) | là một mảng các số 0 logic có cùng kích thước với mảng A. |
| false(...,'like',p) | là một mảng các số 0 logic cùng kiểu dữ liệu và độ thưa thớt như mảng logic p. |
| ind = find(X) | Tìm chỉ số và giá trị của các phần tử khác không; định vị tất cả các phần tử khác không của mảng X và trả về các chỉ số tuyến tính của các phần tử đó trong một vectơ. Nếu X là một vectơ hàng, thì vectơ trả về là một vectơ hàng; nếu không, nó trả về một vectơ cột. Nếu X không chứa phần tử khác 0 hoặc là một mảng trống, thì một mảng trống sẽ được trả về. |
| ind = find(X, k)  ind = find(X, k, 'first') | Trả về nhiều nhất k chỉ số đầu tiên tương ứng với các mục khác không của X. k phải là một số nguyên dương, nhưng nó có thể thuộc bất kỳ loại dữ liệu số nào. |
| tất cả(A) | Xác định xem tất cả các phần tử mảng của mảng A là khác không hay đúng.  Nếu A là một vectơ, all(A) trả về logic 1 (true) nếu tất cả các phần tử khác không và trả về logic 0 (false) nếu một hoặc nhiều phần tử bằng 0.  Nếu A là một ma trận khác rỗng, all(A) coi các cột của A là vectơ, trả về một vectơ hàng của logic 1 và 0.  Nếu A là một ma trận 0-by-0 trống, all(A) trả về logic 1 (true).  Nếu A là một mảng nhiều chiều, thì all(A) hoạt động dọc theo chiều không phải đơn thứ nhất và trả về một mảng các giá trị logic. Kích thước của kích thước này giảm xuống còn 1 trong khi kích thước của tất cả các kích thước khác vẫn giữ nguyên. |
| tất cả (A, mờ) | Kiểm tra dọc theo kích thước của A được chỉ định bởi scalar dim . |
| bất kỳ (A) | Xác định xem có bất kỳ phần tử mảng nào khác không; kiểm tra xem bất kỳ phần tử nào dọc theo các chiều khác nhau của một mảng là một số khác 0 hay là logic 1 (đúng). Hàm bất kỳ bỏ qua các mục là NaN (Không phải là Số).  Nếu A là một vectơ, any(A) trả về logic 1 (true) nếu bất kỳ phần tử nào của A là số khác 0 hoặc logic 1 (true) và trả về logic 0 (false) nếu tất cả các phần tử đều bằng không.  Nếu A là một ma trận khác rỗng, bất kỳ(A) coi các cột của A là vectơ, trả về một vectơ hàng của logic 1 và 0.  Nếu A là một ma trận 0-by-0 trống, any(A) trả về logic 0 (sai).  Nếu A là một mảng nhiều chiều, thì any(A) hoạt động dọc theo chiều không phải đơn thứ nhất và trả về một mảng các giá trị logic. Kích thước của kích thước này giảm xuống còn 1 trong khi kích thước của tất cả các kích thước khác vẫn giữ nguyên. |
| bất kỳ (A, mờ) | Kiểm tra dọc theo kích thước của A được chỉ định bởi scalar dim . |
| Sai | Logic 0 (sai) |
| sai (n) | là ma trận n-by-n của các số 0 logic |
| sai (m, n) | là một ma trận m-by-n của các số 0 logic. |
| sai(m, n, p, ...) | là một mảng m-by-n-by-p-by-... các số 0 logic. |
| sai(kích thước(A)) | là một mảng các số 0 logic có cùng kích thước với mảng A. |
| sai (...,'thích',p) | là một mảng các số 0 logic cùng kiểu dữ liệu và độ thưa thớt như mảng logic p. |
| ind = tìm(X) | Tìm chỉ số và giá trị của các phần tử khác không; định vị tất cả các phần tử khác không của mảng X và trả về các chỉ số tuyến tính của các phần tử đó trong một vectơ. Nếu X là một vectơ hàng, thì vectơ trả về là một vectơ hàng; nếu không, nó trả về một vectơ cột. Nếu X không chứa phần tử khác 0 hoặc là một mảng trống, thì một mảng trống sẽ được trả về. |
| ind = tìm(X, k) | Trả về nhiều nhất k chỉ số đầu tiên tương ứng với các mục khác không của X. k phải là một số nguyên dương, nhưng nó có thể thuộc bất kỳ loại dữ liệu số nào. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| ind = find(X, k, 'đầu tiên') |  |
| ind = find(X, k, 'cuối cùng') | trả về tối đa k chỉ số cuối cùng tương ứng với các mục nhập khác không của X. |
| [hàng,col] = tìm(X,...) | Trả về chỉ số hàng và cột của các mục nhập khác không trong ma trận X. Cú pháp này đặc biệt hữu ích khi làm việc với các ma trận thưa thớt. Nếu X là một mảng N chiều với N > 2, col chứa các chỉ số tuyến tính cho các cột. |
| [hàng,col,v] = tìm(X, ...) | Trả về một vectơ cột hoặc hàng v của các mục khác không trong X, cũng như các chỉ số hàng và cột. Nếu X là biểu thức logic thì v là mảng logic. Đầu ra v chứa các phần tử khác 0 của mảng logic thu được bằng cách đánh giá biểu thức X. |
| đẳng logic(A) | Xác định xem đầu vào có phải là mảng logic hay không; trả về true nếu A là một mảng logic và sai nếu ngược lại. Nó cũng trả về true nếu A là một thể hiện của một lớp dẫn xuất từ lớp logic. |
| logic(A) | Chuyển đổi các giá trị số thành logic; trả về một mảng có thể được sử dụng để lập chỉ mục logic hoặc kiểm tra logic. |
| Thật | Logic 1 (đúng) |
| đúng (n) | là một ma trận n-by-n của những cái logic. |
| đúng (m, n) | là một ma trận m-by-n của những cái logic. |
| true(m, n, p, ...) | là một mảng m-by-n-by-p-by-... của logic. |
| đúng (kích thước (A)) | là một mảng logic có cùng kích thước với mảng A. |
| true(...,'like', p) | là mảng các logic cùng kiểu dữ liệu và độ thưa thớt như mảng logic p. |
| ind = find(X, k, 'đầu tiên') |  |
| ind = find(X, k, 'cuối cùng') | trả về tối đa k chỉ số cuối cùng tương ứng với các mục nhập khác không của X. |
| [hàng,col] = tìm(X,...) | Trả về chỉ số hàng và cột của các mục nhập khác không trong ma trận X. Cú pháp này đặc biệt hữu ích khi làm việc với các ma trận thưa thớt. Nếu X là một mảng N chiều với N > 2, col chứa các chỉ số tuyến tính cho các cột. |
| [hàng,col,v] = tìm(X, ...) | Trả về một vectơ cột hoặc hàng v của các mục khác không trong X, cũng như các chỉ số hàng và cột. Nếu X là biểu thức logic thì v là mảng logic. Đầu ra v chứa các phần tử khác 0 của mảng logic thu được bằng cách đánh giá biểu thức X. |
| đẳng logic(A) | Xác định xem đầu vào có phải là mảng logic hay không; trả về true nếu A là một mảng logic và sai nếu ngược lại. Nó cũng trả về true nếu A là một thể hiện của một lớp dẫn xuất từ lớp logic. |
| logic(A) | Chuyển đổi các giá trị số thành logic; trả về một mảng có thể được sử dụng để lập chỉ mục logic hoặc kiểm tra logic. |
| Thật | Logic 1 (đúng) |
| đúng (n) | là một ma trận n-by-n của những cái logic. |
| đúng (m, n) | là một ma trận m-by-n của những cái logic. |

### Toán tử Bitwise

Toán tử bitwise hoạt động trên các bit và thực hiện thao tác từng bit. Các bảng chân lý cho &, |, và ^ như sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P | Hỏi | p&q | tr| q | p^q |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |

Giả sử nếu A = 60; và B = 13; Bây giờ ở định dạng nhị phân, chúng sẽ giống như sau: A = 0011 1100

B = 0000 1101

A&B = 0000 1100

A|B = 0011 1101

A^B = 0011 0001

~A = 1100 0011

MATLAB cung cấp các chức năng khác nhau cho các phép toán theo bit như phép toán 'bitwise and', 'bitwise or' và 'bitwise not', phép toán dịch chuyển, vân vân.

Bảng sau đây cho thấy các hoạt động bitwise thường được sử dụng:

|  |  |
| --- | --- |
| **Hàm** | **Sự miêu tả** |
| bitand(a, b) | Bit-khôn ngoan AND của số nguyên a và b |
| bitcmp(a) | Bổ sung bit-khôn ngoan của một |
| bitget(a,pos) | Nhận bit tại vị trí xác định pos , trong mảng số nguyên a |
| bitor(a, b) | Bit-khôn ngoan OR của số nguyên a và b |
| bitset(a, pos) | Đặt bit tại vị trí vị trí cụ thể của một |
| dịch chuyển bit(a, k) | Trả về a bị dịch chuyển sang trái k bit, tương đương với việc nhân với 2 k . Các giá trị âm của k tương ứng với việc dịch chuyển các bit sang phải hoặc chia cho 2 |k| và làm tròn đến số nguyên gần nhất về phía âm vô hạn. Bất kỳ bit tràn nào đều bị cắt bớt. |
| bitxor(a, b) | XOR bit-khôn ngoan của số nguyên a và b |
| hoán đổi byte | Hoán đổi thứ tự byte |

**Ví dụ:** Tạo một tệp script và nhập mã sau:

|  |
| --- |
|  |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

### Toán tử Set

MATLAB cung cấp các chức năng khác nhau cho các thao tác tập hợp, như hợp, giao và kiểm tra tư cách thành víên của tập hợp, v.v. Bảng sau đây trình bày một số thao tác tập hợp thường được sử dụng:

|  |  |
| --- | --- |
| **Hàm** | **Sự miêu tả** |
| cắt nhau(A,B) | Đặt giao điểm của hai mảng; trả về các giá trị chung cho cả A và B. Các giá trị được trả về theo thứ tự đã sắp xếp. |
| cắt nhau(A,B,'hàng') | Coi mỗi hàng của A và mỗi hàng của B là các thực thể đơn lẻ và trả về các hàng chung cho cả A và B. Các hàng của ma trận trả về được sắp xếp theo thứ tự. |
| thành víên(A,B) | Trả về một mảng có cùng kích thước với A, chứa 1 (đúng) khi các phần tử của A được tìm thấy trong B. Ở những nơi khác, nó trả về 0 (sai). |
| ismember(A,B,'hàng') | Coi mỗi hàng của A và mỗi hàng của B là các thực thể đơn lẻ và trả về một vectơ chứa 1 (đúng) trong đó các hàng của ma trận A cũng là các hàng của B. Ở những nơi khác, nó trả về 0 (sai). |
| được sắp xếp (A) | Trả về logic 1 (true) nếu các phần tử của A được sắp xếp theo thứ tự và logic 0 (sai) ngược lại. Đầu vào A có thể là một vectơ hoặc một mảng ô N-by-1 hoặc 1-by-N của các chuỗi. A được coi là được sắp xếp nếu A và đầu ra của sort(A) là công bằng. |
| issorted(A, 'hàng') | Trả về logic 1 (đúng) nếu các hàng của ma trận hai chiều A được sắp xếp theo thứ tự và logic 0 (sai) nếu ngược lại. Ma trận A được coi là sắp xếp nếu A và đầu ra của các phép sắp xếp(A) bằng nhau. |
| setdiff(A,B) | Đặt sự khác biệt của hai mảng; trả về các giá trị trong A không có trong B. Các giá trị trong mảng được trả về theo thứ tự đã sắp xếp. |
| setdiff(A,B,'hàng') | Xử lý mỗi hàng của A và mỗi hàng của B là các thực thể đơn lẻ và trả về các hàng từ A không có trong B. Các hàng của ma trận trả về được sắp xếp theo thứ tự.  Tùy chọn 'hàng' không hỗ trợ mảng ô. |
| Setxor | Đặt OR độc quyền của hai mảng |
| liên hiệp | Đặt liên kết của hai mảng |
| Duy nhất | Các giá trị duy nhất trong mảng |

**Ví dụ:** Tạo một tệp script và nhập mã sau:

|  |
| --- |
| a =[723141591282435] |

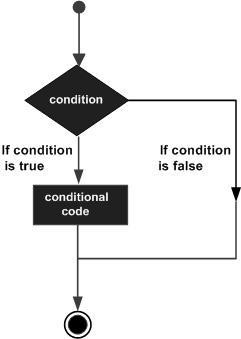
Khi bạn chạy tệp, nó sẽ tạo ra kết quả sau:

|  |
| --- |
|  |

## CẤU TRÚC RẼ NHÁNH

Cấu trúc rẽ nhánh yêu cầu lập trình viên chỉ định một hoặc nhiều điều kiện để chương trình đánh giá hoặc được kiểm tra. Nếu điều kiện là đúng thì một hoặc nhiều câu lệnh sẽ được thực hiện, nếu điều kiện là sai thì các câu lệnh khác sẽ được thực thi.

Sau đây là dạng chung của một cấu trúc ra quyết định điển hình được tìm thấy trong hầu hết các ngôn ngữ lập trình:

**Hình 10:** Lưu đồ câu lệnh rẽ nhánh

MATLAB cung cấp các câu lệnh điều kiện sau đây.

|  |  |
| --- | --- |
| **Câu lệnh** | **Sự miêu tả** |
| if ... end | Câu lệnh if ... end bao gồm một biểu thức boolean theo sau bởi một hoặc nhiều câu lệnh. |
| if...else...end | Một câu lệnh if có thể được theo sau bởi một câu lệnh khác tùy chọn, mà thực thi khi biểu thức boolean là sai. |
| If...elseif...elseif  ...else...end | Một câu lệnh if có thể được theo sau bởi một (hoặc nhiều) câu lệnh elseif... và một câu lệnh khác tùy chọn , rất hữu ích để kiểm tra các điều kiện khác nhau. |
| if lồng nhau | Bạn có thể sử dụng một câu lệnh if hoặc elseif bên trong (các) câu lệnh if hoặc elseif khác . |
| switch | Một câu lệnh chuyển đổi cho phép một biến được kiểm tra sự bằng nhau đối với một danh sách các giá trị. |
| switch lồng nhau | nested switch statements |

### Câu lệnh if … end

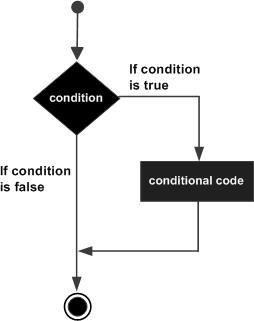
Một câu lệnh if ... end bao gồm một câu lệnh if và một biểu thức boolean theo sau bởi một hoặc nhiều câu lệnh. Nó được phân định bởi câu lệnh kết thúc .

**Cú pháp**

|  |
| --- |
| if <expression>  % statement(s) will execute if the boolean expression is true  <statements>  end |

Nếu biểu thức đánh giá là đúng, thì khối mã bên trong câu lệnh if sẽ được thực thi. Nếu biểu thức đánh giá là sai, thì bộ mã đầu tiên sau câu lệnh kết thúc sẽ được thực thi.

**Lưu đồ:**



Hình:

**Ví dụ:** Tạo một tệp script và nhập mã sau.

|  |
| --- |
| a = 10;  % Kiểm tra điều kiện sử dụng câu lệnh if  if a < 20  % Nếu điều kiện là đúng thì in câu lệnh tiếp theo  fprintf('a is less than 20\n' );  end  fprintf('value of a is: %d\n', a); |

Khi bạn chạy tập tin, nó sẽ hiển thị như sau kết quả:

|  |
| --- |
| a is less than 20  value of a is: 10 |

### Câu lệnh if … else … end

Một câu lệnh if có thể được theo sau bởi một câu lệnh khác tùy chọn, câu lệnh này sẽ thực thi khi biểu thức sai.

**Cú pháp:**

Cú pháp của câu lệnh if...else trong MATLAB là:

|  |
| --- |
| if <expression>  % statement(s) will execute if the boolean expression is true  <statement(s)>  else  <statement(s)>  % statement(s) will execute if the boolean expression is false  end |

Nếu biểu thức boolean đánh giá là đúng, thì khối mã if sẽ được thực thi, nếu không thì khối mã khác sẽ được thực thi.

**Lưu đồ:**

Diagram

Description automatically generated

**Hình 1:**

**Ví dụ:** Tạo một tệp script và nhập mã sau:

|  |
| --- |
| a = 100;  % check the boolean condition  if a < 20  % if condition is true then print the following  fprintf('a is less than 20\n' );  else  % if condition is false then print the following  fprintf('a is not less than 20\n' );  end  fprintf('value of a is: %d\n', a); |

Khi đoạn mã trên được biên dịch và thực thi, nó sẽ tạo ra kết quả như sau:

|  |
| --- |
| a is not less than 20  value of a is: 100 |

### Câu lệnh if…elseif…elseif…else…end

Một câu lệnh if có thể được theo sau bởi một (hoặc nhiều) câu lệnh elseif... và một câu lệnh khác tùy chọn , điều này rất hữu ích để kiểm tra các câu lệnh khác nhau. điều kiện, tình trạng, trạng thái.

Khi sử dụng câu lệnh if...elseif...else, có một số điểm cần lưu ý:

Một if có thể có số 0 hoặc của người khác và nó phải đến sau bất kỳ của người khác.

Một if có thể có từ 0 đến nhiều otherif và chúng phải đứng trước khác.

Khi một cái khác nếu thành công, không cái nào khác của cái khác hoặc cái khác sẽ là thử nghiệm .

**Cú pháp:**

|  |
| --- |
| if <expression 1>  % Executes when the expression 1 is true  <statement(s)>  elseif <expression 2>  % Executes when the boolean expression 2 is true  <statement(s)>  elseif <expression 3>  % Executes when the boolean expression 3 is true  <statement(s)>  else  % Executes when the none of the above condition is true  <statement(s)>  end |

**Ví dụ:** Tạo một tệp script và nhập mã sau vào đó:

|  |
| --- |
| a = 100;  %check the boolean condition  if a == 10  % if condition is true then print the following  fprintf('Value of a is 10\n' );  elseif( a == 20 )  % if else if condition is true  fprintf('Value of a is 20\n' );  elseif a == 30  % if else if condition is true  fprintf('Value of a is 30\n' );  else  % if none of the conditions is true  fprintf('None of the values are matching\n');  fprintf('Exact value of a is: %d\n', a );  end |

Khi đoạn mã trên được biên dịch và thực thi, nó sẽ tạo ra kết quả như sau:

|  |
| --- |
| None of the values are matching  Exact value of a is: 100 |

### Câu lệnh if lồng nhau

Trong MATLAB, việc lồng các câu lệnh if-else luôn hợp pháp, nghĩa là bạn có thể sử dụng một câu lệnh if hoặc elseif bên trong (các) câu lệnh if hoặc elseif khác.

**Cú pháp:**

|  |
| --- |
| if <expression 1>  % Executes when the boolean expression 1 is true  <statement(s)>  if <expression 2>  % Executes when the boolean expression 2 is true  <statement(s)>  end  end |

Bạn có thể lồng elseif...else theo cách tương tự như cách bạn đã lồng câu lệnh if.

**Ví dụ:** Tạo một tệp script và nhập mã sau vào đó:

|  |
| --- |
| a = 100;  b = 200;  % check the boolean condition  if( a == 100 )  % if condition is true then check the following  if( b == 200 )  % if condition is true then print the following  fprintf('Value of a is 100 and b is 200\n' );  end  end  fprintf('Exact value of a is : %d\n', a );  fprintf('Exact value of b is : %d\n', b ); |

Khi bạn chạy tệp, nó sẽ hiển thị:

|  |
| --- |
| Value of a is 100 and b is 200  Exact value of a is : 100  Exact value of b is : 200 |

### Câu lệnh switch

Một khối chuyển đổi thực hiện một cách có điều kiện một tập hợp các câu lệnh từ một số lựa chọn. Mỗi lựa chọn được bao phủ bởi một trường hợp bản tường trình.

Một switch\_expression được đánh giá là một chuỗi hoặc vô hướng.

Một case\_expression được đánh giá là một chuỗi vô hướng, một chuỗi hoặc một mảng ô gồm các chuỗi hoặc chuỗi vô hướng. Khối chuyển đổi kiểm tra từng trường hợp cho đến khi một trong các trường hợp là đúng. Một trường hợp đúng khi:

* Đối với số, eq(case\_expression,switch\_expression) .
* Đối với chuỗi, strcmp(case\_expression,switch\_expression) .
* Đối với các đối tượng hỗ trợ chức năng eq, eq(case\_expression,switch\_expression) .
* Đối với một mảng ô case\_expression, ít nhất một trong các phần tử của mảng ô khớp với switch\_expression, như đã xác định ở trên cho các số, chuỗi và các đối tượng.

Khi một trường hợp đúng, MATLAB thực hiện các câu lệnh tương ứng và sau đó thoát khỏi khối chuyển đổi. Khối khác là tùy chọn và chỉ thực hiện khi không có trường hợp nào đúng.

**Cú pháp**

|  |
| --- |
| switch <switch\_expression>  case <case\_expression>  <statements>  case <case\_expression>  <statements>  ...  ...  otherwise  <statements>  end |

**Ví dụ:** Tạo một tệp script và nhập mã sau vào đó:

|  |
| --- |
| grade = 'B';  switch(grade)  case 'A'  fprintf('Excellent!\n' );  case 'B'  fprintf('Well done\n' );  case 'C'  fprintf('Well done\n' );  case 'D'  fprintf('You passed\n' );  case 'F'  fprintf('Better try again\n' );  otherwise  fprintf('Invalid grade\n' );  end |

Khi bạn chạy tệp, nó sẽ hiển thị:

|  |
| --- |
| Well done |

### Câu lệnh switch lồng nhau

Có thể có một công tắc như một phần của chuỗi câu lệnh của một công tắc bên ngoài. Ngay cả khi các hằng số trường hợp của công tắc bên trong và bên ngoài chứa các giá trị chung, sẽ không có xung đột nào phát sinh.

**Cú pháp:**

Cú pháp của câu lệnh switch lồng nhau như sau:

|  |
| --- |
| switch(ch1)  case 'A'  fprintf('This A is part of outer switch');  switch(ch2)  case 'A'  fprintf('This A is part of inner switch' );  case 'B'  fprintf('This B is part of inner switch' );  end  case 'B'  fprintf('This B is part of outer switch' );  end |

**Ví dụ:** Tạo một tệp script và nhập mã sau vào đó:

|  |
| --- |
| a = 100;  b = 200;  switch(a)  case 100  fprintf('This is part of outer switch %d\n', a );  switch(b)  case 200  fprintf('This is part of inner switch %d\n', a );  end  end  fprintf('Exact value of a is : %d\n', a );  fprintf('Exact value of b is : %d\n', b ); |

Khi bạn chạy tệp, nó sẽ hiển thị:

|  |
| --- |
| This is part of outer switch 100 This is part of inner switch 100 Exact value of a is : 100 Exact value of b is : 200 |

## VÒNG LẶP

Có thể xảy ra tình huống khi bạn cần thực thi một khối mã nhiều lần. Nói chung, các câu lệnh được thực hiện tuần tự. Câu lệnh đầu tiên trong một hàm được thực thi trước, tiếp theo là câu lệnh thứ hai, và như thế.

Các ngôn ngữ lập trình cung cấp các cấu trúc điều khiển khác nhau cho phép thực hiện các đường dẫn phức tạp hơn.

Một câu lệnh lặp cho phép chúng ta thực hiện một câu lệnh hoặc một nhóm câu lệnh nhiều lần và sau đây là dạng chung của một câu lệnh lặp trong hầu hết các ngôn ngữ lập trình:

Diagram

Description automatically generated

Hình:

MATLAB cung cấp các loại vòng lặp sau để xử lý các yêu cầu lặp.

|  |  |
| --- | --- |
| **Câu lệnh** | **Sự miêu tả** |
| while | Lặp lại một câu lệnh hoặc một nhóm câu lệnh trong khi một điều kiện nhất định là đúng. Nó kiểm tra điều kiện trước khi thực hiện thân vòng lặp. |
| for | Thực hiện một chuỗi các câu lệnh nhiều lần và viết tắt mã quản lý biến vòng lặp. |
| Vòng lặp lồng nhau | Bạn có thể sử dụng một hoặc nhiều vòng lặp bên trong bất kỳ vòng lặp nào khác. |

### Vòng lặp while

Vòng lặp while thực hiện lặp đi lặp lại các câu lệnh trong khi điều kiện là đúng.

**Cú pháp:**

|  |
| --- |
| while <expression>  <statements>  end |

Vòng lặp while lặp đi lặp lại việc thực thi (các) câu lệnh của chương trình miễn là biểu thức vẫn đúng.

Một biểu thức là đúng khi kết quả khác rỗng và chứa tất cả các phần tử khác 0 (số logic hoặc số thực). Mặt khác, biểu thức là sai.

**Ví dụ:** Tạo một tệp script và nhập mã sau:

|  |
| --- |
| a = 10;  % while loop execution while( a < 20 )  fprintf('value of a: %d\n', a);  a = a + 1;  end |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| value of a: 10 value of a: 11 value of a: 12 value of a: 13 value of a: 14 value of a: 15 value of a: 16 value of a: 17 value of a: 18 value of a: 19 |

### Vòng lặp for

Vòng lặp for là một cấu trúc điều khiển lặp lại cho phép bạn viết một vòng lặp cần thực hiện một số lần cụ thể một cách hiệu quả.

**Cú pháp:**

|  |
| --- |
| for index = values  <program statements>  ...  end |

Các giá trị là một trong các dạng sau:

|  |  |
| --- | --- |
| **Định dạng** | **Sự miêu tả** |
| initval:endval | tăng biến chỉ mục từ initval đến endval lên 1 và lặp lại việc thực hiện các câu lệnh chương trình cho đến khi chỉ mục lớn hơn endval . |
| initval:bước:endval | tăng chỉ mục theo bước giá trị trên mỗi lần lặp hoặc giảm khi bước âm. |
| valArray | chỉ mục vectơ cột từ các cột tiếp theo của mảng valArray trên mỗi lần lặp. Ví dụ: trong lần lặp đầu tiên, index = valArray(:,1). Vòng lặp thực hiện tối đa n lần, trong đó n là số cột của valArray , được cho bởi numel(valArray, 1, :). ValArray đầu vào có thể thuộc bất kỳ loại dữ liệu MATLAB nào, bao gồm một chuỗi, mảng ô hoặc cấu trúc. |

**Ví dụ:** Tạo một tệp script và nhập mã sau:

|  |
| --- |
| for a = 10:20  fprintf('value of a: %d\n', a);  end |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| value of a: 10 value of a: 11 value of a: 12 value of a: 13 value of a: 14 value of a: 15 value of a: 16 value of a: 17 value of a: 18 value of a: 19 value of a: 20 |

**Ví dụ:** Tạo một tệp script và nhập mã sau:

|  |
| --- |
| for a = 1.0: -0.1: 0.0  disp(a) end |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| 1  0.9000  0.8000  0.7000  0.6000  0.5000  0.4000  0.3000  0.2000  0.1000  0 |

**Ví dụ:**Tạo một tệp script và nhập mã sau:

|  |
| --- |
| for a = [24,18,17,23,28]  disp(a)  end |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| 24  18  17  23  28 |

### Vòng lồng nhau

MATLAB cho phép sử dụng một vòng lặp bên trong một vòng lặp khác. Phần sau đây cho thấy một vài ví dụ để minh họa khái niệm.

Cú pháp

Cú pháp của một câu lệnh lặp for lồng nhau trong MATLAB như sau:

|  |
| --- |
| for m = 1:j  for n = 1:k  <statements>;  end  end |

Cú pháp của một câu lệnh vòng lặp while lồng nhau trong MATLAB như sau:

|  |
| --- |
| while <expression1>  while <expression2>  <statements>  end  end |

**Ví dụ:** Hãy để chúng tôi sử dụng một vòng lặp for lồng nhau để hiển thị tất cả các số nguyên tố từ 1 đến 100.

Tạo một tệp script và nhập mã sau:

|  |
| --- |
| for i=2:100  for j=2:100 if(~mod(i,j))  break; % if factor found, not prime  end end  if(j > (i/j))  fprintf('%d is prime\n', i);  end  end |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| 1. is prime 2. is prime   5 is prime  7 is prime  11 is prime  13 is prime  17 is prime  19 is prime  23 is prime  29 is prime  31 is prime  37 is prime  41 is prime  43 is prime  47 is prime  53 is prime  59 is prime  61 is prime  67 is prime  71 is prime  73 is prime  79 is prime  83 is prime  89 is prime  97 is prime |

### Câu lệnh điều khiển vòng lặp

Các câu lệnh điều khiển vòng lặp thay đổi quá trình thực thi so với trình tự thông thường của nó. Khi thực thi rời khỏi một phạm ví, tất cả các đối tượng tự động được tạo trong phạm viđó sẽ bị hủy.

MATLAB hỗ trợ các câu lệnh điều khiển sau.

|  |  |
| --- | --- |
| **Kiểm soát** | **Sự miêu tả** |
| break | Kết thúc câu lệnh vòng lặp và chuyển thực thi sang câu lệnh ngay sau vòng lặp. |
| continue | Làm cho vòng lặp bỏ qua phần còn lại của phần thân và ngay lập tức kiểm tra lại tình trạng của nó trước khi lặp lại. |

#### Câu lệnh break

Câu lệnh break kết thúc việc thực hiện vòng lặp **for** hoặc **while**. Các câu lệnh trong vòng lặp xuất hiện sau câu lệnh break không được thực thi.

Trong các vòng lặp lồng nhau, break chỉ thoát khỏi vòng lặp mà nó xảy ra. Kiểm soát chuyển đến câu lệnh sau khi kết thúc vòng lặp đó.

Sơ đồ dòng chảy:

Diagram

Description automatically generated

Hình:

**Ví dụ:**Tạo một tệp script và nhập mã sau:

|  |
| --- |
| a = 10;  % while loop execution while (a < 20 )  fprintf('value of a: %d\n', a); a = a+1;  if( a > 15)  % terminate the loop using break statement break;  end |

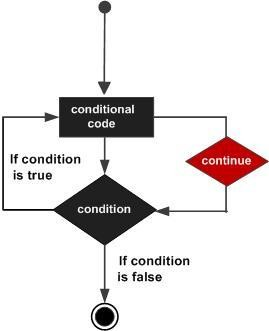
Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| value of a: 10 value of a: 11 value of a: 12 value of a: 13 value of a: 14 value of a: 15 |

#### Câu lệnh continue

Câu lệnh continue được sử dụng để chuyển điều khiển sang lần lặp tiếp theo của vòng lặp for hoặc while. Câu lệnh continue trong MATLAB hoạt động giống như câu lệnh break. Tuy nhiên, thay vì buộc kết thúc, "continue" buộc lần lặp tiếp theo của vòng lặp diễn ra, bỏ qua bất kỳ mã nào ở giữa.

Sơ đồ dòng chảy:



Hình:

**Ví dụ:** Tạo một tệp script và nhập mã sau:

|  |
| --- |
| a = 10;  %while loop execution while a < 20  if a == 15  % skip the iteration a = a + 1; continue;  end  fprintf('value of a: %d\n', a); a = a + 1;  end |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| value of a: 10 value of a: 11 value of a: 12 value of a: 13 value of a: 14 value of a: 16 value of a: 17 value of a: 18 value of a: 19 |

## VECTƠ

Một vector là một mảng số một chiều. MATLAB cho phép tạo hai loại vectơ:

* Vectơ hàng
* Vectơ cột

### Vectơ hàng

Vectơ hàng được tạo bằng cách đặt tập hợp các phần tử trong dấu ngoặc vuông, sử dụng dấu cách hoặc dấu phẩy để phân tách các phần tử.

|  |
| --- |
| r =[7891011] |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| r = Columns 1 through 4 7 8 9 10 Column 5 11 |

### Vectơ cột

Vectơ cột được tạo bằng cách đặt tập hợp các phần tử trong dấu ngoặc vuông, sử dụng dấu chấm phẩy để phân tách các phần tử.

|  |
| --- |
| c =[7;8;9;10;11] |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| c =  7  8  9  10  11 |

### Tham khảo các phần tử của một véc tơ

Bạn có thể tham chiếu một hoặc nhiều phần tử của vectơ theo nhiều cách. Thành phần thứ i của vectơ v được gọi là v(i).

**Ví dụ:**

|  |
| --- |
| v =[1;2;3;4;5;6]; % creating a column vector of 6 elements  v(3) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans = 3 |

Khi bạn tham chiếu một vectơ bằng dấu hai chấm, chẳng hạn như v(:), tất cả các thành phần của vectơ được liệt kê.

|  |
| --- |
| v =[1;2;3;4;5;6]; % creating a column vector of 6 elements  v(:) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans = 1  2  3  4  5  6 |

MATLAB cho phép bạn chọn một loạt các phần tử từ một vectơ.

**Ví dụ:** chúng ta hãy tạo một vectơ hàng rv gồm 9 phần tử, sau đó chúng ta sẽ tham chiếu các phần tử từ 3 đến 7 bằng cách viết **rv(3:7)** và tạo một vectơ mới có tên phụ\_rv.

|  |
| --- |
| rv =[123456789];  sub\_rv = rv(3:7) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| sub\_rv = 3 4 5 6 7 |

### Phép toán vectơ

* Trong phần này, chúng ta hãy thảo luận về các phép toán vectơ sau:
* Cộng và Trừ củavectơ
* Phép nhân vô hướng củavectơ
* Chuyển vị của mộtvéc tơ
* Đang bổ sungvectơ
* Độ lớn của mộtvéc tơ
* Véc Tơ ChấmSản phẩm
* Các vectơ cách đều nhauyếu tố

#### Phép cộng và phép trừ vectơ

Bạn có thể cộng hoặc trừ hai vectơ. Cả hai vectơ toán hạng phải cùng loại và có cùng số phần tử.

**Ví dụ:**Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| A = [7, 11, 15, 23, 9];  B = [2, 5, 13, 16, 20]; C = A + B;  D = A - B;  disp(C);  disp(D); |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| 9 16 28 39 29 5 6 2 7 -11 |

#### Phép nhân vô hướng của vectơ

Khi bạn nhân một vectơ với một số, đây được gọi là phép nhân vô hướng. Phép nhân vô hướng tạo ra một vectơ mới cùng loại với mỗi phần tử của vectơ ban đầu được nhân với số.

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| v = [ 12 34 10 8];  m = 5 \* v |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| m = 60 170 50 40 |

Xin lưu ý rằng bạn có thể thực hiện tất cả các thao tác vô hướng trên vectơ. Ví dụ: bạn có thể cộng, trừ và chia một vectơ với một đại lượng vô hướng.

#### Chuyển vị của một vectơ

Thao tác hoán vị thay đổi một vectơ cột thành một vectơ hàng và ngược lại. Hoạt động chuyển đổi được thể hiện bằng một trích dẫn đơn (').

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| r = [ 1 2 3 4 ];  tr = r';  v = [1;2;3;4];  tv = v';  disp(tr); disp(tv); |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| 1 2 3 4  1 2 3 4 |

#### Nối Vector

MATLAB cho phép bạn nối các vectơ lại với nhau để tạo các vectơ mới.

Nếu bạn có hai vectơ hàng r1 và r2 với n và m số phần tử, để tạo một vectơ hàng r gồm n cộng với m phần tử, bằng cách nối thêm các vectơ này, bạn viết:

|  |
| --- |
| r = [r1,r2] |

Bạn cũng có thể tạo một ma trận r bằng cách nối hai vectơ này, vectơ r2, sẽ là hàng thứ hai của ma trận:

|  |
| --- |
| r = [r1;r2] |

Tuy nhiên, để làm được điều này, cả hai vectơ phải có cùng số phần tử.

Tương tự, bạn có thể nối hai vectơ cột c1 và c2 với n và m số phần tử. Để tạo một vectơ cột c gồm n cộng m phần tử, bằng cách nối thêm các vectơ này, bạn viết:

|  |
| --- |
| c = [c1; c2] |

Bạn cũng có thể tạo một ma trận c bằng cách nối thêm hai vectơ này; vectơ c2 sẽ là cột thứ hai của ma rận:

|  |
| --- |
| c = [c1, c2] |

Tuy nhiên, để làm được điều này, cả hai vectơ phải có cùng số phần tử.

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| r1 = [ 1 2 3 4 ];  r2 = [5 6 7 8 ];  r = [r1,r2] rMat = [r1;r2]  c1 = [ 1; 2; 3; 4 ];  c2 = [5; 6; 7; 8 ];  c = [c1; c2] cMat = [c1,c2] |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

### Độ lớn của một vectơ

Độ lớn của vectơ v với các phần tử v1, v2, v3, …, vn, được cho bởi phương trình:

|  |
| --- |
| |v| = √(v1 2 + v2 2 + v3 2 + … + vn 2 ) |

Bạn cần thực hiện các bước sau để tính độ lớn của vectơ:

1. Lấy tích của vectơ với chính nó, sử dụng phép nhân mảng (.\*). Điều này tạo ra một vectơ sv, có các phần tử là bình phương của các phần tử của vectơ v.

|  |
| --- |
| sv = v.\*v; |

2. Sử dụng hàm sum để lấy tổng bình phương các phần tử của vectơ v. Đây còn được gọi là tích vô hướng của vectơ v.

|  |
| --- |
| dp = sum (sv); |

3. Sử dụng các câu hỏi chức năng đến được các Quảng trường nguồn gốc của các tổng mà Là Mà còn các kích cỡ của các véc tơ v.

|  |
| --- |
| mag = sqrt(s); |

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| v = [1: 2: 20];  sv = v.\* v; %the vector with elements  % as square of v's elements  dp = sum(sv); % sum of squares -- the dot product mag = sqrt(dp); % magnitude  disp('Magnitude:'); disp(mag); |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| Magnitude:  36.4692 |

### Vector Chấm Sản Phẩm

Tích vô hướng của hai vectơ a = (a1, a2, …, an) và b = (b1, b2, …, bn) được cho bởi:

ab = ∑(ai.bi)

Tích vô hướng của hai vectơ a và b được tính bằng hàm **dot**.

|  |
| --- |
| dot(a, b); |

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| v1 = [2 3 4];  v2 = [1 2 3];  dp = dot(v1, v2);  disp('Dot Product:'); disp(dp); |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| Dot Product:  20 |

### Vectơ với các phần tử cách đều nhau

MATLAB cho phép bạn tạo một véc-tơ với các phần tử cách đều nhau.

Để tạo một vectơ v với phần tử đầu f, phần tử cuối l và hiệu giữa các phần tử là một số thực n bất kỳ, ta viết:

|  |
| --- |
| v = [f : n : l] |

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| v = [1: 2: 20];  sqv = v.^2; disp(v);disp(sqv); |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

## MA TRẬN

Một ma trận là một mảng số hai chiều.

Trong MATLAB, bạn tạo một ma trận bằng cách nhập các phần tử trong mỗi hàng dưới dạng các số được phân cách bằng dấu phẩy hoặc dấu cách và sử dụng dấu chấm phẩy để đánh dấu phần cuối của mỗi hàng.

**Ví dụ**: Chúng ta hãy tạo ma trận a 4x5:

|  |
| --- |
| a = [1 2 3 4 5 ; 2 3 4 5 6 ; 3 4 5 6 7 ; 4 5 6 7 8] |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| a =   1 2 3 4 5  2 3 4 5 6  3 4 5 6 7  4 5 6 7 8 |

### Tham khảo các phần tử của một ma trận

Để tham chiếu đến một phần tử ở hàng thứ m và cột thứ n của ma trận mx , ta viết:

|  |
| --- |
| mx(m, n); |

**Ví dụ:** Để tham chiếu đến phần tử ở hàng thứ 2 và cột thứ 5 của ma trận a.

|  |
| --- |
| a = [1 2 3 4 5 ; 2 3 4 5 6 ; 3 4 5 6 7 ; 4 5 6 7 8];  a(2,5) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans =   6 |

Để tham chiếu tất cả các phần tử trong cột thứ m, chúng ta gõ a(:,m).

**Ví dụ:** Tạo một vectơ cột v, từ các phần tử của hàng thứ 4 của ma trận a.

|  |
| --- |
| a = [1 2 3 4 5 ; 2 3 4 5 6 ; 3 4 5 6 7 ; 4 5 6 7 8];  v = a(:,4) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| v =   4  5  6  7 |

Bạn cũng có thể chọn các phần tử trong các cột thứ m đến thứ n, vì điều này chúng tôi viết:

|  |
| --- |
| a(:,m:n) |

**Ví dụ:** Hãy để chúng tôi tạo một ma trận nhỏ hơn lấy các phần tử từ cột thứ hai và thứ ba:

|  |
| --- |
| a = [1 2 3 4 5 ; 2 3 4 5 6 ; 3 4 5 6 7 ; 4 5 6 7 8];  a(:,2:3) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans =   2 3  3 4  4 5  5 6 |

Theo cách tương tự, bạn có thể tạo một ma trận con lấy một phần con của ma trận.

|  |
| --- |
| a = [1 2 3 4 5 ; 2 3 4 5 6 ; 3 4 5 6 7 ; 4 5 6 7 8];  a(:,2:3) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans =   2 3  3 4  4 5  5 6 |

Theo cách tương tự, bạn có thể tạo một ma trận con lấy một phần con của ma trận.

**Ví dụ:** Chúng ta hãy tạo một ma trận con sa lấy phần con bên trong của a:

|  |
| --- |
| 3 4 5  4 5 6 |

Để làm điều này, hãy viết:

|  |
| --- |
| a = [1 2 3 4 5 ; 2 3 4 5 6 ; 3 4 5 6 7 ; 4 5 6 7 8];  sa = a(2:3,2:4) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| sa =   3 4 5  4 5 6 |

### Xóa một hàng hoặc một cột trong ma trận

Bạn có thể xóa toàn bộ hàng hoặc cột của ma trận bằng cách gán một tập trống các dấu ngoặc vuông [] cho hàng hoặc cột đó. Về cơ bản, [] biểu thị một mảng trống.

**Ví dụ:** Chúng ta hãy xóa hàng thứ tư của a:

|  |
| --- |
| a = [1 2 3 4 5 ; 2 3 4 5 6 ; 3 4 5 6 7 ; 4 5 6 7 8]  a(4,:) = [] |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| a =   1 2 3 4 5  2 3 4 5 6  3 4 5 6 7 |

**Ví dụ:** Xóa cột thứ năm của a:

|  |
| --- |
| a = [1 2 3 4 5 ; 2 3 4 5 6 ; 3 4 5 6 7 ; 4 5 6 7 8]  a(:,5) = [] |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| a =   1 2 3 4  2 3 4 5  3 4 5 6  4 5 6 7 |

**Ví dụ:** Tạo ma trận 3x3 m, sau đó chúng ta sẽ sao chép hàng 2 và 3 của ma trận này hai lần để tạo ma trận 4 nhân 3.

Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| a = [ 1 2 3 ; 4 5 6 ; 7 8 9 ];  new\_mat = a([2,3,2,3],:) |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| new\_mat =   4 5 6  7 8 9  4 5 6  7 8 9 |

**Ví dụ:** Tạo ma trận 3x3 m, sau đó chúng ta sẽ sao chép hàng 2 và 3, cột 1 và 2 của ma trận này.

Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| a = [ 1 2 3 ; 4 5 6 ; 7 8 9 ];  new\_mat = a([2,3],1:2) |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| new\_mat =   4 5  7 8 |

### Toán tử ma trận

Trong phần này, chúng ta hãy thảo luận về các hoạt động ma trận cơ bản và thường được sử dụng sau đây:

* Cộng và Trừ củama trận
* Phép chia ma trận
* Hoạt động vô hướng của ma trận
* Chuyển vị của mộtma trận
* Nốima trận
* Phép nhân ma trận
* Định thức của mộtma trận
* Nghịch đảo của ma trận

#### Phép cộng và phép trừ ma trận

Bạn có thể cộng hoặc trừ các ma trận. Cả hai ma trận toán hạng phải có cùng số hàng và số cột.

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| a = [ 1 2 3 ; 4 5 6 ; 7 8 9 ];  b = [ 7 5 6 ; 2 0 8 ; 5 7 1 ];  c = a + b  d = a - b |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| c =   8 7 9  6 5 14  12 15 10   d =   -6 -3 -3  2 5 -2  2 1 8 |

#### Phép chia ma trận

Bạn có thể chia hai ma trận bằng toán tử chia trái (\) hoặc phải (/). Cả hai ma trận toán hạng phải có cùng số hàng và số cột.

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| a = [ 1 2 3 ; 4 5 6 ; 7 8 9 ];  b = [ 7 5 6 ; 2 0 8 ; 5 7 1 ];  c = a / b  d = a \ b |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| c =  -0.5254 0.6864 0.6610  -0.4237 0.9407 1.0169  -0.3220 1.1949 1.3729  Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND = 1.541976e-18.  > In [**Test1**](matlab:matlab.internal.language.introspective.errorDocCallback('Test1',%20'/MATLAB%20Drive/Test1.m',%204)) ([line 4](matlab:%20opentoline('/MATLAB%20Drive/Test1.m',4,0)))    d =  1.0e+17 \*  -0.3603 -0.5404 0.4053  0.7206 1.0809 -0.8106  -0.3603 -0.5404 0.4053 |

#### Hoạt động vô hướng của ma trận

Khi bạn cộng, trừ, nhân hoặc chia một ma trận cho một số, đây được gọi là phép toán vô hướng.

Các phép toán vô hướng tạo ra một ma trận mới có cùng số hàng và cột với mỗi phần tử của ma trận ban đầu được thêm vào, trừ đi, nhân hoặc chia cho một số.

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| a = [ 10 12 23 ; 14 8 6 ; 27 8 9 ];  b = 2;  c = a + b  d = a - b  e = a \* b  f = a / b |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| c =   12 14 25  16 10 8  29 10 11   d =   8 10 21  12 6 4  25 6 7   e =   20 24 46  28 16 12  54 16 18   f =   5.0000 6.0000 11.5000  7.0000 4.0000 3.0000  13.5000 4.0000 4.5000 |

#### Hoán vị của một ma trận

Thao tác hoán vị sẽ chuyển đổi các hàng và cột trong một ma trận. Nó được biểu thị bằng một trích dẫn đơn (').

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| a = [ 10 12 23 ; 14 8 6 ; 27 8 9 ]  b = a' |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| a =   10 12 23  14 8 6  27 8 9   b =   10 14 27  12 8 8  23 6 9 |

#### Ma trận nối

Bạn có thể nối hai ma trận để tạo ma trận lớn hơn. Cặp dấu ngoặc vuông '[]' là toán tử nối.

MATLAB cho phép hai kiểu nối:

* Nằm ngang nối
* Thẳng đứng nối

Khi bạn nối hai ma trận bằng cách tách chúng bằng dấu phẩy, chúng chỉ được nối theo chiều ngang. Nó được gọi là nối ngang.

Ngoài ra, nếu bạn nối hai ma trận bằng cách tách chúng bằng dấu chấm phẩy, chúng sẽ được nối theo chiều dọc. Nó được gọi là dọc nối.

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| a = [ 10 12 23 ; 14 8 6; 27 8 9]  b = [ 12 31 45 ; 8 0 -9; 45 2 11]  c = [a, b]  d = [a; b] |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| c =   10 12 23 12 31 45  14 8 6 8 0 -9  27 8 9 45 2 11   d =   10 12 23  14 8 6  27 8 9  12 31 45  8 0 -9  45 2 11 |

#### Phép nhân ma trận

Xét hai ma trận A và B. Nếu A là ma trận mxn và B là ma trận anxp, chúng có thể được nhân với nhau để tạo ra ma trận mxn C. Phép nhân ma trận chỉ có thể thực hiện được nếu số cột n trong A bằng số cột hàng n trong B.

Trong phép nhân ma trận, các phần tử của các hàng trong ma trận thứ nhất được nhân với các cột tương ứng trong ma trận thứ hai.

Thứ (i, j) trong ma trận kết quả C là tổng tích của các phần tử ở hàng thứ i của ma trận thứ nhất với phần tử tương ứng ở cột thứ j của ma trận thứ hai.

Trong MATLAB, phép nhân ma trận được thực hiện bằng toán tử \*.

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| a = [ 1 2 3 ; 2 3 4 ; 1 2 5 ]  b = [ 2 1 3 ; 5 0 -2 ; 2 3 -1 ]  prod = a \* b |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| a =   1 2 3  2 3 4  1 2 5   b =   2 1 3  5 0 -2  2 3 -1   prod =   18 10 -4  27 14 -4  22 16 -6 |

#### Định thức của một ma trận

Định thức của ma trận được tính bằng hàm det của MATLAB. Định thức của ma trận A được cho bởi det(A).

**Ví dụ:** Tạo một tệp script với đoạn mã sau:

|  |
| --- |
| a = [ 1 2 3 ; 2 3 4 ; 1 2 5 ]  det(a) |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| a =   1 2 3  2 3 4  1 2 5   ans =   -2 |

#### Nghịch đảo của ma trận

Ma trận nghịch đảo của A được ký hiệu là A −1 sao cho có mối quan hệ sau:

|  |
| --- |
| AA−1 = A−1A = 1 |

Ma trận nghịch đảo không phải lúc nào cũng tồn tại. Nếu định thức của ma trận bằng không, thì nghịch đảo không tồn tại và ma trận là số ít .

Trong MATLAB, nghịch đảo của ma trận được tính bằng hàm inv . Nghịch đảo của ma trận A được cho bởi inv(A).

**Ví dụ:** Tạo một tệp script và nhập mã sau:

|  |
| --- |
| a = [ 1 2 3; 2 3 4; 1 2 5]  inv(a) |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| a =   1 2 3  2 3 4  1 2 5   ans =   -3.5000 2.0000 0.5000  3.0000 -1.0000 -1.0000  -0.5000 0 0.5000 |

## MẢNG

Trong MATLAB tất cả các biến thuộc mọi kiểu dữ liệu đều là mảng nhiều chiều. Một vectơ là một mảng một chiều và một ma trận là một mảng hai chiều.

Chúng ta đã thảo luận về vectơ và ma trận. Trong chương này, chúng ta sẽ thảo luận về mảng nhiều chiều. Tuy nhiên, trước đó, chúng ta hãy thảo luận về một số loại mảng đặc biệt.

### Mảng đặc biệt trong MATLAB

Trong phần này, chúng ta sẽ thảo luận về một số hàm tạo ra một số mảng đặc biệt. Đối với tất cả các chức năng này, một đối số duy nhất tạo ra một mảng hình vuông, đối số kép tạo ra mảng hình chữ nhật.

**Ví dụ:** Hàm **zeros()** tạo một mảng gồm tất cả các số không.

|  |
| --- |
| zeros(5) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans =   0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0 0 |

**Ví dụ:** Hàm **ones()** tạo một mảng gồm tất cả những cái.

|  |
| --- |
| ones(4,3) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans =   1 1 1  1 1 1  1 1 1  1 1 1 |

**Ví dụ:** Hàm **eye()** tạo ma trận nhận dạng.

|  |
| --- |
| eye(4) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans =   1 0 0 0  0 1 0 0  0 0 1 0  0 0 0 1 |

**Ví dụ:** Hàm **rand()** tạo một mảng các số ngẫu nhiên phân bố đều trên (0,1).

|  |
| --- |
| rand(3,5) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans =   0.8147 0.9134 0.2785 0.9649 0.9572  0.9058 0.6324 0.5469 0.1576 0.4854  0.1270 0.0975 0.9575 0.9706 0.8003 |

#### Một hình vuông ma thuật

Một hình vuông kỳ diệu là một hình vuông tạo ra tổng bằng nhau, khi các phần tử của nó được cộng theo hàng, theo cột hoặc theo đường chéo.

Hàm magic() tạo một mảng hình vuông ma thuật. Nó nhận một đối số duy nhất đưa ra kích thước của hình vuông. Đối số phải là một số vô hướng lớn hơn hoặc bằng 3.

|  |
| --- |
| magic(4) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans =   16 2 3 13  5 11 10 8  9 7 6 12  4 14 15 1 |

### Mảng đa chiều

Một mảng có nhiều hơn hai chiều được gọi là mảng nhiều chiều trong MATLAB. Mảng nhiều chiều trong MATLAB là một phần mở rộng của ma trận hai chiều thông thường.

Nói chung để tạo một mảng nhiều chiều, trước tiên chúng ta tạo một mảng hai chiều và mở rộng nó. Ví dụ, hãy tạo một mảng hai chiều a.

|  |
| --- |
| a = [ 7 9 5 ; 6 1 9 ; 4 3 2 ] |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| a =   7 9 5  6 1 9  4 3 2 |

Mảng a là mảng 3 nhân 3; chúng ta có thể thêm thứ nguyên thứ ba vào a, bằng cách cung cấp các giá trị như:

|  |
| --- |
| a(:,:,2) = [ 1 2 3 ; 4 5 6 ; 7 8 9 ] |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| a(:,:,1) =   0 0 0  0 0 0  0 0 0   a(:,:,2) =   1 2 3  4 5 6  7 8 9 |

Chúng ta cũng có thể tạo các mảng nhiều chiều bằng cách sử dụng các hàm **one()**, **zeros()** hoặc **rand()**.

**Ví dụ:**

|  |
| --- |
| b = rand(4,3,2) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| b(:,:,1) =   0.7094 0.6551 0.9597  0.7547 0.1626 0.3404  0.2760 0.1190 0.5853  0.6797 0.4984 0.2238   b(:,:,2) =   0.7513 0.8909 0.1493  0.2551 0.9593 0.2575  0.5060 0.5472 0.8407  0.6991 0.1386 0.2543 |

Chúng ta cũng có thể sử dụng hàm cat() để xây dựng các mảng nhiều chiều. Nó nối một danh sách các mảng dọc theo một chiều xác định:

Cú pháp của hàm cat() là:

|  |
| --- |
| B = cat(dim, A1, A2...) |

Trong đó:

* B là mảng mới tạo
* A1 , A2 , ... là các mảng được nối liền
* dim là kích thước dọc theo đó để nối các mảng

**Ví dụ:** Tạo một tệp script và nhập đoạn mã sau vào đó:

|  |
| --- |
| a = [ 9 8 7 ; 6 5 4 ; 3 2 1 ];  b = [ 1 2 3 ; 4 5 6 ; 7 8 9 ];  c = cat(3, a, b,[ 2 3 1 ; 4 7 8 ; 3 9 0 ]) |

Khi bạn chạy tệp, nó sẽ hiển thị:

|  |
| --- |
| c(:,:,1) =   9 8 7  6 5 4  3 2 1   c(:,:,2) =   1 2 3  4 5 6  7 8 9   c(:,:,3) =   2 3 1  4 7 8  3 9 0 |

### Hàm mảng

MATLAB cung cấp các hàm sau để sắp xếp, xoay, hoán vị, định hình lại hoặc dịch chuyển nội dung mảng.

|  |  |
| --- | --- |
| **Hàm** | **Mục đích** |
| Length | Độ dài của vectơ hoặc kích thước mảng lớn nhất |
| Ndims | Số kích thước mảng |
| Numel | Số phần tử mảng |
| Size | Kích thước mảng |
| iscolumn | Xác định xem đầu vào có phải là vectơ cột không |
| isempty | Xác định xem mảng có trống không |
| ismatrix | Xác định xem đầu vào có phải là ma trận hay không |
| isrow | Xác định xem đầu vào có phải là vectơ hàng không |
| isscalar | Xác định xem đầu vào có vô hướng hay không |
| isvector | Xác định xem đầu vào có phải là vectơ không |
| blkdiag | Xây dựng ma trận đường chéo khối từ các đối số đầu vào |
| circshift | Dịch chuyển mảng tròn |
| ctranspose | Chuyển vị liên hợp phức tạp |
| diag | Ma trận đường chéo và đường chéo của ma trận |
| flipdim | Lật mảng dọc theo kích thước được chỉ định |
| fliplr | Lật ma trận trái sang phải |
| flipud | Lật ma trận lên xuống |
| ipermute | Kích thước hoán vị nghịch đảo của mảng ND |
| permute | Sắp xếp lại kích thước của mảng ND |
| repmat | Sao chép và gạch mảng |
| reshape | định hình lại mảng |
| rot90 | Xoay ma trận 90 độ |
| shiftdim | Kích thước thay đổi |
| issorted | Xác định xem các phần tử của tập hợp có được sắp xếp theo thứ tự không |
| sort | Sắp xếp các phần tử mảng tăng dần hoặc giảm dần |
| sortrows | Sắp xếp các hàng theo thứ tự tăng dần |
| squeeze | Xóa kích thước đơn lẻ |
| transpose | chuyển vị |
| vectorize | Vector hóa biểu thức |

### Các ví dụ sau đây minh họa một số chức năng được đề cập ở trên.

#### Chiều dài, Kích thước và Số phần tử

Tạo một tệp script và nhập đoạn mã sau vào nó:

|  |
| --- |
| x =[7.1,3.4,7.2,28/4,3.6,17,9.4,8.9];  length(x)% length of x vector y = rand(3,4,5,2);  ndims(y)%no of dimensions in array y  s =['Zara','Nuha','Shamim','Riz','Shadab']; numel(s)%no of elements in s |

Khi bạn chạy tập tin, nó sẽ hiển thị như sau kết quả:

|  |
| --- |
| ans = 8  ans = 4  ans = 23 |

#### Dịch chuyển vòng tròn các phần tử mảng:

Tạo một tệp script và nhập đoạn mã sau vào đó:

|  |
| --- |
| a =[123;456;789]% the original array a  b = circshift(a,1)% circular shift first dimension values down by1.  c = circshift(a,[1-1])% circular shift first dimension values % down by1  %and second dimension values to the left %by1. |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

#### Sắp xếp mảng

Tạo một tệp script và nhập đoạn mã sau vào nó:

|  |
| --- |
| v =[2345129501917]% horizonal vector sort(v)%sorting v  m =[264;539;201]% two dimensional array sort(m,1)% sorting m along the row sort(m,2)% sorting m along the column |

Khi bạn chạy tập tin, nó sẽ hiển thị như sau kết quả:

|  |
| --- |
|  |

#### Mảng ô

Mảng ô là mảng các ô được lập chỉ mục trong đó mỗi ô có thể lưu trữ một mảng có kích thước và kiểu dữ liệu khác nhau.

Hàm **cell** được sử dụng để tạo một mảng ô. Cú pháp cho hàm ô là:

|  |
| --- |
| C = cell(dim) |

Trong khi:

* C là tế bào mảng;
* dim là một số nguyên vô hướng hoặc vectơ của các số nguyên xác định kích thước của mảng ô C;
* dim1, ... , dimN là các số nguyên vô hướng xác định kích thước của C;
* obj là một trong những tiếp theo:
* mảng Java hoặc vật
* Mảng .NET kiểu System.String hoặc Hệ thống.Object

**Ví dụ:** Tạo một tệp script và nhập đoạn mã sau vào nó:

|  |
| --- |
| c = cell(2,5);  c ={'Red','Blue','Green','Yellow','White';12345} |

Khi bạn chạy tập tin, nó sẽ hiển thị như sau kết quả:

|  |
| --- |
|  |

#### Truy cập dữ liệu trong mảng ô

Có hai cách để tham chiếu đến các phần tử của một mảng ô:

* Kèm theo các chỉ số trong dấu ngoặc đầu tiên (), để chỉ các bộ tế bào
* Kèm theo các chỉ số trong dấu ngoặc nhọn {}, để chỉ dữ liệu trong từng cá nhân các ô.

Khi bạn đặt các chỉ số trong dấu ngoặc đơn đầu tiên, nó đề cập đến tập hợp các tế bào.

Chỉ số mảng ô trong ngoặc đơn đề cập đến tập hợp các ô.

**Ví dụ:**

|  |
| --- |
| c ={'Red','Blue','Green','Yellow','White';12345};  c(1:2,1:2) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

Bạn cũng có thể truy cập nội dung của các ô bằng cách lập chỉ mục bằng dấu ngoặc nhọn.

**Ví dụ:**

|  |
| --- |
| c ={'Red','Blue','Green','Yellow','White';12345};  c{1,2:4} |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| ans =  Blue  ans =  Green  ans = Yellow |

## KÝ HIỆU DẤU HAI CHẤM

Dấu hai chấm (:) là một trong những toán tử hữu ích nhất trong MATLAB. Nó được sử dụng để tạo các vectơ, mảng chỉ số,

và chỉ định cho các lần lặp.

Nếu bạn muốn tạo một vectơ hàng, chứa các số nguyên từ 1 đến 10, bạn viết:

|  |
| --- |
| 1:10 |

MATLAB thực thi câu lệnh và trả về một vectơ hàng chứa các số nguyên từ 1 đến 10:

|  |
| --- |
|  |

**Ví dụ:** Nếu bạn muốn chỉ định một giá trị gia tăng khác với một giá trị:

|  |
| --- |
| 100:-5:50 |

MATLAB thực hiện câu lệnh và trả về kết quả như sau:

|  |
| --- |
|  |

Hãy để chúng tôi lấy một ví dụ khác:

|  |
| --- |
| 0:pi/8:pi |

MATLAB thực hiện câu lệnh và trả về kết quả như sau:

|  |
| --- |
|  |

Bạn có thể sử dụng toán tử dấu hai chấm để tạo một vectơ chỉ số để chọn hàng, cột hoặc phần tử của mảng. Bảng sau đây mô tả việc sử dụng nó cho mục đích này (hãy để chúng tôi có một ma trận A):

|  |  |
| --- | --- |
| **Định dạng** | **Mục đích** |
| A(:,j) | là cột thứ j của A. |
| A(i, :) | là hàng thứ i của A. |
| A(:,:) | là mảng hai chiều tương đương. Đối với ma trận, điều này giống như A. |
| A(j:k) | là A(j), A(j+1),...,A(k). |
| A(:,j:k) | là A(:,j), A(:,j+1),...,A(:,k). |
| A(:,:,k) | thứ k của mảng ba chiều A. |
| A(i,j,k,:) | là một vectơ trong mảng bốn chiều A. Vectơ bao gồm A(i,j,k,1), A(i,j,k,2), A(i,j,k,3), v.v. |
| A(:) | là tất cả các phần tử của A, được coi là một cột duy nhất. Ở phía bên trái của câu lệnh gán, A(:) điền vào A, giữ nguyên hình dạng của nó từ trước đó. Trong trường hợp này, vế phải phải chứa cùng số phần tử với A. |

**Ví dụ:** Tạo một tệp script và nhập mã sau vào đó:

|  |
| --- |
| A =[1234;4567;78910]  A(:,2)% second column of A  A(:,2:3)% second and third column of A  A(2:3,2:3)% second and third rows and second and third columns |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

## SỐ

MATLAB hỗ trợ các lớp số khác nhau bao gồm các số nguyên có dấu và không dấu và độ chính xác đơn và số dấu phẩy động có độ chính xác kép. Theo mặc định, MATLAB lưu trữ tất cả các giá trị số dưới dạng dấu phẩy động có độ chính xác kép con số.

Bạn có thể chọn lưu trữ bất kỳ số hoặc dãy số nào dưới dạng số nguyên hoặc số có độ chính xác đơn.

Tất cả các kiểu số đều hỗ trợ các phép toán mảng và phép toán cơ bản.

Chuyển đổi sang các loại dữ liệu số khác nhau

MATLAB cung cấp các hàm sau để chuyển đổi sang các kiểu dữ liệu số khác nhau:

|  |  |
| --- | --- |
| **Lệnh** | **Mục đích** |
| int8 | Số nguyên có dấu 8 bit |
| uint8 | Số nguyên không dấu 8 bit |
| int16 | Số nguyên có dấu 16 bit |
| uint16 | Số nguyên không dấu 16 bit |
| int32 | Số nguyên có dấu 32 bit |
| uint32 | Số nguyên không dấu 32 bit |
| int64 | Số nguyên có dấu 64 bit |
| uint64 | Số nguyên không dấu 64-bit |

**Ví dụ:** Tạo một tệp script và nhập mã sau:

|  |
| --- |
| x = single([5.323.476.28]).\*7.5  x =double([5.323.476.28]).\*7.5  x = int8([5.323.476.28]).\*7.5  x = int16([5.323.476.28]).\*7.5  x = int32([5.323.476.28]).\*7.5  x = int64([5.323.476.28]).\*7.5 |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

**Ví dụ:** Tạo một tệp script và nhập mã sau:

|  |
| --- |
| x = int32([5.323.476.28]).\*7.5  x = int64([5.323.476.28]).\*7.5  x = num2cell(x) |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

### Số nguyên nhỏ nhất và lớn nhất

Các hàm intmax() và intmin() trả về các giá trị lớn nhất và nhỏ nhất có thể được biểu diễn với tất cả các loại số nguyên.

Cả hai hàm đều lấy kiểu dữ liệu số nguyên làm đối số, ví dụ: intmax(int8) hoặc intmin(int64) và trả về các giá trị lớn nhất và nhỏ nhất mà bạn có thể biểu thị bằng kiểu dữ liệu số nguyên.

**Ví dụ:** Ví dụ sau minh họa cách lấy giá trị nhỏ nhất và lớn nhất của số nguyên. Tạo một tệp script và viết đoạn mã sau vào đó:

|  |
| --- |
| % displaying the smallest and largest signed integer data str ='The range for int8 is:\n\t%d to %d ';  sprintf(str, intmin('int8'), intmax('int8')) str ='The range for int16 is:\n\t%d to %d ';  sprintf(str, intmin('int16'), intmax('int16')) str ='The range for int32 is:\n\t%d to %d '; sprintf(str, intmin('int32'), intmax('int32')) str ='The range for int64 is:\n\t%d to %d '; sprintf(str, intmin('int64'), intmax('int64'))  % displaying the smallest and largest unsigned integer data str ='The range for uint8 is:\n\t%d to %d ';  sprintf(str, intmin('uint8'), intmax('uint8')) str ='The range for uint16 is:\n\t%d to %d '; sprintf(str, intmin('uint16'), intmax('uint16')) str ='The range for uint32 is:\n\t%d to %d '; sprintf(str, intmin('uint32'), intmax('uint32')) str ='The range for uint64 is:\n\t%d to %d '; sprintf(str, intmin('uint64'), intmax('uint64')) |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

### Số dấu phẩy động nhỏ nhất và lớn nhất

Các hàm realmax() và realmin() trả về các giá trị lớn nhất và nhỏ nhất có thể được biểu diễn bằng các số dấu phẩy động.

Cả hai hàm khi được gọi với đối số 'đơn', trả về các giá trị tối đa và tối thiểu mà bạn có thể biểu thị bằng kiểu dữ liệu có độ chính xác đơn và khi được gọi với đối số 'kép', trả về các giá trị tối đa và tối thiểu mà bạn có thể biểu thị bằng kiểu dữ liệu chính xác kép.

**Ví dụ:** Ví dụ sau minh họa cách lấy số dấu phẩy động nhỏ nhất và lớn nhất. Tạo một tệp script và viết đoạn mã sau vào đó:

|  |
| --- |
| % displaying the smallest and largest single-precision  % floating point number  str ='The range for single is:\n\t%g to %g and\n\t %g to %g'; sprintf(str,-realmax('single'),-realmin('single'),...  realmin('single'), realmax('single'))  % displaying the smallest and largest double-precision  % floating point number  str ='The range for double is:\n\t%g to %g and\n\t %g to %g'; sprintf(str,-realmax('double'),-realmin('double'),...  realmin('double'), realmax('double')) |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

## CHUỖI

Việc tạo một chuỗi ký tự khá đơn giản trong MATLAB. Trong thực tế, chúng tôi đã sử dụng nó nhiều lần.

**Ví dụ:** Bạn gõ như sau trong dấu nhắc lệnh:

|  |
| --- |
| my\_string = 'Tutorial''s Point' |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

MATLAB coi tất cả các biến là mảng, và các chuỗi được coi là mảng ký tự. Hãy để chúng tôi sử dụng lệnh **whos** để kiểm tra biến được tạo ở trên:

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

Điều thú vị là bạn có thể sử dụng các hàm chuyển đổi số như uint8 hoặc uint16 để chuyển đổi các ký tự trong chuỗi thành mã số của chúng. Hàm char chuyển đổi vectơ số nguyên thành ký tự:

**Ví dụ:** Tạo một tệp script và nhập đoạn mã sau vào đó:

|  |
| --- |
| my\_string ='Tutorial''s Point';  str\_ascii = uint8(my\_string)%8-bit ascii values str\_back\_to\_char=char(str\_ascii)  str\_16bit = uint16(my\_string)%16-bit ascii values str\_back\_to\_char =char(str\_16bit) |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

### Mảng ký tự hình chữ nhật

Các chuỗi chúng ta đã thảo luận cho đến nay là các mảng ký tự một chiều; tuy nhiên, chúng ta cần lưu trữ nhiều hơn thế. Chúng tôi cần lưu trữ dữ liệu văn bản nhiều chiều hơn trong chương trình của mình. Điều này đạt được bằng cách tạo các mảng ký tự hình chữ nhật.

Cách đơn giản nhất để tạo một mảng ký tự hình chữ nhật là nối hai hoặc nhiều mảng ký tự một chiều, theo chiều dọc hoặc chiều ngang theo yêu cầu.

Bạn có thể kết hợp các chuỗi theo chiều dọc theo một trong các cách sau:

Sử dụng toán tử nối MATLAB [] và phân tách mỗi hàng bằng dấu chấm phẩy (;). Xin lưu ý rằng trong phương pháp này, mỗi hàng phải chứa cùng một số ký tự. Đối với các chuỗi có độ dài khác nhau, bạn nên thêm các ký tự khoảng trắng nếu cần.

Sử dụng hàm char . Nếu các chuỗi có độ dài khác nhau, ký tự đệm các chuỗi ngắn hơn bằng các khoảng trống ở cuối để mỗi hàng có cùng số lượng nhân vật.

**Ví dụ:** Tạo một tệp script và nhập đoạn mã sau vào nó:

|  |
| --- |
|  |

Khi bạn chạy tập tin, nó sẽ hiển thị như sau kết quả:

|  |
| --- |
|  |

Bạn có thể kết hợp các chuỗi theo chiều ngang theo một trong các cách sau:

|  |
| --- |
|  |

Sử dụng toán tử nối MATLAB, [] và phân tách các chuỗi đầu vào bằng dấu phẩy hoặc dấu cách. Phương pháp này bảo toàn bất kỳ dấu cách nào trong đầu vào mảng.

Sử dụng chức năng nối chuỗi, strcat . Phương pháp này loại bỏ dấu cách trong đầu vào

**Ví dụ:** Tạo một tệp script và nhập đoạn mã sau vào nó:

|  |
| --- |
|  |

Khi bạn chạy tập tin, nó sẽ hiển thị như sau kết quả:

|  |
| --- |
|  |

### Kết hợp các chuỗi thành một mảng ô

Từ cuộc thảo luận trước đây của chúng ta, rõ ràng là việc kết hợp các chuỗi có độ dài khác nhau có thể là một điều khó khăn vì tất cả các chuỗi trong các mảng có đến thì là ở của các như nhau chiều dài. chúng tôi có được sử dụng khoảng trông tại các chấm dứt của dây đến cân bằng của họ chiều dài.

Tuy nhiên, một cách hiệu quả hơn để kết hợp các chuỗi là chuyển mảng kết quả thành một mảng ô.

Mảng ô MATLAB có thể chứa các kích thước và loại dữ liệu khác nhau trong một mảng. Mảng ô cung cấp một cách linh hoạt hơn để lưu trữ các chuỗi có độ dài khác nhau.

Hàm cellstr chuyển đổi một mảng ký tự thành một mảng ô gồm các chuỗi.

**Ví dụ:** Tạo một tệp script và nhập đoạn mã sau vào nó:

|  |
| --- |
|  |

Khi bạn chạy tập tin, nó sẽ hiển thị như sau kết quả:

|  |
| --- |
|  |

### Hàm chuỗi trong MATLAB

MATLAB cung cấp nhiều hàm chuỗi để tạo, kết hợp, phân tích cú pháp, so sánh và thao tác với các chuỗi. Bảng sau đây cung cấp mô tả ngắn gọn về các hàm chuỗi trong MATLAB:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| khoảng trống | Tạo chuỗi ký tự trống |
| di động | Tạo mảng ô của chuỗi từ mảng ký tự |
| Char | Chuyển thành mảng ký tự (chuỗi) |
| iscellstr | Xác định xem đầu vào có phải là mảng ô của chuỗi không |
| ischar | Xác định xem mục có phải là mảng ký tự không |
| tăng tốc | Định dạng dữ liệu thành chuỗi |
| Strcat | Nối chuỗi theo chiều ngang |
| liên kết | Nối các chuỗi trong mảng ô thành một chuỗi |
| Các hàm xác định các phần của chuỗi, tìm và thay thế chuỗi con | |
| ischar | Xác định xem mục có phải là mảng ký tự không |
| thư tín | Các phần tử của mảng là các chữ cái |
| không gian | Các phần tử mảng là các ký tự khoảng trắng |
| isstrprop | Xác định xem chuỗi có thuộc danh mục được chỉ định hay không |
| Sscanf | Đọc dữ liệu được định dạng từ chuỗi |
| Strfind | Tìm một chuỗi trong một chuỗi khác |
| Strep | Tìm và thay thế chuỗi con |
| chia tách | Tách chuỗi tại dấu phân cách đã chỉ định |
| sải bước | Các phần được chọn của chuỗi |
| chuỗi xác thực | Kiểm tra tính hợp lệ của chuỗi văn bản |
| Symvar | Xác định các biến ký hiệu trong biểu thức |
| biểu thức chính quy | Khớp biểu thức chính quy (phân biệt chữ hoa chữ thường) |
| biểu thức chính quy | Khớp biểu thức chính quy (không phân biệt chữ hoa chữ thường) |
| biểu diễn lại | Thay thế chuỗi bằng biểu thức chính quy |
| Regexptranslate | Dịch chuỗi thành biểu thức chính quy |
| Hàm so sánh chuỗi | |
| Strcmp | So sánh các chuỗi (phân biệt chữ hoa chữ thường) |
| Strcmpi | So sánh các chuỗi (không phân biệt chữ hoa chữ thường) |
| strncmp | So sánh n ký tự đầu tiên của chuỗi (phân biệt chữ hoa chữ thường) |
| Strncmpi | So sánh n ký tự đầu tiên của chuỗi (không phân biệt chữ hoa chữ thường) |

|  |  |
| --- | --- |
| Các hàm đổi chuỗi thành chữ hoa hoặc chữ thường, tạo hoặc xóa khoảng trắng | |
| bỏ trống | Tách các khoảng trống ở cuối chuỗi |
| Strtrim | Xóa khoảng trắng ở đầu và cuối chuỗi |
| Thấp hơn | Chuyển đổi chuỗi thành chữ thường |
| Phía trên | Chuyển đổi chuỗi thành chữ hoa |
| chỉ cần | Căn chỉnh mảng ký tự |

### Ví dụ

Các ví dụ sau minh họa một số hàm chuỗi đã đề cập ở trên:

#### Định dạng chuỗi

Tạo một tệp script và nhập đoạn mã sau vào đó:

|  |
| --- |
| A = pi\*1000\*ones(1,5);  sprintf(' %f \n %.2f \n %+.2f \n %12.2f \n %012.2f \n', A) |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

#### tham gia chuỗi

Tạo một tệp script và nhập đoạn mã sau vào nó:

|  |
| --- |
| %cell array of strings  str\_array ={'red','blue','green','yellow','orange'};  %Join strings in cell array into single string str1 = strjoin("-", str\_array)  str2 = strjoin(",", str\_array) |

Khi bạn chạy tập tin, nó sẽ hiển thị như sau kết quả:

|  |
| --- |
|  |

#### Tìm và thay thế chuỗi

Tạo một tệp script và nhập đoạn mã sau vào đó:

|  |
| --- |
| students ={'Zara Ali','Neha Bhatnagar',... 'Monica Malik','Madhu Gautam',...  'Madhu Sharma','Bhawna Sharma',... 'Nuha Ali','Reva Dutta',... 'Sunaina Ali','Sofia Kabir'};  %The strrep function searches and replaces sub-string. new\_student = strrep(students(8),'Reva','Poulomi')  %Display first names first\_names = strtok(students) |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

#### So sánh chuỗi

Tạo một tệp script và nhập đoạn mã sau vào nó:

|  |
| --- |
|  |

Khi bạn chạy tập tin, nó sẽ hiển thị như sau kết quả:

|  |
| --- |
|  |

## HÀM

Hàm là một nhóm các câu lệnh cùng nhau thực hiện một nhiệm vụ. Trong MATLAB, các hàm được định nghĩa trong

các tập tin riêng biệt. Tên của tệp và của hàm phải giống nhau.

Các hàm hoạt động trên các biến trong không gian làm việc riêng của chúng, còn được gọi là không gian làm việc cục bộ, riêng biệt

từ không gian làm việc mà bạn truy cập tại dấu nhắc lệnh MATLAB được gọi là không gian làm việc cơ sở.

Các hàm có thể chấp nhận nhiều hơn một đối số đầu vào và có thể trả về nhiều hơn một đối số đầu ra

Cú pháp của một câu lệnh hàm là:

Các hàm có thể chấp nhận nhiều hơn một đối số đầu vào và có thể trả về nhiều hơn một đối số đầu ra Cú pháp của một câu lệnh hàm là:

|  |
| --- |
| function[out1,out2,..., outN]= myfun(in1,in2,in3,..., inN) |

**Ví dụ:** Hàm sau có tên mymax phải được viết trong tệp có tên mymax.m . Nó nhận năm số làm đối số và trả về giá trị lớn nhất của các số.

Tạo một tệp chức năng, được đặt tên là mymax.m và nhập mã sau vào đó:

|  |
| --- |
| function max = mymax(n1, n2, n3, n4, n5)  %Thisfunction calculates the maximum of the  % five numbers given as input max = n1;  if(n2 > max)  max = n2;  end  if(n3 > max) max = n3;  end  if(n4 > max)  max = n4;  end  if(n5 > max)  max = n5;  end |

Dòng đầu tiên của hàm bắt đầu bằng từ khóa function . Nó đưa ra tên của hàm và thứ tự của các đối số. Trong ví dụ của chúng ta, hàm mymax có năm đối số đầu vào và một đối số đầu ra.

Các dòng nhận xét xuất hiện ngay sau câu lệnh chức năng cung cấp văn bản trợ giúp. Những dòng này được in khi bạn gõ:

|  |
| --- |
| help mymax |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
| This function calculates the maximum of the five  numbers given as input |

Bạn có thể gọi hàm là:

|  |
| --- |
| mymax(34,78,89,23,11) |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

### Hàm ẩn danh

Một hàm ẩn danh giống như một hàm nội tuyến trong các ngôn ngữ lập trình truyền thống, được định nghĩa trong một câu lệnh MATLAB. Nó bao gồm một biểu thức MATLAB duy nhất và bất kỳ số lượng đối số đầu vào và đầu ra nào.

Bạn có thể định nghĩa một hàm ẩn danh ngay tại dòng lệnh MATLAB hoặc trong một hàm hoặc tập lệnh. Bằng cách này, bạn có thể tạo các hàm đơn giản mà không phải tạo tệp cho chúng.

Cú pháp để tạo một hàm ẩn danh từ một biểu thức là

|  |
| --- |
| f =@(arglist)expression |

**Ví dụ**

Trong ví dụ này, chúng ta sẽ viết một hàm ẩn danh có tên là power, hàm này sẽ lấy hai số làm đầu vào và trả về số đầu tiên được nâng lên lũy thừa của số thứ hai.

power =@(x, n) x.^n; result1 = power(7,3) result2 = power(49,0.5) result3 = power(10,-10) result4 = power (4.5,1.5)

Tạo một tệp script và nhập mã sau vào đó:

Khi bạn chạy tệp, nó sẽ hiển thị:

|  |
| --- |
|  |

result1 = 343

result2 =

7

result3 =

1.0000e-10

result4 =

9.5459

Chức năng chính và phụ

Bất kỳ chức năng nào không phải là chức năng ẩn danh phải được xác định trong một tệp. Mỗi tệp chức năng chứa một chức năng chính bắt buộc xuất hiện đầu tiên và bất kỳ số lượng chức năng phụ tùy chọn nào xuất hiện sau chức năng chính và được nó sử dụng.

Các chức năng chính có thể được gọi từ bên ngoài tệp xác định chúng, từ dòng lệnh hoặc từ các chức năng khác, nhưng các chức năng phụ không thể được gọi từ dòng lệnh hoặc các chức năng khác, bên ngoài tệp chức năng.

Các chức năng phụ chỉ hiển thị với chức năng chính và các chức năng phụ khác trong tệp chức năng xác định chúng.

**Ví dụ**

Chúng ta hãy viết một hàm có tên là bậc hai để tính các nghiệm của một phương trình bậc hai. Hàm sẽ nhận ba đầu vào, hệ số bậc hai, hệ số tuyến tính và số hạng không đổi. Nó sẽ trả lại gốc rễ.

Tệp hàm quadratic.m sẽ chứa hàm chính bậc hai và hàm phụ disc , hàm này tính toán phân biệt.

Tạo một tệp hàm quadratic.m và nhập mã sau vào đó:

function[x1,x2]= quadratic(a,b,c)

%thisfunction returns the roots of

% a quadratic equation.

%It takes 3 input arguments

% which are the co-efficients of x2, x and the

%constant term

%It returns the roots d = disc(a,b,c);

x1 =(-b + d)/(2\*a);

x2 =(-b - d)/(2\*a); end%end of quadratic

function dis = disc(a,b,c)

%function calculates the discriminant dis = sqrt(b^2-4\*a\*c);

end%end of sub-function

|  |
| --- |
|  |

Bạn có thể gọi hàm trên từ dấu nhắc lệnh là:

|  |
| --- |
|  |

quadratic(2,4,-4)

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

ans = 0.7321

### Hàm lồng nhau

Bạn có thể định nghĩa các hàm bên trong phần thân của một hàm khác. Chúng được gọi là các hàm lồng nhau. Hàm lồng nhau chứa bất kỳ hoặc tất cả các thành phần của bất kỳ hàm nào khác.

Các hàm lồng nhau được xác định trong phạm vicủa một hàm khác và chúng chia sẻ quyền truy cập vào các hàm chứa không gian làm việc.

Hàm lồng nhau tuân theo cú pháp sau:

function x = A(p1, p2)

... B(p2)

function y = B(p3)

...

end

...

end

|  |
| --- |
|  |

**Ví dụ:**

Chúng ta hãy viết lại hàm bậc hai , từ ví dụ trước, tuy nhiên, lần này hàm đĩa sẽ là một hàm lồng nhau.

Tạo một tệp hàm quadratic2.m và nhập mã sau vào đó:

function[x1,x2]= quadratic2(a,b,c) function disc % nested function

d = sqrt(b^2-4\*a\*c); end%end of function disc disc;

x1 =(-b + d)/(2\*a);

x2 =(-b - d)/(2\*a);

end%end of function quadratic2

|  |
| --- |
|  |

Bạn có thể gọi hàm trên từ dấu nhắc lệnh là:

|  |
| --- |
|  |

quadratic2(2,4,-4)

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

ans = 0.7321

|  |
| --- |
|  |

Chức năng riêng tư

Chức năng riêng tư là chức năng chính chỉ hiển thị đối với một nhóm giới hạn các chức năng khác. Nếu bạn không muốn tiết lộ việc triển khai (các) chức năng, bạn có thể tạo chúng dưới dạng các chức năng riêng tư.

Các chức năng riêng tư nằm trong các thư mục con có tên đặc biệt private . Chúng chỉ hiển thị với các chức năng trong thư mục mẹ.

**Ví dụ:**

Hãy viết lại hàm bậc hai . Tuy nhiên, lần này, hàm đĩa tính toán phân biệt đối xử, sẽ là một hàm riêng tư.

Tạo một thư mục con có tên private trong working directory. Lưu trữ tệp chức năng sau disc.m trong đó:

function dis = disc(a,b,c)

%function calculates the discriminant dis = sqrt(b^2-4\*a\*c);

end%end of sub-function

|  |
| --- |
|  |

Tạo một hàm quadratic3.m trong thư mục làm việc của bạn và nhập đoạn mã sau vào đó:

|  |
| --- |
|  |

function[x1,x2]= quadratic3(a,b,c)

%thisfunction returns the roots of

% a quadratic equation.

%It takes 3 input arguments

% which are the co-efficients of x2, x and the

%constant term

%It returns the roots d = disc(a,b,c);

x1 =(-b + d)/(2\*a);

x2 =(-b - d)/(2\*a); end%end of quadratic3

Bạn có thể gọi hàm trên từ dấu nhắc lệnh là:

quadratic3(2,4,-4)

|  |
| --- |
|  |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

ans = 0.7321

Biến toàn cục

Các biến toàn cục có thể được chia sẻ bởi nhiều hàm. Đối với điều này, bạn cần khai báo biến là toàn cầu trong tất cả các chức năng.

Nếu bạn muốn truy cập biến đó từ vùng làm việc cơ sở thì hãy khai báo biến đó tại dòng lệnh.

Khai báo toàn cục phải xảy ra trước khi biến thực sự được sử dụng trong một hàm. Nên sử dụng chữ in hoa cho tên của các biến toàn cục để phân biệt chúng với các biến khác.

**Ví dụ:** Hãy để chúng tôi tạo một tệp chức năng có tên là trung bình.m và nhập mã sau vào đó:

function avg = average(nums) global TOTAL

avg = sum(nums)/TOTAL; end

|  |
| --- |
|  |

Tạo một tệp script và nhập mã sau vào đó:

|  |
| --- |
|  |

global TOTAL; TOTAL =10;

n =[34,45,25,45,33,19,40,34,38,42];

av = average(n)

Khi bạn chạy file sẽ hiển thị kết quả như sau:

av = 35.5000

|  |
| --- |
|  |

## NHẬP DỮ LIỆU

tôi đang nhập da ta trong MA T L AB m ean s loading da ta từ một e xte rnal tập tin . anh ấy tôi p ortdata fu n cti on cho phép tải nhiều tệp dữ liệu có định dạng khác nhau. Nó có năm hình thức sau:

|  |  |
| --- | --- |
| **SN** | **Chức năng và Mô tả** |
| 1 | A = nhập dữ liệu (tên tệp)  Tải dữ liệu vào mảng A từ tệp được biểu thị bằng tên tệp . |
| 2 | A = nhập dữ liệu ('-pastespecial')  Tải dữ liệu từ khay nhớ tạm của hệ thống thay vì từ tệp. |
| 3 | Một = nhập dữ liệu( , dấu phân cáchIn)  Diễn giải delimiterIn là dấu tách cột trong tệp ASCII, tên tệp hoặc dữ liệu khay nhớ tạm. Bạn có thể sử dụng delimiterIn với bất kỳ đối số đầu vào nào trong các cú pháp trên. |
| 4 | Một = nhập dữ liệu( , delimiterIn, headerlinesIn)  Tải dữ liệu từ tệp ASCII, tên tệp hoặc khay nhớ tạm, đọc dữ liệu số bắt đầu từ dòng headerlinesIn+1 . |
| 5 | [A, delimiterOut, headerlinesOut] = nhập dữ liệu ( )  Ngoài ra, trả về ký tự dấu phân cách được phát hiện cho tệp ASCII đầu vào trong delimiterOut và số dòng tiêu đề được phát hiện trong headerlinesOut , sử dụng bất kỳ đối số đầu vào nào trong các cú pháp trước đó. |

Theo mặc định, Octave không hỗ trợ hàm importdata() , vì vậy bạn sẽ phải tìm kiếm và cài đặt gói này để các ví dụ sau hoạt động với cài đặt Octave của bạn.

ví dụ 1

Hãy để chúng tôi tải và hiển thị một tệp hình ảnh. Tạo một tệp script và nhập mã sau vào đó:

filename ='smile.jpg';

A = importdata(filename); image(A);

|  |
| --- |
|  |

Khi bạn chạy tệp, MATLAB sẽ hiển thị tệp hình ảnh. Tuy nhiên, bạn phải lưu trữ nó trong thư mục hiện hành.



**ví dụ 2**

Trong ví dụ này, chúng tôi nhập tệp văn bản và chỉ định Dấu phân cách và Tiêu đề cột. Hãy để chúng tôi tạo một tệp ASCII được phân tách bằng dấu cách với các tiêu đề cột, được đặt tên là Weeklydata.txt .

Tệp văn bản của chúng tôi Weeklydata.txt trông như thế này:

|  |
| --- |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 95,01 76,21 61,54 | 40,57 | 55,79 | 70.28 | 81,53 |
| 73,11 45,65 79,19 | 93,55 | 75.29 | 69,87 | 74,68 |
| 60,68 41,85 92,18 | 91,69 | 81,32 | 90,38 | 74,51 |
| 48,60 82,14 73,82 | 41.03 | 0,99 | 67.22 | 93.18 |
| 89,13 44,47 57,63 | 89,36 | 13,89 | 19,88 | 46,60 |

Tạo một tệp script và nhập mã sau vào đó:

filename ='weeklydata.txt'; delimiterIn =' '; headerlinesIn =1;

A = importdata(filename,delimiterIn,headerlinesIn);

%View data for k =[1:7]

disp(A.colheaders{1, k}) disp(A.data(:, k)) disp(' ')

end

|  |
| --- |
|  |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

SunDay

95.0100

73.1100

60.6800

48.6000

89.1300

MonDay

76.2100

45.6500

41.8500

82.1400

44.4700

TuesDay

61.5400

79.1900

92.1800

73.8200

57.6300

WednesDay

40.5700

93.5500

91.6900

41.0300

89.3600

ThursDay

55.7900

75.2900

81.3200

0.9900

13.8900

FriDay

70.2800

69.8700

90.3800

67.2200

19.8800

SatureDay

81.5300

74.6800

74.5100

93.1800

46.6000

Ví dụ 3

Trong ví dụ này, chúng ta hãy nhập dữ liệu từ clipboard. Sao chép các dòng sau vào khay nhớ tạm:

Toán học thật đơn giản

Tạo một tệp script và nhập mã sau:

A = importdata('-pastespecial')

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

A =

'Mathematics is simple'

### I/O tệp cấp thấp

Hàm importdata là một hàm cấp cao. Các chức năng I/O tệp cấp thấp trong MATLAB cho phép kiểm soát nhiều nhất đối với việc đọc hoặc ghi dữ liệu vào một tệp. Tuy nhiên, các chức năng này cần thông tin chi tiết hơn về tệp của bạn để hoạt động hiệu quả.

MATLAB cung cấp các chức năng sau cho thao tác đọc và ghi ở mức byte hoặc ký tự:

|  |  |
| --- | --- |
| Chức năng | Sự miêu tả |
| Fđóng | Đóng một hoặc tất cả các tệp đang mở |
| tội lỗi | Kiểm tra cuối tập tin |
| sợ hãi | Thông tin về lỗi tệp I/O |
| Fgetl | Đọc dòng từ tệp, xóa ký tự dòng mới |
| Fgets | Đọc dòng từ tệp, giữ ký tự dòng mới |
| mở | Mở tệp hoặc lấy thông tin về các tệp đang mở |
| fprintf | Ghi dữ liệu vào tệp văn bản |
| sợ hãi | Đọc dữ liệu từ tệp nhị phân |
| giải trí | Di chuyển chỉ báo vị trí tệp đến đầu tệp đang mở |
| fscanf | Đọc dữ liệu từ tệp văn bản |
| Fseek | Di chuyển đến vị trí được chỉ định trong tệp |
| kể chuyện | Vị trí trong tệp đang mở |
| Fwrite | Ghi dữ liệu vào tệp nhị phân |

Nhập tệp dữ liệu văn bản với I/O cấp thấp

MATLAB cung cấp các chức năng sau để nhập các tệp dữ liệu văn bản ở mức độ thấp:

Hàm fscanf đọc dữ liệu được định dạng trong văn bản hoặc ASCII tập tin.

Các hàm fgetl và fgets đọc một dòng của tệp tại một thời điểm, trong đó một ký tự dòng mới phân tách từng dòng dòng.

Hàm fread đọc một luồng dữ liệu ở byte hoặc bit cấp độ.

Ví dụ

Chúng tôi có một tệp dữ liệu văn bản 'myfile.txt' được lưu trong thư mục làm việc của chúng tôi. Tệp lưu trữ dữ liệu lượng mưa trong ba tháng; tháng 6, tháng 7 và tháng 8 năm 2012.

Dữ liệu trong myfile.txt chứa các tập hợp lặp lại các phép đo thời gian, tháng và lượng mưa tại năm địa điểm. Dữ liệu header lưu số tháng M; vì vậy chúng ta có M bộ phép đo.

Các tập tin trông như thế này:

|  |  |  |  |
| --- | --- | --- | --- |
| 9,59 9,33 | NaN | 0,31 | 0,23 |
| 10,46 13,17 | NaN | 14,89 | 19.33 |
| 20,97 19,50 | 17,65 | 14,45 | 14.00 |
| 18,23 10,34 | 17,95 | 16,46 | 19.34 |
| 09:10:02 |  |  |  |
| Tháng 7-2012 |  |  |  |
| 12,76 16,94 | 14,38 | 11,86 | 16,89 |
| 20,46 23,17 | NaN | 24,89 | 19.33 |
| 30,97 49,50 | 47,65 | 24,45 | 34.00 |
| 18,23 30,34 | 27,95 | 16,46 | 19.34 |
| 30,46 33,17 | NaN | 34,89 | 29.33 |
| 30,97 49,50 | 47,65 | 24,45 | 34.00 |
| 28,67 30,34 | 27,95 | 36,46 | 29,34 |
| 15:03:40 |  |  |  |
| Tháng 8-2012 |  |  |  |
| 17,09 16,55 | 19,59 | 17,25 | 19.22 |
| 17,54 11,45 | 13,48 | 22,55 | 24.01 |
| NaN 21,19 | 25,85 | 25.05 | 27.21 |
| 26,79 24,98 | 12.23 | 16,99 | 18,67 |
| 17,54 11,45 | 13,48 | 22,55 | 24.01 |
| NaN 21,19 | 25,85 | 25.05 | 27.21 |
| 26,79 24,98 | 12.23 | 16,99 | 18,67 |

Chúng tôi sẽ nhập dữ liệu từ tệp này và hiển thị dữ liệu này. Thực hiện các bước sau:

Mở tệp bằng chức năng fopen và lấy tệp định danh.

Mô tả dữ liệu trong tệp bằng các chỉ định định dạng , chẳng hạn như ' %s ' cho chuỗi, ' %d ' cho số nguyên hoặc ' %f ' cho dấu phẩy động con số.

Để bỏ qua các ký tự chữ trong tệp, hãy đưa chúng vào phần mô tả định dạng. Để bỏ qua trường dữ liệu, hãy sử dụng dấu hoa thị ('\*') trong người chỉ định.

|  |
| --- |
|  |

Ví dụ: để đọc các tiêu đề và trả về một giá trị duy nhất cho M, chúng tôi viết:

M = fscanf(fid,'%\*s %\*s\n%\*s %\*s %\*s %\*s\nM=%d\n\n',1);

Theo mặc định, fscanf đọc dữ liệu theo mô tả định dạng của chúng tôi cho đến khi nó không thể khớp mô tả với dữ liệu hoặc đọc đến cuối tệp. Ở đây chúng ta sẽ sử dụng vòng lặp for để đọc 3 bộ dữ liệu và mỗi lần nó sẽ đọc 7 hàng và 5 cột.

Chúng tôi sẽ tạo một cấu trúc có tên mydata trong không gian làm việc để lưu trữ dữ liệu được đọc từ tệp. Cấu trúc này có ba trường - thời gian , tháng và dữ liệu mưa mảng.

Tạo một tệp script và nhập mã sau vào đó:

filename ='/data/myfile.txt'; rows =7;

cols =5;

% open the file

fid = fopen(filename);

% read the file headers, find M (number of months)

M = fscanf(fid,'%\*s %\*s\n%\*s %\*s %\*s %\*s\nM=%d\n\n',1);

% read each set of measurements for n =1:M

mydata(n).time = fscanf(fid,'%s',1); mydata(n).month = fscanf(fid,'%s',1);

|  |
| --- |
|  |

% fscanf fills the array in column order,

% so transpose the results mydata(n).raindata =...

fscanf(fid,'%f',[rows, cols]);

end

for n =1:M

disp(mydata(n).time), disp(mydata(n).month) disp(mydata(n).raindata)

end

% close the file fclose(fid);

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 17.2100 | 17.5700 | 11.0900 | 13.1700 | 14.4500 |
| 28.5200 | NaN | 9.5900 | NaN | 14.0000 |
| 39.7800 | 12.0100 | 9.3300 | 14.8900 | 18.2300 |
| 16.5500 | 17.9200 | NaN | 19.3300 | 10.3400 |
| 23.6700 | 28.4900 | 0,3100 | 20.9700 | 17.9500 |
| 19.1500 | 17.4000 | 0,2300 | 19.5000 | 16.4600 |
| 0,3500 | 17.0600 | 10.4600 | 17.6500 | 19.3400 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 12.7600 | NaN | 34.0000 | 33.1700 | 24.4500 |
| 16.9400 | 24.8900 | 18.2300 | NaN | 34.0000 |
| 14.3800 | 19.3300 | 30.3400 | 34.8900 | 28.6700 |
| 11.8600 | 30.9700 | 27.9500 | 29.3300 | 30.3400 |
| 16.8900 | 49.5000 | 16.4600 | 30.9700 | 27.9500 |
| 20.4600 | 47.6500 | 19.3400 | 49.5000 | 36.4600 |
| 23.1700 | 24.4500 | 30.4600 | 47.6500 | 29.3400 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 17.0900 | 13.4800 | 27.2100 | 11.4500 | 25.0500 |
| 16.5500 | 22.5500 | 26.7900 | 13.4800 | 27.2100 |
| 19.5900 | 24.0100 | 24.9800 | 22.5500 | 26.7900 |
| 17.2500 | NaN | 12.2300 | 24.0100 | 24.9800 |
| 19.2200 | 21.1900 | 16.9900 | NaN | 12.2300 |
| 17.5400 | 25.8500 | 18.6700 | 21.1900 | 16.9900 |
| 11.4500 | 25.0500 | 17.5400 | 25.8500 | 18.6700 |

## XUẤT DỮ LIỆU

Xuất dữ liệu trong MATLAB có nghĩa là ghi vào tệp. MATLAB cho phép bạn sử dụng dữ liệu của mình trong một ứng dụng đọc các tập tin ASCII. Đối với điều này, MATLAB cung cấp một số tùy chọn xuất dữ liệu.

Bạn có thể tạo các loại tệp sau:

* Tệp dữ liệu ASCII hình chữ nhật, được phân tách từ một mảng.
* Tệp nhật ký (hoặc nhật ký) của các lần nhấn phím và văn bản kết quả đầu ra.
* Tệp ASCII chuyên dụng sử dụng các chức năng cấp thấp như fprintf.
* MEX-file để truy cập quy trình C/C++ hoặc Fortran ghi vào một tệp văn bản cụ thể định dạng.

Ngoài ra, bạn cũng có thể xuất dữ liệu sang bảng tính.

Có hai cách để xuất một mảng số dưới dạng tệp dữ liệu ASCII được phân tách:

* Sử dụng chức năng lưu và chỉ định -ASCII vòng loại
* Sử dụng hàm dlmwrite

Cú pháp sử dụng hàm lưu Là:

|  |
| --- |
| save my\_data.out num\_array -ASCII |

Trong đó, my\_data.out là tệp dữ liệu ASCII được phân tách được tạo, num\_array là một mảng số và � ASCII là mã xác định.

Cú pháp sử dụng hàm dlmwrite là:

|  |
| --- |
| dlmwrite('my\_data.out', num\_array, 'dlm\_char') |

Trong đó, my\_data.out là tệp dữ liệu ASCII được phân tách được tạo, num\_array là một mảng số và dlm\_char là ký tự phân cách.

### Ví dụ

Ví dụ sau minh họa khái niệm này. Tạo một tệp script và nhập mã sau:

|  |
| --- |
| num\_array = [ 1 2 3 4 ; 4 5 6 7; 7 8 9 0];  save array\_data1.out num\_array -ASCII;  type array\_data1.out  dlmwrite('array\_data2.out', num\_array, ' ');  type array\_data2.out |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| 1.0000000e+00 2.0000000e+00 3.0000000e+00 4.0000000e+00  4.0000000e+00 5.0000000e+00 6.0000000e+00 7.0000000e+00  7.0000000e+00 8.0000000e+00 9.0000000e+00 0.0000000e+00  1 2 3 4  4 5 6 7  7 8 9 0 |

Xin lưu ý rằng lệnh save -ascii và lệnh dlmwrite không hoạt động với các mảng ô làm đầu vào. Để tạo một tệp ASCII được phân tách từ nội dung của một mảng ô, bạn có thể:

* Hoặc, chuyển đổi mảng ô thành ma trận bằng cách sử dụng cell2mat chức năng
* Hoặc xuất mảng ô bằng I/O tệp cấp thấp chức năng.

Nếu bạn sử dụng chức năng lưu để ghi một mảng ký tự vào tệp ASCII, nó sẽ ghi mã ASCII tương đương của các ký tự vào tệp.

**Ví dụ:** Chúng ta hãy viết từ 'xin chào' vào một tệp:

|  |
| --- |
| h = 'hello';  save textdata.out h -ascii  type textdata.out |

MATLAB thực hiện các câu lệnh trên và hiển thị kết quả sau:

|  |
| --- |
| 1.0400000e+02 1.0100000e+02 1.0800000e+02 1.0800000e+02 1.1100000e+02 |

Đó là các ký tự của chuỗi 'xin chào' ở định dạng ASCII 8 chữ số.

### Viết vào tập tin nhật ký

Các tệp nhật ký là nhật ký hoạt động của phiên MATLAB của bạn. Chức năng nhật ký tạo một bản sao chính xác phiên của bạn trong một tệp đĩa, không bao gồm đồ họa.

Để bật chức năng nhật ký, gõ:

|  |
| --- |
| diary |

Theo tùy chọn, bạn có thể cung cấp tên của tệp nhật ký, chẳng hạn như:

|  |
| --- |
| diary logdata.out |

Để tắt chức năng nhật ký:

|  |
| --- |
| diary off |

Bạn có thể mở tệp nhật ký trong trình soạn thảo văn bản.

### Xuất dữ liệu sang tệp dữ liệu văn bản với I/O cấp thấp

Cho đến nay, chúng tôi đã xuất các mảng số. Tuy nhiên, bạn có thể cần tạo các tệp văn bản khác, bao gồm tổ hợp dữ liệu số và ký tự, tệp đầu ra không phải hình chữ nhật hoặc tệp có sơ đồ mã hóa không phải ASCII. Với những mục đích này, MATLAB cung cấp hàm fprintf cấp thấp .

Như trong các hoạt động tệp I/O cấp thấp, trước khi xuất, bạn cần mở hoặc tạo tệp bằng hàm fopen và lấy mã định danh tệp. Theo mặc định, fopen mở tệp để truy cập chỉ đọc. Bạn nên chỉ định quyền ghi hoặc thêm, chẳng hạn như 'w' hoặc 'a'.

Sau khi xử lý tệp, bạn cần đóng tệp bằng hàm fclose(fid) . Ví dụ sau minh họa khái niệm:

**Ví dụ:** Tạo một tệp script và nhập mã sau vào đó:

|  |
| --- |
| % create a matrix y, with two rows x = 0:10:100; y = [x; log(x)]; % open a file for writing fid = fopen('logtable.txt', 'w'); % Table Header fprintf(fid, 'Log Function\n\n'); % print values in column order % two values appear on each row of the file fprintf(fid, '%f %f\n', y); fclose(fid); % display the file created type logtable.txt |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
| Log Function  0.000000 -Inf  10.000000 2.302585  20.000000 2.995732  30.000000 3.401197  40.000000 3.688879  50.000000 3.912023  60.000000 4.094345  70.000000 4.248495  80.000000 4.382027  90.000000 4.499810  100.000000 4.605170 |

## ÂM MƯU

Xác định x , bằng cách chỉ định phạm vi giá trị cho biến x , mà hàm sẽ là âm mưu

Xác định chức năng, y = f(x)

Gọi lệnh cốt truyện , như cốt truyện (x, y)

Ví dụ sau đây sẽ chứng minh khái niệm này. Chúng ta hãy vẽ đồ thị hàm đơn giản y = x cho phạm vigiá trị của x từ 0 đến 100, với số gia là 5.

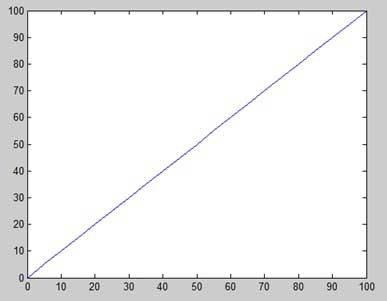
Tạo một tệp script và nhập mã sau:

x =[0:5:100];

y = x; plot(x, y)

|  |
| --- |
|  |

Khi bạn chạy tệp, MATLAB sẽ hiển thị biểu đồ sau:



Chúng ta hãy lấy thêm một ví dụ để vẽ đồ thị của hàm y = x 2 . Trong ví dụ này, chúng ta sẽ vẽ hai đồ thị có cùng hàm số, nhưng ở lần thứ hai, chúng ta sẽ giảm giá trị của số gia. Xin lưu ý rằng khi chúng tôi giảm gia số, biểu đồ sẽ trở nên mượt mà hơn.

Tạo một tệp script và nhập mã sau:

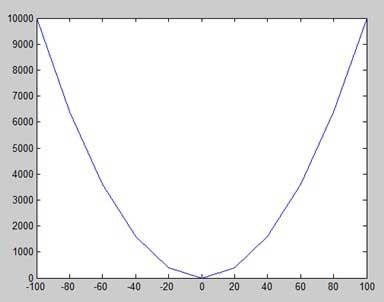
x =[12345678910];

x =[-100:20:100];

y = x.^2; plot(x, y)

|  |
| --- |
|  |

Khi bạn chạy tệp, MATLAB sẽ hiển thị biểu đồ sau:



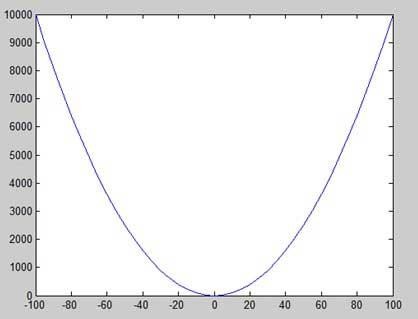
Thay đổi tệp mã một chút, giảm số gia xuống 5:

|  |
| --- |
|  |

x =[-100:5:100];

y = x.^2; plot(x, y)

MATLAB vẽ một biểu đồ mượt mà hơn:



Thêm tiêu đề, nhãn, đường lưới và chia tỷ lệ trên biểu đồ

MATLAB cho phép bạn thêm tiêu đề, nhãn dọc theo trục x và trục y, đường lưới và cũng như điều chỉnh các trục để làm nổi bật biểu đồ.

Các lệnh xlabel và ylabel tạo nhãn dọc theo trục x và trục y.

Lệnh title cho phép bạn đặt tiêu đề trên đồ thị.

Lệnh grid on cho phép bạn đặt các đường lưới trên đồ thị.

Lệnh cân bằng trục cho phép tạo biểu đồ có cùng hệ số tỷ lệ và khoảng cách trên cả hai trục.

Lệnh axis square tạo ra một hình vuông mảnh đất.

**Ví dụ:**

Tạo một tệp script và nhập mã sau:

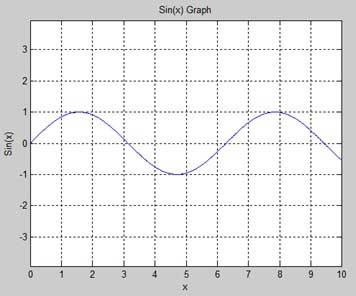
x =[0:0.01:10];

y = sin(x);

plot(x, y), xlabel('x'), ylabel('Sin(x)'), title('Sin(x) Graph'), grid on, axis equal

|  |
| --- |
|  |

MATLAB tạo biểu đồ sau:



### Vẽ nhiều hàm trên cùng một đồ thị

Bạn có thể vẽ nhiều biểu đồ trên cùng một ô. Ví dụ sau minh họa khái niệm:

**Ví dụ:**

Tạo một tệp script và nhập mã sau:

x =[0:0.01:10];

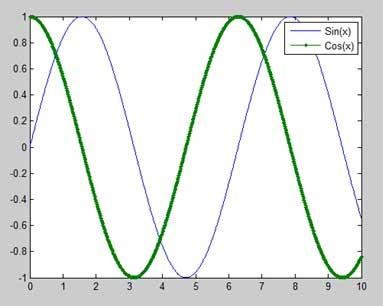
y = sin(x);

g = cos(x);

plot(x, y, x, g,'.-'), legend('Sin(x)','Cos(x)')

|  |
| --- |
|  |

MATLAB tạo biểu đồ sau:



### Đặt màu trên đồ thị

MATLAB cung cấp tám tùy chọn màu cơ bản để vẽ đồ thị. Bảng sau đây cho biết màu sắc và mã của chúng:

|  |  |
| --- | --- |
| Màu | Mã số |
| Trắng | w |
| Đen | k |
| Màu xanh da trời | b |
| Màu đỏ | r |
| lục lam | c |
| Xanh | g |
| đỏ tươi | tôi |
| Màu vàng | y |

**Ví dụ:**

Hãy vẽ đồ thị của hai đa thức 1. f(x) = 3x 4 + 2x 3 + 7x 2 + 2x + 9 và

2. g(x) = 5x 3 + 9x + 2

Tạo một tệp script và nhập mã sau:

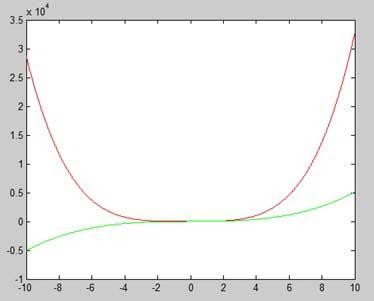
x =[-10:0.01:10];

y =3\*x.^4+2\* x.^3+7\* x.^2+2\* x +9; g =5\* x.^3+9\* x +2;

|  |
| --- |
|  |

plot(x, y,'r', x, g,'g')

Khi bạn chạy tệp, MATLAB tạo biểu đồ sau:



### Đặt tỷ lệ trục

Lệnh axis cho phép bạn đặt tỷ lệ trục. Bạn có thể cung cấp các giá trị tối thiểu và tối đa cho các trục x và y bằng cách sử dụng lệnh axis theo cách sau:

|  |
| --- |
| axis ([xmin xmax ymin ymax]) |

Ví dụ sau đây cho thấy điều này:

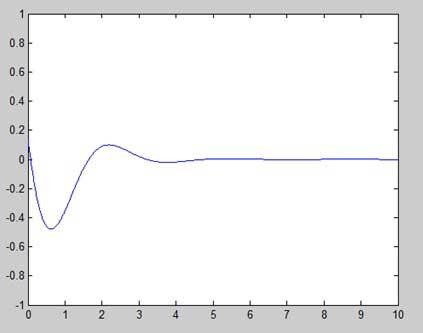
Ví dụ:

Tạo một tệp script và nhập mã sau:

|  |
| --- |
| x =[0:0.01:10];  y = exp(-x).\* sin(2\*x +3);  plot(x, y), axis([010-11]) |

Khi bạn chạy tệp, MATLAB tạo biểu đồ sau:

|  |
| --- |
|  |



### Tạo các ô phụ

Khi bạn tạo một mảng các ô trong cùng một hình, mỗi ô này được gọi là một ô con. Lệnh subplot dùng để tạo ô phụ.

Cú pháp của lệnh là:

|  |
| --- |
| subplot(m, n, p) |

Trong đó, m và n là số hàng và cột của mảng ô và p chỉ định vị trí đặt một ô cụ thể.

Mỗi ô được tạo bằng lệnh subplot có thể có các đặc điểm riêng. Ví dụ sau minh họa các Ý tưởng:

Ví dụ:

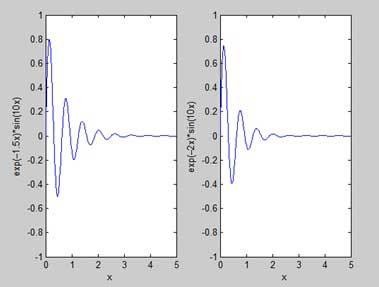
Hãy để chúng tôi tạo ra hai lô:

y = e −1,5x sin(10x) y = e −2x sin(10x)

Tạo một tệp script và nhập mã sau:

|  |
| --- |
| x =[0:0.01:5];  y = exp(-1.5\*x).\*sin(10\*x); subplot(1,2,1)  plot(x,y), xlabel('x'),ylabel('exp(–1.5x)\*sin(10x)'),axis([05-11]) y = exp(-2\*x).\*sin(10\*x);  subplot(1,2,2) plot(x,y),xlabel('x'),ylabel('exp(–2x)\*sin(10x)'),axis([05-11]) |

Khi bạn chạy tệp, MATLAB tạo biểu đồ sau:



## ĐỒ HỌA

Xin chào \_ c ha p ter bị bệnh c on t i n ue e x cày cuốc anh ấy không plo t t i n g một n d g raph i c s c a p ab i li t ie s của \_ M A T L A B . chúng tôi bị bệnh d i s c u s s :

thanh vẽ biểu đồ

Vẽ đường viền

ba chiều mảnh đất

Vẽ biểu đồ thanh

Lệnh bar vẽ biểu đồ thanh hai chiều. Hãy để chúng tôi lấy một ví dụ để chứng minh ý tưởng.

Ví dụ

Hãy để chúng tôi có một lớp học tưởng tượng với 10 sinh víên. Chúng ta biết phần trăm số điểm mà những sinh víên này đạt được là 75, 58, 90, 87, 50, 85, 92, 75, 60 và 95. Chúng ta sẽ vẽ biểu đồ cột cho dữ liệu này.

|  |
| --- |
|  |

Tạo một tệp script và nhập mã sau:

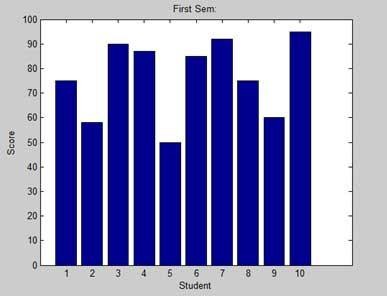
x =[1:10];

y =[75,58,90,87,50,85,92,75,60,95];

bar(x,y), xlabel('Student'),ylabel('Score'), title('First Sem:')

print-deps graph.eps

Khi bạn chạy tệp, MATLAB hiển thị biểu đồ thanh sau:



### Vẽ đường viền

Đường đồng mức của hàm hai biến là một đường cong dọc theo đó hàm có giá trị không đổi. Đường đồng mức được sử dụng để tạo bản đồ đường đồng mức bằng cách nối các điểm có độ cao bằng nhau trên một mức nhất định, chẳng hạn như mực nước biển trung bình.

MATLAB cung cấp chức năng đường đồng mức để vẽ bản đồ đường đồng mức.

**Ví dụ:**

Chúng ta hãy tạo một bản đồ đường đồng mức hiển thị các đường đồng mức cho một hàm đã cho g = f(x, y). Hàm này có hai biến. Vì vậy, chúng ta sẽ phải tạo hai biến độc lập, tức là hai bộ dữ liệu x và y. Điều này được thực hiện bằng cách gọi lệnh meshgrid .

Lệnh meshgrid được sử dụng để tạo ma trận các phần tử cung cấp phạm vitrên x và y cùng với thông số kỹ thuật về số gia trong mỗi trường hợp.

Chúng ta hãy vẽ đồ thị hàm g = f(x, y), trong đó −5 ≤ x ≤ 5, −3 ≤ y ≤ 3. Chúng ta hãy tăng 0,1 cho cả hai giá trị. Các biến được đặt là:

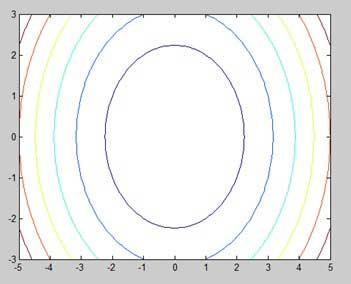
[x,y]= meshgrid(–5:0.1:5,–3:0.1:3);

Cuối cùng, chúng ta cần gán chức năng. Giả sử hàm của chúng ta là: x 2 + y 2 Tạo một tệp script và nhập đoạn mã sau:

[x,y]= meshgrid(-5:0.1:5,-3:0.1:3);%independent variables g = x.^2+ y.^2;%ourfunction

contour(x,y,g)% call the contour function print-deps graph.eps

Khi bạn chạy tệp, MATLAB hiển thị bản đồ đường viền sau:



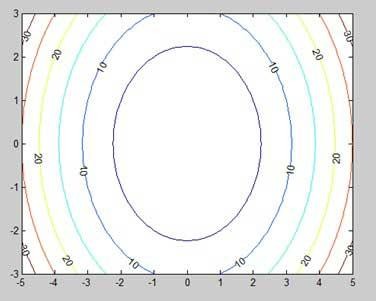
Hãy để chúng tôi sửa đổi mã một chút để tạo bản đồ:

[x,y]= meshgrid(-5:0.1:5,-3:0.1:3);%independent variables g = x.^2+ y.^2;%ourfunction

[C, h]= contour(x,y,g);% call the contour function set(h,'ShowText','on','TextStep',get(h,'LevelStep')\*2) print-deps graph.eps

|  |
| --- |
|  |

Khi bạn chạy tệp, MATLAB hiển thị bản đồ đường viền sau:



#### Ô ba chiều

Các đồ thị ba chiều về cơ bản hiển thị một bề mặt được xác định bởi một hàm hai biến, g = f (x, y).

Như trước đây, để xác định g, trước tiên chúng ta tạo một tập hợp các điểm (x,y) trên miền của hàm bằng cách sử dụng lệnh meshgrid . Tiếp theo, chúng tôi chỉ định chính chức năng đó. Cuối cùng, chúng tôi sử dụng lệnh lướt để tạo một biểu đồ bề mặt.

Ví dụ sau minh họa khái niệm:

Ví dụ

Hãy để chúng tôi tạo bản đồ bề mặt 3D cho hàm g = xe -(x2 + y2)

Tạo một tệp script và nhập mã sau:

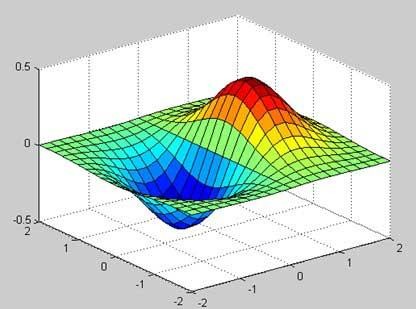
|  |
| --- |
|  |

[x,y]= meshgrid(-2:.2:2);

g = x .\* exp(-x.^2- y.^2); surf(x, y, g)

print-deps graph.eps

Khi bạn chạy tệp, MATLAB hiển thị bản đồ 3-D sau:



Bạn cũng có thể sử dụng lệnh lưới để tạo bề mặt ba chiều. Tuy nhiên, lệnh lướt hiển thị cả các đường kết nối và các mặt của bề mặt có màu, trong khi đó, lệnh lưới tạo ra một bề mặt khung dây với các đường màu kết nối các điểm xác định.

## ĐẠI SỐ HỌC

Cho đến nay , chúng tôi ha v e s een t hà t tất cả các anh ấy không e x ample s work in M AT L AB a s well một s tôi ts G NU , al t erna tiv el yc alled O ctav e . Nhưng để giải các phương trình đại số cơ bản, cả MATLAB và Octave đều có một chút khác biệt, vì vậy chúng tôi sẽ cố gắng đề cập đến MATLAB và Octave trong các phần riêng biệt.

Chúng ta cũng sẽ thảo luận về phân tích thừa số và đơn giản hóa các biểu thức đại số.

Giải các phương trình đại số cơ bản trong MATLAB

Lệnh giải được sử dụng để giải các phương trình đại số. Ở dạng đơn giản nhất, hàm giải quyết lấy phương trình được đặt trong dấu ngoặc kép làm đối số.

Ví dụ: chúng ta hãy giải x trong phương trình x-5 = 0

solve('x-5=0')

|  |
| --- |
|  |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

ans = 5

Bạn cũng có thể gọi hàm giải quyết là:

y = solve('x-5 = 0')

|  |
| --- |
|  |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

y = 5

Bạn thậm chí có thể không tính vế phải của phương trình:

solve('x-5')

|  |
| --- |
|  |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

ans = 5

Nếu phương trình bao gồm nhiều ký hiệu, thì MATLAB theo mặc định giả định rằng bạn đang giải tìm x,

solve(equation, variable)

|  |
| --- |
|  |

tuy nhiên, lệnh giải có dạng khác:

ở đâu, bạn cũng có thể đề cập đến biến.

|  |
| --- |
|  |

Ví dụ, chúng ta hãy giải phương trình v – u – 3t 2 = 0, cho v. Trong trường hợp này, chúng ta nên viết:

solve('v-u-3\*t^2=0','v')

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

ans = 3\*t^2 + u

Giải các phương trình đại số cơ bản trong quãng tám

Lệnh root được sử dụng để giải các phương trình đại số trong Octave và bạn có thể viết các ví dụ trên như sau:

Ví dụ: chúng ta hãy giải x trong phương trình x-5 = 0

roots([1,-5])

|  |
| --- |
|  |

Octave sẽ thực thi câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

ans = 5

Bạn cũng có thể gọi hàm giải quyết là:

y = roots([1,-5])

|  |
| --- |
|  |

Octave sẽ thực thi câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

y = 5

Giải phương trình bậc hai trong MATLAB

Lệnh giải cũng có thể giải các phương trình bậc cao hơn. Nó thường được sử dụng để giải phương trình bậc hai. Hàm trả về nghiệm của phương trình trong một mảng.

Ví dụ sau đây giải phương trình bậc hai x 2 -7x +12 = 0. Tạo một tệp tập lệnh và nhập mã sau:

eq ='x^2 -7\*x + 12 = 0'; s = solve(eq);

disp('The first root is: '), disp(s(1));

disp('The second root is: '), disp(s(2));

|  |
| --- |
|  |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

The first root is:

3

The second root is:

4

Giải phương trình bậc hai trong quãng tám

Ví dụ sau đây giải phương trình bậc hai x 2 -7x +12 = 0 trong Octave. Tạo một tệp script và nhập mã sau:

s = roots([1,-7,12]);

disp('The first root is: '), disp(s(1));

disp('The second root is: '), disp(s(2));

|  |
| --- |
|  |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

The first root is:

4

The second root is:

3

Giải phương trình bậc cao trong MATLAB

Lệnh giải cũng có thể giải các phương trình bậc cao hơn. Ví dụ: chúng ta hãy giải phương trình bậc ba là (x-3) 2 (x- 7) = 0

solve('(x-3)^2\*(x-7)=0')

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

ans = 3

3

7

Trường hợp phương trình bậc cao thì nghiệm dài chứa nhiều số hạng. Bạn có thể nhận được giá trị số của các gốc như vậy bằng cách chuyển đổi chúng thành gấp đôi.

Ví dụ sau giải phương trình bậc 4 x 4−7x 3 + 3x 2−5x + 9 = 0.

Tạo một tệp script và nhập mã sau:

eq ='x^4 - 7\*x^3 + 3\*x^2 - 5\*x + 9 = 0'; s = solve(eq);

disp('The first root is: '), disp(s(1));

disp('The second root is: '), disp(s(2));

disp('The third root is: '), disp(s(3));

disp('The fourth root is: '), disp(s(4));

% converting the roots to double type

disp('Numeric value of first root'), disp(double(s(1))); disp('Numeric value of second root'), disp(double(s(2))); disp('Numeric value of third root'), disp(double(s(3))); disp('Numeric value of fourth root'), disp(double(s(4)));

|  |
| --- |
|  |

|  |
| --- |
|  |

Khi bạn chạy tệp, nó sẽ trả về kết quả như sau:

|  |
| --- |
|  |

The first root is: 6.630396332390718431485053218985

The second root is:

1.0597804633025896291682772499885

The third root is:

- 0.34508839784665403032666523448675 - 1.0778362954630176596831109269793\*i

The fourth root is:

- 0.34508839784665403032666523448675 + 1.0778362954630176596831109269793\*i

Numeric value of first root 6.6304

Numeric value of second root 1.0598

Numeric value of third root

-0.3451 - 1.0778i

Numeric value of fourth root

-0.3451 + 1.0778i

Xin lưu ý rằng hai gốc cuối cùng là số phức.

Giải phương trình bậc cao trong quãng tám

Ví dụ sau đây giải phương trình bậc 4 x 4 − 7x 3 + 3x 2 − 5x + 9 = 0.

Tạo một tệp tập lệnh và nhập mã sau:

|  |
| --- |
|  |

v =[1,-7,3,-5,9];

s = roots(v);

% converting the roots to double type

disp('Numeric value of first root'), disp(double(s(1))); disp('Numeric value of second root'), disp(double(s(2))); disp('Numeric value of third root'), disp(double(s(3))); disp('Numeric value of fourth root'), disp(double(s(4)));

Khi bạn chạy tệp, nó sẽ trả về kết quả như sau:

Numeric value of first root 6.6304

Numeric value of second root

-0.34509 + 1.07784i

Numeric value of third root

-0.34509 - 1.07784i

Numeric value of fourth root 1.0598

|  |
| --- |
|  |

Giải hệ phương trình trong MATLAB

Lệnh giải cũng có thể được sử dụng để tạo nghiệm của các hệ phương trình có nhiều hơn một biến. Hãy để chúng tôi lấy một ví dụ đơn giản để chứng minh việc sử dụng này.

Hãy để chúng tôi giải các phương trình:

5x + 9y = 5

3x – 6y = 4

Tạo một tệp script và nhập mã sau:

|  |
| --- |
|  |

s = solve('5\*x + 9\*y = 5','3\*x - 6\*y = 4'); s.x

s.y

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

ans = 22/19

ans =

-5/57

|  |
| --- |
|  |

Theo cách tương tự, bạn có thể giải các hệ thống tuyến tính lớn hơn. Xét tập hợp các phương trình sau: x + 3y -2z = 5

3x + 5y + 6z = 7 2x + 4y + 3z = 8

Giải hệ phương trình trong quãng tám

Chúng ta có một cách tiếp cận hơi khác để giải một hệ phương trình tuyến tính 'n' với 'n' ẩn số. Hãy để chúng tôi lấy một ví dụ đơn giản để chứng minh việc sử dụng này.

Hãy để chúng tôi giải các phương trình:

5x + 9y = 5

3x – 6y = 4

Một hệ phương trình tuyến tính như vậy có thể được viết dưới dạng phương trình ma trận đơn Ax = b, trong đó A là ma trận hệ số, b là vectơ cột chứa vế phải của phương trình tuyến tính và x là vectơ cột biểu diễn nghiệm như thể hiện trong chương trình dưới đây:

Tạo một tệp script và nhập mã sau:

A =[5,9;3,-6];

b =[5;4];

A \ b

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

ans =

1.157895

-0.087719

Theo cách tương tự, bạn có thể giải các hệ phương trình tuyến tính lớn hơn như dưới đây: x + 3y -2z = 5

3x + 5y + 6z = 7 2x + 4y + 3z = 8

Khai triển và thu thập phương trình trong MATLAB

Lệnh mở rộng và thu thập mở rộng và thu thập một phương trình tương ứng. Ví dụ sau minh họa các khái niệm:

Khi bạn làm việc với nhiều hàm tượng trưng, bạn nên khai báo rằng các biến của bạn là tượng trưng. Tạo một tệp script và nhập mã sau:

syms x %symbolic variable x syms y %symbolic variable x

% expanding equations expand((x-5)\*(x+9))

expand((x+2)\*(x-3)\*(x-5)\*(x+7)) expand(sin(2\*x)) expand(cos(x+y))

% collecting equations collect(x^3\*(x-7)) collect(x^4\*(x-3)\*(x-5))

|  |
| --- |
|  |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

ans =

x^2 + 4\*x - 45 ans =

x^4 + x^3 - 43\*x^2 + 23\*x + 210 ans =

2\*cos(x)\*sin(x) ans =

cos(x)\*cos(y) - sin(x)\*sin(y) ans =

x^4 - 7\*x^3 ans =

x^6 - 8\*x^5 + 15\*x^4

|  |
| --- |
|  |

Khai triển và thu thập phương trình trong quãng tám

Bạn cần có gói tượng trưng , cung cấp lệnh mở rộng và thu thập tương ứng để mở rộng và thu thập một phương trình. Ví dụ sau minh họa các khái niệm:

Khi bạn làm việc với nhiều hàm tượng trưng, bạn nên khai báo rằng các biến của bạn là tượng trưng nhưng Octave có cách tiếp cận khác để xác định các biến tượng trưng. Lưu ý việc sử dụng Sin và Cos , chúng cũng được xác định trong gói tượng trưng.

Tạo một tệp script và nhập mã sau:

% first of all load the package, make sure its installed. pkg load symbolic

% make symbols module available symbols

% define symbolic variables x = sym ('x');

y = sym ('y');

z = sym ('z');

% expanding equations expand((x-5)\*(x+9))

|  |
| --- |
|  |

expand((x+2)\*(x-3)\*(x-5)\*(x+7)) expand(Sin(2\*x)) expand(Cos(x+y))

% collecting equations collect(x^3\*(x-7), z) collect(x^4\*(x-3)\*(x-5), z)

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

ans =

-45.0+x^2+(4.0)\*x

ans =

210.0+x^4-(43.0)\*x^2+x^3+(23.0)\*x

ans =

sin((2.0)\*x) ans =

cos(y+x) ans =

x^(3.0)\*(-7.0+x)

ans =

(-3.0+x)\*x^(4.0)\*(-5.0+x)

|  |
| --- |
|  |

Thừa số và đơn giản hóa đại số Biểu thức

Lệnh thừa số phân tích biểu thức và lệnh đơn giản hóa đơn giản hóa biểu thức. Ví dụ sau minh họa khái niệm:

Ví dụ

Tạo một tệp script và nhập mã sau:

syms x syms y

factor(x^3- y^3) factor([x^2-y^2,x^3+y^3]) simplify((x^4-16)/(x^2-4))

|  |
| --- |
|  |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

ans =

(x - y)\*(x^2 + x\*y + y^2) ans =

[ (x - y)\*(x + y), (x + y)\*(x^2 - x\*y + y^2)] ans =

x^2 + 4

## GIẢI TÍCH

M AT L AB pro v ide sv ariou s way for s ol v ing vấn đề của di ff eren t ial và trong te gral c a culu s , s ol v ing ing ing ing ing s l e gral và phép tính các giới hạn. Hơn hết, bạn có thể dễ dàng vẽ đồ thị của các hàm phức tạp và kiểm tra các điểm cực đại, cực tiểu và các điểm bất thường khác trên đồ thị bằng cách giải hàm gốc cũng như đạo hàm của nó.

Trong chương này và các chương tiếp theo, chúng ta sẽ giải quyết các vấn đề về giải tích. Trong chương này, chúng ta sẽ thảo luận về các khái niệm tiền giải tích, tức là tính giới hạn của hàm số và kiểm chứng các tính chất của giới hạn.

Trong chương tiếp theo Ví phân , chúng ta sẽ tính đạo hàm của một biểu thức và tìm cực đại và cực tiểu cục bộ trên đồ thị. Chúng ta cũng sẽ thảo luận về việc giải các phương trình ví phân.

Cuối cùng, trong chương Tích phân, chúng ta sẽ thảo luận về phép tính tích phân.

Tính toán giới hạn

MATLAB cung cấp lệnh giới hạn để tính các giới hạn. Ở dạng cơ bản nhất, lệnh giới hạn lấy biểu thức làm đối số và tìm giới hạn của biểu thức khi biến độc lập tiến tới 0.

Ví dụ: chúng ta hãy tính giới hạn của hàm f(x) = (x 3 + 5)/(x 4 + 7), vì x có xu hướng bằng không.

syms x limit((x^3+5)/(x^4+7))

|  |
| --- |
|  |

MATLAB sẽ thực thi câu lệnh trên và trả về kết quả như sau kết quả:

|  |
| --- |
|  |

ans = 5/7

Lệnh giới hạn nằm trong lĩnh vực tính toán tượng trưng; bạn cần sử dụng lệnh syms để báo cho MATLAB biết bạn đang sử dụng các biến tượng trưng nào. Bạn cũng có thể tính giới hạn của một hàm, vì biến có xu hướng tiến tới một số khác không. Để tính lim x->a (f(x)) ta dùng lệnh giới hạn có đối số. Đầu tiên là biểu thức và thứ hai là số mà x tiến đến, đây là một .

Ví dụ: chúng ta hãy tính giới hạn của hàm f(x) = (x-3)/(x-1), vì x có xu hướng tiến tới 1.

limit((x -3)/(x-1),1)

|  |
| --- |
|  |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

ans =

NaN

Hãy lấy một ví dụ khác:

limit(x^2+5,3)

|  |
| --- |
|  |

MATLAB sẽ thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

ans = 14

Tính toán giới hạn bằng Octave

Sau đây là phiên bản Octave của ví dụ trên sử dụng gói tượng trưng , hãy thử thực thi và so sánh kết quả:

pkg load symbolic symbols x=sym("x");

subs((x^3+5)/(x^4+7),x,0)

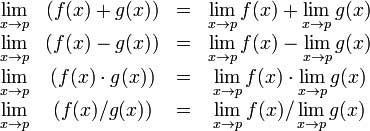
Octave sẽ thực thi câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

ans = 0.7142857142857142857

Xác minh các thuộc tính cơ bản của Hạn mức

Định lý giới hạn đại số cung cấp một số tính chất cơ bản của giới hạn. Đây là như sau:



Chúng ta hãy xem xét hai chức năng:

1. f(x) = (3x + 5)/(x - 3)

2. g(x) = x 2 + 1.

Chúng ta hãy tính giới hạn của các hàm khi x có xu hướng bằng 5, của cả hai hàm và xác minh các tính chất cơ bản của giới hạn bằng cách sử dụng hai hàm này và MATLAB.

Ví dụ

Tạo một tệp script và nhập đoạn mã sau vào đó:

syms x

f =(3\*x +5)/(x-3); g = x^2+1;

|  |
| --- |
|  |

l1 = limit(f,4) l2 = limit (g,4)

lAdd = limit(f + g,4) lSub = limit(f - g,4) lMult = limit(f\*g,4) lDiv = limit (f/g,4)

Khi bạn chạy tệp, nó sẽ hiển thị:

l1 = 17

l2 = 17

lAdd = 34

lSub = 0

lMult = 289

lDiv = 1

|  |
| --- |
|  |

Xác minh các thuộc tính cơ bản của giới hạn bằng Octave

Sau đây là phiên bản Octave của ví dụ trên sử dụng gói tượng trưng , hãy thử thực thi và so sánh kết quả:

pkg load symbolic symbols

x = sym("x");

f =(3\*x +5)/(x-3); g = x^2+1;

l1=subs(f, x,4)

l2 = subs (g, x,4) lAdd = subs (f+g, x,4) lSub = subs (f-g, x,4)

lMult = subs (f\*g, x,4) lDiv = subs (f/g, x,4)

Octave sẽ thực thi câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

l1 =

17.0

l2 =

17.0

lAdd =

34.0

lSub =

0.0

lMult =

289.0

lDiv =

1.0

Giới hạn bên trái và bên phải

Khi nào một chức năng có một sự gián đoạn vì một số cụ thể giá trị của các Biến đổi, các giới hạn làm không phải hiện hữu tại điều đó điểm. Nói cách khác, giới hạn của hàm số f(x) gián đoạn tại x = a, khi giá trị của giới hạn khi x tiến dần đến x từ vế trái không bằng giá trị của giới hạn khi x tiến dần từ vế phải cạnh.

Điều này dẫn đến khái niệm giới hạn thuận tay trái và thuận tay phải. Giới hạn thuận tay trái được định nghĩa là giới hạn khi x -> a, từ bên trái, tức là x tiến dần đến a, với các giá trị của x < a. Giới hạn bên phải được định nghĩa là giới hạn x -> a, từ bên phải, tức là x tiến dần đến a, với các giá trị của x > a. Khi giới hạn tay trái và giới hạn tay phải không bằng nhau, giới hạn không tồn tại.

Chúng ta hãy xem xét một chức năng:

f(x) = (x - 3)/|x - 3|

Ta sẽ chứng minh lim x->3 f(x) không tồn tại. MATLAB giúp chúng tôi thiết lập thực tế này theo hai cách:

Bằng cách vẽ đồ thị của hàm số và chỉ ra sự gián đoạn

Bằng cách tính toán các giới hạn và chỉ ra rằng cả hai đều khác biệt.

Các giới hạn thuận tay trái và thuận tay phải được tính toán bằng cách chuyển các chuỗi ký tự 'trái' và 'phải' cho lệnh giới hạn làm đối số cuối cùng.

Ví dụ

Tạo một tệp script và nhập đoạn mã sau vào đó:

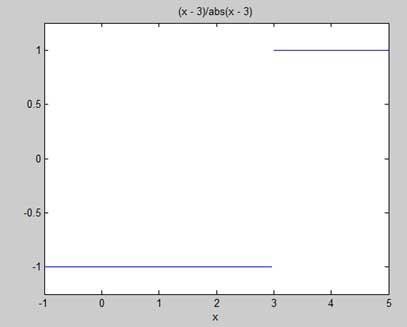
f =(x -3)/abs(x-3);

ezplot(f,[-1,5])

l = limit(f,x,3,'left')

r = limit(f,x,3,'right')

Khi bạn chạy tệp, MATLAB sẽ vẽ biểu đồ sau,



và hiển thị đầu ra sau:

l =

-1

r = 1

|  |
| --- |
|  |

## SỰ KHÁC BIỆT

M AT L AB pro v ide s t he lệnh khác cho biểu tượng máy tính boli c deri v a t i v e s . \_ tôi n i t ss imple s t form , bạn ou pa ss hàm bạn muốn phân biệt thành lệnh diff làm đối số.

Ví dụ: chúng ta hãy tính đạo hàm của hàm f(t) = 3t 2 + 2t -2

Ví dụ

Tạo một tệp script và nhập đoạn mã sau vào đó:

syms t

f =3\*t^2+2\*t^(-2); diff(f)

|  |
| --- |
|  |

Khi đoạn mã trên được biên dịch và thực thi, nó sẽ tạo ra kết quả như sau:

|  |
| --- |
|  |

ans =

6\*t - 4/t^3

Sau đây là Octave tương đương với phép tính trên:

pkg load symbolic symbols

t = sym("t");

f =3\*t^2+2\*t^(-2); differentiate(f,t)

|  |
| --- |
|  |

Octave thực thi mã và trả về kết quả sau:

|  |
| --- |
|  |

ans =

-(4.0)\*t^(-3.0)+(6.0)\*t

Xác minh các quy tắc cơ bản của khác biệt hóa

Hãy để chúng tôi phát biểu ngắn gọn các phương trình hoặc quy tắc khác nhau để phân biệt các chức năng và kiểm tra các quy tắc này. Với mục đích này, chúng ta sẽ viết f'(x) cho đạo hàm cấp một và f"(x) cho đạo hàm cấp hai.

Sau đây là các quy tắc để phân biệt:

QUY TẮC 1

Đối với mọi hàm f và g và mọi số thực a và b, đạo hàm của hàm:

h(x) = af(x) + bg(x) đối với x được cho bởi:

h'(x) = af'(x) + bg'(x)

QUY TẮC 2

Quy tắc tổng và trừ nêu rõ rằng nếu f và g là hai hàm số, thì f' và g' lần lượt là đạo hàm của chúng, thì,

( f + g)' = f' + g'

(f - g)' = f' - g'

QUY TẮC 3

Quy tắc tích phát biểu rằng nếu f và g là hai hàm thì f' và g' lần lượt là đạo hàm của chúng, thì,

(fg)' = f'.g + g'.f

QUY TẮC 4

Quy tắc thương số phát biểu rằng nếu f và g là hai hàm số thì f' và g' lần lượt là đạo hàm của chúng, thì,

(f/g)' = (f'.g - g'.f)/g 2

QUY TẮC 5

Quy tắc lũy thừa cơ bản hoặc đa thức phát biểu rằng, nếu y = f(x) = x n , thì f' = n. x (n-1)

Kết quả trực tiếp của quy tắc này là đạo hàm của bất kỳ hằng số nào bằng 0, nghĩa là, nếu y = k , bất kỳ hằng số nào, thì

f' = 0

QUY TẮC 6

Quy tắc dây chuyền phát biểu rằng, Đạo hàm của hàm h(x) = f(g(x)) đối với x là,

h'(x)= f'(g(x)).g'(x)

Ví dụ

Tạo một tệp script và nhập đoạn mã sau vào đó:

syms x syms t

f =(x +2)\*(x^2+3) der1 = diff(f)

f =(t^2+3)\*(sqrt(t)+ t^3) der2 = diff(f)

f =(x^2-2\*x +1)\*(3\*x^3-5\*x^2+2) der3 = diff(f)

f =(2\*x^2+3\*x)/(x^3+1) der4 = diff(f)

f =(x^2+1)^17 der5 = diff(f)

f =(t^3+3\* t^2+5\*t -9)^(-6) der6 = diff(f)

|  |
| --- |
|  |

Khi bạn chạy tệp, MATLAB sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

f =

(x^2 + 3)\*(x + 2)

der1 =

2\*x\*(x + 2) + x^2 + 3

f =

(t^(1/2) + t^3)\*(t^2 + 3)

der2 =

(t^2 + 3)\*(3\*t^2 + 1/(2\*t^(1/2))) + 2\*t\*(t^(1/2) + t^3)

f =

(x^2 - 2\*x + 1)\*(3\*x^3 - 5\*x^2 + 2)

der3 =

(2\*x - 2)\*(3\*x^3 - 5\*x^2 + 2) - (- 9\*x^2 + 10\*x)\*(x^2 - 2\*x + 1)

f =

(2\*x^2 + 3\*x)/(x^3 + 1)

der4 =

(4\*x + 3)/(x^3 + 1) - (3\*x^2\*(2\*x^2 + 3\*x))/(x^3 + 1)^2

f =

(x^2 + 1)^17

der5 =

34\*x\*(x^2 + 1)^16

f =

1/(t^3 + 3\*t^2 + 5\*t - 9)^6

der6 =

-(6\*(3\*t^2 + 6\*t + 5))/(t^3 + 3\*t^2 + 5\*t - 9)^7

Sau đây là Octave tương đương với phép tính trên:

pkg load symbolic symbols x=sym("x");

t=sym("t");

f =(x +2)\*(x^2+3)

der1 = differentiate(f,x) f =(t^2+3)\*(t^(1/2)+ t^3) der2 = differentiate(f,t)

f =(x^2-2\*x +1)\*(3\*x^3-5\*x^2+2) der3 = differentiate(f,x)

f =(2\*x^2+3\*x)/(x^3+1) der4 = differentiate(f,x) f =(x^2+1)^17

der5 = differentiate(f,x) f =(t^3+3\* t^2+5\*t -9)^(-6) der6 = differentiate(f,t)

|  |
| --- |
|  |

Đạo hàm của hàm số mũ, logarit và lượng giác

Bảng sau đây cung cấp đạo hàm của các hàm số mũ, logarit và lượng giác thường được sử dụng:

|  |  |
| --- | --- |
| Chức năng | Phát sinh |
| rìu c | c ax .ln ca (ln là logarit tự nhiên) |
| e x | e x |
| ln x | 1/x |
| ln c x | 1/x.ln c |
| x x | x x .(1 + ln x) |
| tội lỗi(x) | cos(x) |
| cos(x) | -sin(x) |
| rám nắng(x) | sec 2 (x), hoặc 1/cos 2 (x), hoặc 1 + tan 2 (x) |
| cũi(x) | -csc 2 (x), hoặc -1/sin 2 (x), hoặc -(1 + cot 2 (x)) |
| giây(x) | giây(x).tan(x) |
| csc(x) | -csc(x).cot(x) |

Ví dụ

Tạo một tệp script và nhập đoạn mã sau vào đó:

syms x

y = exp(x) diff(y)

y = x^9 diff(y)

y = sin(x) diff(y)

y = tan(x) diff(y)

y = cos(x) diff(y)

y = log(x) diff(y)

y = log10(x) diff(y)

y = sin(x)^2 diff(y)

y = cos(3\*x^2+2\*x +1) diff(y)

y = exp(x)/sin(x) diff(y)

|  |
| --- |
|  |

Khi bạn chạy tệp, MATLAB sẽ hiển thị kết quả sau:

y = exp(x) ans = exp(x)

y = x^9

ans = 9\*x^8

y = sin(x) ans = cos(x)

y = tan(x)

ans = tan(x)^2 + 1

y = cos(x) ans =

-sin(x)

y = log(x) ans = 1/x

y = log(x)/log(10) ans = 1/(x\*log(10))

y = sin(x)^2

ans = 2\*cos(x)\*sin(x)

y =

cos(3\*x^2 + 2\*x + 1) ans =

-sin(3\*x^2 + 2\*x + 1)\*(6\*x + 2)

y = exp(x)/sin(x) ans =

exp(x)/sin(x) - (exp(x)\*cos(x))/sin(x)^2

Sau đây là Octave tương đương với phép tính trên:

pkg load symbolic symbols

x = sym("x"); y =Exp(x)

differentiate(y,x)

y = x^9 differentiate(y,x)

y =Sin(x) differentiate(y,x)

y =Tan(x) differentiate(y,x)

y =Cos(x) differentiate(y,x)

y =Log(x) differentiate(y,x)

% symbolic packages does not have this support

%y =Log10(x)

%differentiate(y,x)

y =Sin(x)^2 differentiate(y,x)

y =Cos(3\*x^2+2\*x +1) differentiate(y,x)

y =Exp(x)/Sin(x) differentiate(y,x)

Tính toán các công cụ phái sinh bậc cao hơn

Để tính các đạo hàm cấp cao hơn của hàm f, chúng ta sử dụng cú pháp diff(f,n) . Hãy tính đạo hàm cấp hai của hàm y = f(x) = x .e -3x

|  |
| --- |
|  |

f = x\*exp(-3\*x); diff(f,2)

MATLAB thực thi mã và trả về kết quả sau:

ans =

9\*x\*exp(-3\*x) - 6\*exp(-3\*x)

|  |
| --- |
|  |

Sau đây là Octave tương đương với phép tính trên:

|  |
| --- |
|  |

pkg load symbolic symbols

x = sym("x");

f = x\*Exp(-3\*x);

differentiate(f, x,2)

Ví dụ

Trong ví dụ này, chúng ta hãy giải quyết một vấn đề. Cho rằng hàm số y = f(x) = 3 sin(x) + 7 cos(5x) . Chúng ta sẽ phải tìm xem liệu phương trình f" + f = -5cos(2x) có đúng hay không.

Tạo một tệp script và nhập đoạn mã sau vào nó:

syms x

y =3\*sin(x)+7\*cos(5\*x);% defining the function

lhs = diff(y,2)+y;%evaluting the lhs of the equation rhs =-5\*cos(2\*x);%rhs of the equation if(isequal(lhs,rhs))

disp('Yes, the equation holds true'); else

disp('No, the equation does not hold true');

end

disp('Value of LHS is: '), disp(lhs);

|  |
| --- |
|  |

Khi bạn chạy tập tin, nó sẽ hiển thị như sau kết quả:

|  |
| --- |
|  |

No, the equation does not hold true Value of LHS is:

-168\*cos(5\*x)

Sau đây là Octave tương đương với phép tính trên:

pkg load symbolic symbols

x = sym("x");

y =3\*Sin(x)+7\*Cos(5\*x);% defining the function

lhs = differentiate(y, x,2)+ y;%evaluting the lhs of the equation rhs =-5\*Cos(2\*x);%rhs of the equation

if(lhs == rhs)

disp('Yes, the equation holds true'); else

disp('No, the equation does not hold true');

end

disp('Value of LHS is: '), disp(lhs);

|  |
| --- |
|  |

Tìm Cực đại và Cực tiểu của một đường cong

Nếu chúng ta đang tìm kiếm cực đại và cực tiểu cục bộ cho một đồ thị, thì về cơ bản, chúng ta đang tìm kiếm các điểm cao nhất hoặc thấp nhất trên đồ thị của hàm tại một vị trí cụ thể hoặc cho một phạm vigiá trị cụ thể của biến ký hiệu.

Đối với hàm số y = f(x) các điểm trên đồ thị mà đồ thị có hệ số góc bằng 0 được gọi là các điểm bất động . Nói cách khác, các điểm dừng tại đó f'(x) = 0.

Để tìm các điểm dừng của một hàm mà chúng ta phân biệt, chúng ta cần đặt đạo hàm bằng 0 và giải phương trình.

Ví dụ

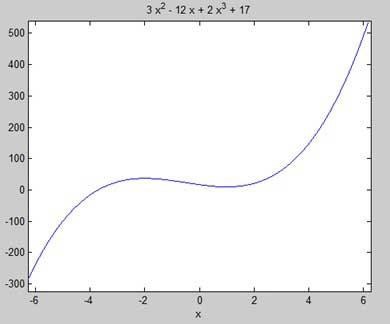
Hãy để chúng tôi tìm các điểm dừng của hàm f(x) = 2x 3 + 3x 2 − 12x + 17 Thực hiện các bước sau:

|  |
| --- |
|  |

Đầu tiên chúng ta hãy nhập hàm và vẽ đồ thị của nó biểu đồ:

syms x

y =2\*x^3+3\*x^2-12\*x +17;% defining the function ezplot(y)

MATLAB thực thi mã và trả về biểu đồ sau:

Đây là mã tương đương Octave cho ví dụ trên:

|  |
| --- |
|  |

pkg load symbolic symbols

x = sym('x');

y =inline("2\*x^3 + 3\*x^2 - 12\*x + 17");

ezplot(y)

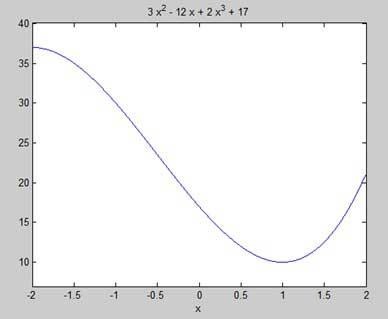
print-deps graph.eps

Mục đích của chúng ta là tìm một số cực đại và cực tiểu cục bộ trên đồ thị, vì vậy chúng ta hãy tìm cực đại và cực tiểu cục bộ cho khoảng [-2, 2] trên đồ thị.

syms x

y =2\*x^3+3\*x^2-12\*x +17;% defining the function ezplot(y,[-2,2])

MATLAB thực thi mã và trả về biểu đồ sau:



Đây là mã tương đương Octave cho ví dụ trên:

pkg load symbolic symbols

x = sym('x');

y =inline("2\*x^3 + 3\*x^2 - 12\*x + 17");

ezplot(y,[-2,2]) print-deps graph.eps

|  |
| --- |
|  |

Tiếp theo, chúng ta hãy tính toán phát sinh

g = diff(y)

MATLAB thực thi mã và trả về kết quả sau:

g =

6\*x^2 + 6\*x - 12

|  |
| --- |
|  |

Đây là Octave tương đương với phép tính trên:

|  |
| --- |
|  |

pkg load symbolic symbols

x = sym("x");

y =2\*x^3+3\*x^2-12\*x +17; g = differentiate(y,x)

Hãy để chúng tôi giải hàm đạo hàm, g, để có được các giá trị mà nó trở thành số không.

s = solve(g)

MATLAB thực thi mã và trả về kết quả sau:

|  |
| --- |
|  |

s =

1

-2

Sau đây là Octave tương đương với phép tính trên:

pkg load symbolic symbols

x = sym("x");

y =2\*x^3+3\*x^2-12\*x +17; g = differentiate(y,x) roots([6,6,-12])

|  |
| --- |
|  |

Điều này phù hợp với cốt truyện của chúng tôi. Vì vậy, hãy để chúng tôi đánh giá hàm f tại các điểm giới hạn x = 1, -2. Chúng ta có thể thay thế một giá trị trong một hàm tượng trưng bằng cách sử dụng subs chỉ huy.

|  |
| --- |
|  |

subs(y,1), subs(y,-2)

MATLAB thực thi mã và trả về kết quả sau:

ans = 10

ans = 37

|  |
| --- |
|  |

Sau đây là Octave tương đương với phép tính trên:

|  |
| --- |
|  |

pkg load symbolic symbols

x = sym("x");

y =2\*x^3+3\*x^2-12\*x +17; g = differentiate(y,x)

roots([6,6,-12])

subs(y, x,1), subs(y, x,-2)

Do đó, Các giá trị nhỏ nhất và lớn nhất của hàm số f(x) = 2x 3 + 3x 2 − 12x + 17, trong khoảng [-2,2] là 10 và 37.

Giải phương trình ví phân

MATLAB cung cấp lệnh dsolve để giải các phương trình ví phân một cách tượng trưng.

Dạng cơ bản nhất của lệnh dsolve để tìm nghiệm cho một phương trình là:

dsolve('eqn')

trong đó eqn là một chuỗi văn bản được sử dụng để nhập phương trình.

Nó trả về một giải pháp tượng trưng với một tập hợp các hằng số tùy ý mà MATLAB đánh dấu C1, C2, v.v.

Bạn cũng có thể chỉ định điều kiện ban đầu và điều kiện biên cho bài toán, dưới dạng danh sách được phân cách bằng dấu phẩy theo sau phương trình như:

dsolve('eqn','cond1','cond2',…)

Với mục đích sử dụng lệnh dsolve, các đạo hàm được biểu thị bằng chữ D . Ví dụ, một phương trình như f'(t)

= -2\*f + cost(t) được nhập là:

'Df = -2\*f + cos(t)'

Các đạo hàm cao hơn được biểu thị bằng cách theo sau D theo thứ tự của đạo hàm. Ví dụ: phương trình f"(x) + 2f'(x) = 5sin3x phải được nhập là:

'D2y + 2Dy = 5\*sin(3\*x)'

Chúng ta hãy lấy một ví dụ đơn giản về phương trình ví phân cấp một: y' = 5y.

s = dsolve('Dy = 5\*y')

MATLAB thực thi mã và trả về kết quả sau:

|  |
| --- |
|  |

s = C2\*exp(5\*t)

Chúng ta hãy lấy một ví dụ khác về phương trình ví phân cấp hai như sau: y" - y = 0, y(0) = -1, y'(0) = 2.

dsolve('D2y - y = 0','y(0) = -1','Dy(0) = 2')

MATLAB thực thi mã và trả về kết quả sau:

|  |
| --- |
|  |

ans =

exp(t)/2 - (3\*exp(-t))/2

## HỘI NHẬP

Trong loại đầu tiên, đạo hàm của một hàm được cho trước và chúng ta muốn tìm hàm đó. Do đó, về cơ bản chúng ta đảo ngược quá trình phân hóa. Quá trình ngược lại này được gọi là chống phân biệt, hoặc tìm hàm nguyên thủy hoặc tìm một hàm không xác định . tích phân .

Loại bài toán thứ hai liên quan đến việc cộng một số lượng rất lớn với các đại lượng rất nhỏ rồi lấy giới hạn khi kích thước của các đại lượng tiến dần đến 0, trong khi số hạng có xu hướng vô cùng. Quá trình này dẫn đến việc xác định định nghĩa tích phân .

Tích phân xác định được dùng để tìm diện tích, thể tích, trọng tâm, mômen quán tính, công do lực thực hiện và trong nhiều ứng dụng khác.

Tìm tích phân không xác định bằng MATLAB

Theo định nghĩa, nếu đạo hàm của hàm f(x) là f'(x), thì ta nói rằng tích phân bất định của f'(x) theo x là f(x). Ví dụ, vì đạo hàm (đối với x) của x 2 là 2x, nên chúng ta có thể nói rằng tích phân không xác định của 2x là x 2 .

Trong các ký hiệu:

f'(x 2 ) = 2x , do đó,

∫ 2xdx = x 2 .

Tích phân bất định không phải là duy nhất, bởi vì đạo hàm của x 2 + c, với bất kỳ giá trị nào của hằng số c, cũng sẽ là 2x. Điều này được thể hiện bằng các ký hiệu như:

∫ 2xdx = x 2 + c .

Trong đó, c được gọi là 'tùy ý hằng số'.

MATLAB cung cấp một lệnh int để tính tích phân của một biểu thức. Để rút ra một biểu thức cho tích phân không xác định của một hàm, chúng ta viết:

int(f);

Ví dụ: từ ví dụ trước của chúng tôi:

|  |
| --- |
|  |

syms x

int(2\*x)

MATLAB thực hiện câu lệnh trên và trả về kết quả như sau:

ans = x^2

|  |
| --- |
|  |

ví dụ 1

Trong ví dụ này, chúng ta hãy tìm tích phân của một số biểu thức thường dùng. Tạo một tệp script và nhập mã sau vào đó:

syms x n int(sym(x^n))

f ='sin(n\*t)'

int(sym(f)) syms a t

int(a\*cos(pi\*t)) int(a^x)

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

ans =

piecewise([n == -1, log(x)], [n ~= -1, x^(n + 1)/(n + 1)]) f =

sin(n\*t) ans =

-cos(n\*t)/n ans =

(a\*sin(pi\*t))/pi ans =

a^x/log(a)

ví dụ 2

Tạo một tệp script và nhập mã sau vào đó:

syms x n int(cos(x))

int(exp(x))

int(log(x))

int(x^-1) int(x^5\*cos(5\*x)) pretty(int(x^5\*cos(5\*x))) int(x^-5)

int(sec(x)^2)

pretty(int(1-10\*x +9\* x^2)) int((3+5\*x -6\*x^2-7\*x^3)/2\*x^2)

pretty(int((3+5\*x -6\*x^2-7\*x^3)/2\*x^2))

|  |
| --- |
|  |

Lưu ý rằng lệnh đẹp trả về một biểu thức ở định dạng dễ đọc hơn. Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

ans = sin(x)

trả lời =

hết hạn(x)

trả lời =

x\*(log(x) - 1)

trả lời =

nhật ký(x)

trả lời =

(24\*cos(5\*x))/3125 + (24\*x\*sin(5\*x))/625 - (12\*x^2\*cos(5\*x))/125 + (x^4\* cos(5\*x))/5

- (4\*x^3\*sin(5\*x))/25 + (x^5\*sin(5\*x))/5

2 4

24 cos(5 x) 24 x tội lỗi (5 x) 12 xcos (5 x) x cos(5 x)

+ - + -

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3125 |  | 625 |  | 125 |  | 5 |
| 3 |  | 5 |  |  |  |  |

4 x sin(5 x) x sin(5 x)

+ 25 5

trả lời =

-1/(4\*x^4)

ans = tan(x)

2

x (3 x - 5 x + 1)

trả lời =

- (7\*x^6)/12 - (3\*x^5)/5 + (5\*x^4)/8 + x^3/2

6 5 4 3

7 x 3 x 5 x x

- - + +

12 5 8 2

Tìm tích phân xác định bằng MATLAB

Theo định nghĩa, tích phân xác định về cơ bản là giới hạn của một tổng. Chúng ta sử dụng tích phân xác định để tìm các diện tích chẳng hạn như diện tích giữa một đường cong và trục x và diện tích giữa hai đường cong. Tích phân xác định cũng có thể được sử dụng trong các tình huống khác, trong đó đại lượng cần thiết có thể được biểu thị dưới dạng giới hạn của một tổng.

Lệnh int có thể được sử dụng để tích phân xác định bằng cách vượt qua các giới hạn mà bạn muốn tính tích phân.

Definite IntegralTính toán

chúng tôi viết,

int(x, a, b)

ExampleVí dụ, để tính toán giá trị của chúng tôi viết:

int(x,4,9)

MATLAB thực hiện câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

ans = 65/2

Sau đây là Octave tương đương với phép tính trên:

pkg load symbolic symbols

x = sym("x"); f = x;

c =[1,0];

integral = polyint(c);

a = polyval(integral,9)- polyval(integral,4); display('Area: '), disp(double(a));

|  |
| --- |
|  |

Một giải pháp thay thế có thể được đưa ra bằng cách sử dụng hàm quad() do Octave cung cấp như sau:

pkg load symbolic symbols

f =inline("x");

[a, ierror, nfneval]= quad(f,4,9);

display('Area: '), disp(double(a));

ví dụ 1

Chúng ta hãy tính diện tích bao quanh giữa trục x và đường cong y = x 3 −2x+5 và tọa độ x = 1 và x = 2.

Diện tích cần thiết được cho bởi:

Area Calculation

Tạo một tệp script và nhập mã sau:

f = x^3-2\*x +5; a =int(f,1,2)

display('Area: '), disp(double(a));

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

a = 23/4

Area:

5.7500

|  |
| --- |
|  |

Sau đây là Octave tương đương với phép tính trên:

|  |
| --- |
|  |

pkg load symbolic symbols

x = sym("x");

f = x^3-2\*x +5; c =[1,0,-2,5];

integral = polyint(c);

a = polyval(integral,2)- polyval(integral,1); display('Area: '), disp(double(a));

Một giải pháp thay thế có thể được đưa ra bằng cách sử dụng hàm quad() do Octave cung cấp như sau:

pkg load symbolic symbols

x = sym("x");

f =inline("x^3 - 2\*x +5");

[a, ierror, nfneval]= quad(f,1,2); display('Area: '), disp(double(a));

|  |
| --- |
|  |

Ví dụ 2

Tìm diện tích bên dưới đường cong: f(x) = x 2 cos(x) với −4 ≤ x ≤ 9. Tạo một tệp kịch bản và viết đoạn mã sau:

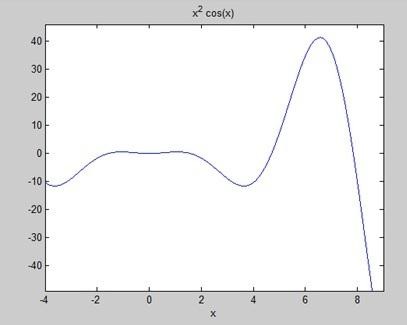
|  |
| --- |
|  |

f = x^2\*cos(x);

ezplot(f,[-4,9])

a =int(f,-4,9)

disp('Area: '), disp(double(a));

Khi bạn chạy tệp, MATLAB vẽ đồ thị:

và hiển thị kết quả sau:

|  |
| --- |
|  |

a =

8\*cos(4) + 18\*cos(9) + 14\*sin(4) + 79\*sin(9) Area:

0.3326

Sau đây là Octave tương đương với phép tính trên:

pkg load symbolic symbols

x = sym("x");

f =inline("x^2\*cos(x)");

ezplot(f,[-4,9]) print-deps graph.eps

[a, ierror, nfneval]= quad(f,-4,9);

display('Area: '), disp(double(a));

|  |
| --- |
|  |

## ĐA THỨC

M AT L AB đại diện cho đa thức s a s hàng v e ctors c on t aing c oe ff i cien ts \_ \_ \_ ra lệnh b y de sc kết thúc quyền hạn . Ví dụ: phương trình P(x) = x 4 + 7x 3 - 5x + 9 có thể được biểu diễn dưới dạng:

p = [1 7 0 -5 9];

Đánh giá đa thức

Hàm đa giá trị được sử dụng để đánh giá một đa thức ở một giá trị đã chỉ định. Ví dụ: để đánh giá đa thức p trước đây của chúng tôi , tại x = 4, loại:

p =[170-59];

polyval(p,4)

MATLAB thực hiện các câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

ans = 693

MATLAB cũng cung cấp hàm polyvalm để đánh giá đa thức ma trận. Một đa thức ma trận là một đa thức với các ma trận là các biến.

Ví dụ, chúng ta hãy tạo một ma trận vuông X và đánh giá đa thức p, tại X:

p =[170-59];

X =[12-34;2-563;3102;5-738];

polyvalm(p, X)

MATLAB thực hiện các câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| trả lời = |  | | | |
|  | 2307 | -1769 | -939 | 4499 |
|  | 2314 | -2376 | -249 | 4695 |
|  | 2256 | -1892 | -549 | 4310 |
|  | 4570 | -4532 | -1062 | 9269 |

Tìm nghiệm của đa thức

Hàm root tính toán nghiệm của đa thức. Ví dụ, để tính nghiệm của đa thức p, hãy gõ:

p =[170-59];

r = roots(p)

|  |
| --- |
|  |

MATLAB thực hiện các câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

r =

-6.8661 + 0.0000i

-1.4247 + 0.0000i

0.6454 + 0.7095i

0.6454 - 0.7095i

Hàm poly là một hàm nghịch đảo của hàm gốc và trả về các hệ số đa thức.

p2 = poly(r)

Ví dụ:

MATLAB thực hiện các câu lệnh trên và trả về kết quả như sau:

|  |
| --- |
|  |

p2 = 1.0000

7.0000

0.0000 -5.0000

9.0000

Lắp đường cong đa thức

Hàm polyfit tìm các hệ số của một đa thức phù hợp với một tập hợp dữ liệu theo nghĩa bình phương nhỏ nhất. Nếu x và y là hai vectơ chứa dữ liệu x và y được khớp với đa thức bậc n, thì chúng ta có được đa thức khớp với dữ liệu bằng cách viết:

p = polyfit(x,y,n)

|  |
| --- |
|  |

Ví dụ

Tạo một tệp script và nhập mã sau:

x =[123456]; y =[5.543.1128290.7498.4978.67];%data

p = polyfit(x,y,4)%get the polynomial

%Compute the values of the polyfit estimate over a finer range,

%and plot the estimate over the real data values for comparison: x2 =1:.1:6;

y2 = polyval(p,x2); plot(x,y,'o',x2,y2) grid on

|  |
| --- |
|  |

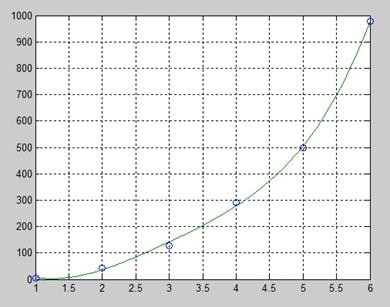
Khi bạn chạy tệp, MATLAB sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

p =

4.1056 -47.9607 222.2598 -362.7453 191.1250

Và vẽ đồ thị sau:



## BIẾN ĐỔI

M AT L AB pro v ide sc omand f hoặc làm việc wi th tr an forms , s u c h một anh ấy \_ Lapla c e và Fo urier tran sforms . Biến đổi được sử dụng trong khoa học và kỹ thuật như một công cụ để đơn giản hóa việc phân tích và xem xét dữ liệu từ một góc độ khác.

Ví dụ, phép biến đổi Fourier cho phép chúng ta chuyển đổi tín hiệu được biểu diễn dưới dạng hàm thời gian thành hàm tần số. Biến đổi Laplace cho phép chúng ta chuyển đổi phương trình ví phân thành phương trình đại số.

MATLAB cung cấp các lệnh laplace , Fourier và fft để làm việc với các phép biến đổi Laplace, Fourier và Fast Fourier.

Biến đổi Laplace

Biến đổi Laplace của hàm thời gian f(t) được cho bởi tích phân sau:

Laplace Transform

Biến đổi Laplace còn được ký hiệu là biến đổi của f(t) thành F(s). Bạn có thể thấy quá trình biến đổi hoặc tích phân này chuyển f(t), một hàm của biến tượng trưng t, thành một hàm F(s) khác, với một biến s khác.

Biến đổi Laplace biến phương trình ví phân thành phương trình đại số. Để tính biến đổi Laplace của hàm f(t), hãy viết:

laplace(f(t))

Ví dụ

Trong ví dụ này, chúng ta sẽ tính biến đổi Laplace của một số hàm thường được sử dụng. Tạo một tệp script và nhập mã sau:

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

ans = 1/s^2

ans = 2/s^3

ans = 362880/s^10

ans = 1/(b + s)

ans =

w/(s^2 + w^2)

ans =

s/(s^2 + w^2)

|  |
| --- |
|  |

### Biến đổi Laplace nghịch đảo

MATLAB cho phép chúng ta tính toán phép biến đổi Laplace nghịch đảo bằng cách sử dụng lệnh ilaplace.

Ví dụ:

|  |
| --- |
|  |

ilaplace(1/s^3)

MATLAB sẽ thực hiện câu lệnh trên và hiển thị kết quả:

ans = t^2/2

|  |
| --- |
|  |

Ví dụ

Tạo một tệp script và nhập mã sau:

syms s t a b w ilaplace(1/s^7) ilaplace(2/(w+s)) ilaplace(s/(s^2+4)) ilaplace(exp(-b\*t)) ilaplace(w/(s^2+ w^2)) ilaplace(s/(s^2+ w^2))

|  |
| --- |
|  |

Khi bạn chạy tệp, nó sẽ hiển thị kết quả sau:

|  |
| --- |
|  |

ans = t^6/720

ans = 2\*exp(-t\*w)

ans = cos(2\*t)

ans =

ilaplace(exp(-b\*t), t, x)

ans = sin(t\*w)

ans = cos(t\*w)

Phép biến đổi Fourier

Biến đổi Fourier thường biến đổi một hàm toán học của thời gian, f(t), thành một hàm mới, đôi khi được ký hiệu là hoặc F, có đối số là tần số với đơn vị là chu kỳ/s (hertz) hoặc radian trên giây. Khi đó hàm mới được gọi là biến đổi Fourier và/hoặc phổ tần số của hàm f.

Ví dụ

Tạo một tệp script và nhập mã sau vào đó:

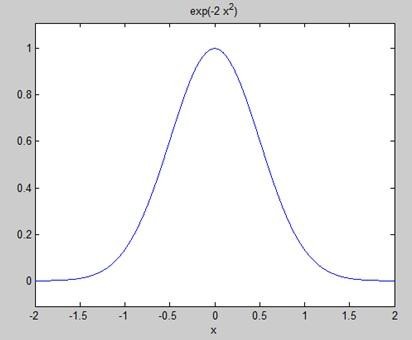
syms x

f = exp(-2\*x^2);%ourfunction ezplot(f,[-2,2])% plot of ourfunction FT = fourier(f) %Fourier transform

|  |
| --- |
|  |

Khi bạn chạy tệp, MATLAB vẽ biểu đồ sau:

|  |
| --- |
|  |



Và hiển thị như sau kết quả:

FT =

(2^(1/2)\*pi^(1/2)\*exp(-w^2/8))/2

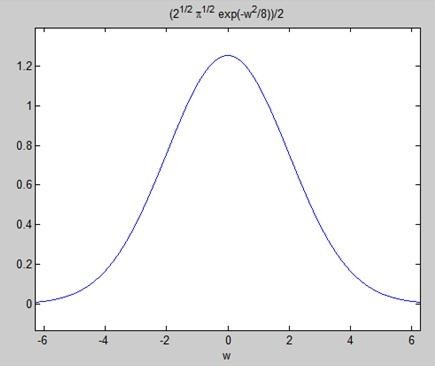
|  |
| --- |
|  |

Vẽ đồ thị biến đổi Fourier như:

|  |
| --- |
|  |

ezplot(FT)

Cho đồ thị sau:



Biến đổi Fourier nghịch đảo

MATLAB cung cấp lệnh ifourier để tính biến đổi Fourier ngược của một hàm.

f = ifourier(-2\*exp(-abs(w)))

Ví dụ:

MATLAB sẽ thực hiện câu lệnh trên và hiển thị kết quả:

|  |
| --- |
|  |

f =

-2/(pi\*(x^2 + 1))

## GNU QUÃNG TÁM

G N U Ô c ta v e tôi là một hi g h -le v el prog r a m m i n g l a ng ua ge tôi k e \_ M A T L AB và tôi t tôi là m o s t l y c o m pa t i ble với MATLAB . \_ Nó cũng được sử dụng cho số tính toán.

Octave có các tính năng chung sau với MATLAB:

ma trận là dữ liệu cơ bản loại

nó có hỗ trợ tích hợp cho phức tạp con số

nó có các hàm toán học tích hợp và thư víện

nó hỗ trợ các chức năng do người dùng định nghĩa

GNU Octave cũng là phần mềm có thể phân phối lại tự do. Bạn có thể phân phối lại và/hoặc sửa đổi nó theo các điều khoản của Giấy phép Công cộng GNU (GPL) do Tổ chức Phần mềm Tự do xuất bản.

MATLAB so với Octave

Hầu hết các chương trình MATLAB chạy trong Octave, nhưng một số chương trình Octave có thể không chạy trong MATLAB vì Octave cho phép một số cú pháp mà MATLAB không cho phép.

Ví dụ: MATLAB chỉ hỗ trợ dấu ngoặc đơn, nhưng Octave hỗ trợ cả dấu ngoặc đơn và dấu ngoặc kép để xác định chuỗi. Nếu bạn đang tìm hướng dẫn về Octave, thì vui lòng xem qua hướng dẫn này từ đầu bao gồm cả MATLAB cũng như Octave.

VÍ DỤ TƯƠNG THÍCH

Hầu như tất cả các ví dụ trong hướng dẫn này đều tương thích với MATLAB cũng như Octave. Hãy thử ví dụ sau trong MATLAB và Octave, ví dụ này tạo ra kết quả tương tự mà không có bất kỳ thay đổi cú pháp nào:

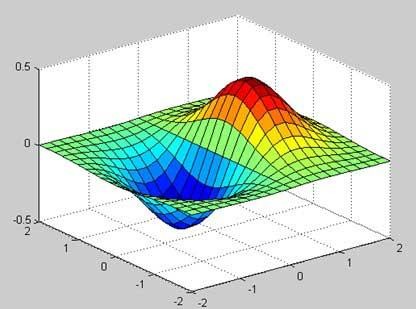
Ví dụ này tạo bản đồ bề mặt 3D cho hàm g = xe -(x2 + y2) . Tạo một tệp script và nhập mã sau:

[x,y]= meshgrid(-2:.2:2);

g = x .\* exp(-x.^2- y.^2); surf(x, y, g)

|  |
| --- |
|  |

print-deps graph.eps

Khi bạn chạy tệp, MATLAB hiển thị bản đồ 3-D sau:

VÍ DỤ KHÔNG TƯƠNG THÍCH

Mặc dù tất cả các chức năng cốt lõi của MATLAB đều có sẵn trong Octave, nhưng có một số chức năng, chẳng hạn như Phép tính ví phân & tích phân, không khớp chính xác trong cả hai ngôn ngữ. Hướng dẫn này đã cố gắng đưa ra cả hai loại ví dụ mà chúng khác nhau về cú pháp.

Xem xét ví dụ sau trong đó MATLAB và Octave sử dụng các hàm khác nhau để tính diện tích đường cong: f(x) = x 2 cos(x) với −4 ≤ x ≤ 9. Sau đây là phiên bản mã MATLAB:

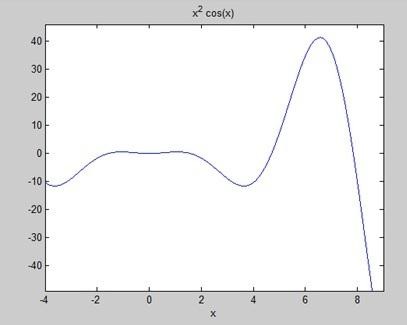
f = x^2\*cos(x); ezplot(f,[-4,9])

a =int(f,-4,9)

disp('Area: '), disp(double(a));

|  |
| --- |
|  |

Khi bạn chạy tệp, MATLAB vẽ đồ thị:



và hiển thị kết quả sau:

|  |
| --- |
|  |

a =

8\*cos(4) + 18\*cos(9) + 14\*sin(4) + 79\*sin(9) Area:

0.3326

Octave , bạn sẽ phải sử dụng gói ký hiệu như sau:

|  |
| --- |
|  |

pkg load symbolic symbols

x = sym("x");

f =inline("x^2\*cos(x)");

ezplot(f,[-4,9]) print-deps graph.eps

[a, ierror, nfneval]= quad(f,-4,9);

display('Area: '), disp(double(a));

## SIMULINK

S i m u l in k tôi là một s i m u la t i o n một n d m od e l -b a s ed d e s i g n ví v iro n m en t cho d y na m i c một n d e m được d ded s y s t e m s , tích hợp với MATLAB. liên kết, Mà còn phát triển qua toán học, Là một dữ liệu lưu lượng đồ họa lập trình công cụ ngôn ngữ để mô hình hóa, mô phỏng và phân tích các hệ thống động đa miền. Về cơ bản, nó là một công cụ lập sơ đồ khối đồ họa với bộ khối có thể tùy chỉnh thư víện.

Nó cho phép bạn kết hợp các thuật toán MATLAB vào các mô hình cũng như xuất kết quả mô phỏng sang MATLAB để phân tích thêm.

Simulink hỗ trợ:

cấp độ hệ thống thiết kế

mô phỏng

mã tự động thế hệ

thử nghiệm và xác minh nhúng hệ thống

Có một số sản phẩm bổ sung khác được cung cấp bởi MathWorks và các sản phẩm phần cứng và phần mềm của bên thứ ba có sẵn để sử dụng với Simulink.

Danh sách sau đây mô tả ngắn gọn về một số trong số họ:

Stateflow cho phép phát triển các máy trạng thái và dòng chảy biểu đồ.

Simulink Coder cho phép tự động tạo mã nguồn C để triển khai theo thời gian thực các hệ thống.

xPC Target cùng với các hệ thống thời gian thực dựa trên x86 cung cấp một môi trường để mô phỏng và kiểm tra các mô hình Simulink và Stateflow trong thời gian thực trên thiết bị vật lý. hệ thống.

Embedded Coder hỗ trợ nhúng cụ thể chỉ tiêu.

HDL Coder cho phép tự động tạo VHDL tổng hợp và Verilog

SimEvents cung cấp một thư víện các khối xây dựng đồ họa để lập mô hình hàng đợi hệ thống

Simulink có khả năng xác minh và xác nhận có hệ thống các mô hình thông qua kiểm tra kiểu mô hình, truy xuất nguồn gốc yêu cầu và phân tích phạm vimô hình.

Simulink Design Verifier cho phép bạn xác định các lỗi thiết kế và tạo các kịch bản trường hợp thử nghiệm để kiểm tra mô hình.

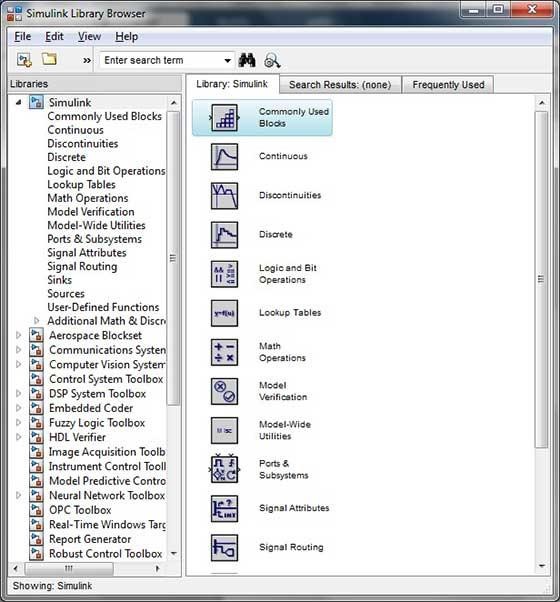
Sử dụng Simulink

Để mở Simulink, gõ vào không gian làm việc MATLAB:

simulink

|  |
| --- |
|  |

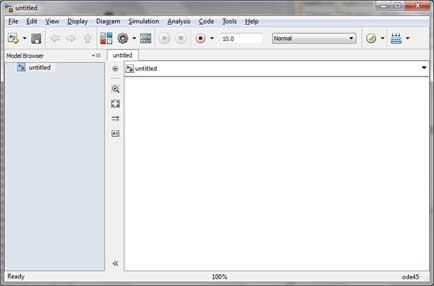
Simulink mở bằng Trình duyệt thư víện . Trình duyệt thư víện được sử dụng để xây dựng các mô hình mô phỏng.



Trên khung cửa sổ bên trái, bạn sẽ tìm thấy một số thư víện được phân loại trên cơ sở của các hệ thống khác nhau, nhấp vào từng thư víện sẽ hiển thị các khối thiết kế trên khung cửa sổ bên phải.

Mô Hình Tòa Nhà

Để tạo một mô hình mới, hãy nhấp vào nút Mới trên thanh công cụ của Trình duyệt thư víện. Thao tác này sẽ mở ra một cửa sổ mô hình mới không có tiêu đề



Một mô hình Simulink là một sơ đồ khối.

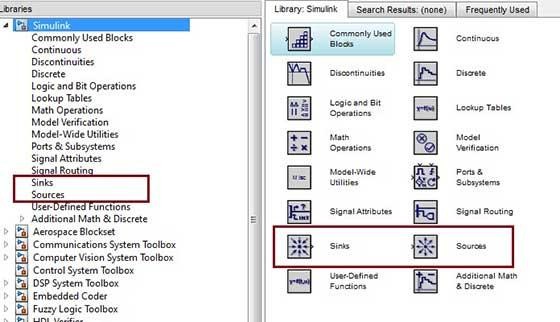
Các phần tử mô hình được thêm vào bằng cách chọn các phần tử thích hợp từ Trình duyệt thư víện và kéo chúng vào Mô hình cửa sổ.

Ngoài ra, bạn có thể sao chép các phần tử mô hình và dán chúng vào cửa sổ mô hình.

ví dụ

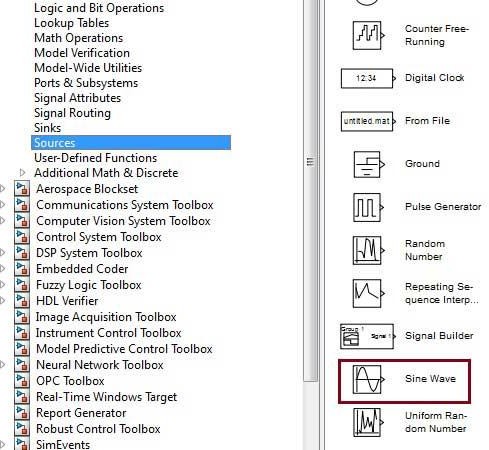
Kéo và thả các mục từ thư víện Simulink để tạo dự án của bạn.

Với mục đích của ví dụ này, 2 khối sẽ được sử dụng cho mô phỏng - Nguồn (tín hiệu) và Sink (phạm ví). Bộ tạo tín hiệu (nguồn) tạo ra tín hiệu tương tự, tín hiệu này sau đó sẽ được hiển thị bằng đồ họa bởi phạm vi(phần chìm).

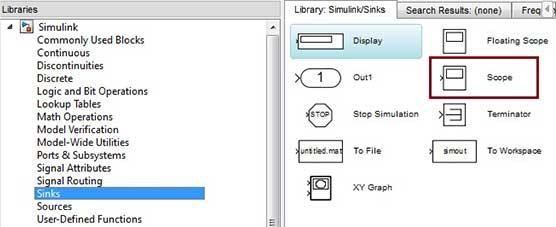


Bắt đầu bằng cách kéo các khối cần thiết từ thư víện vào cửa sổ dự án. Sau đó, kết nối các khối lại với nhau, có thể thực hiện bằng cách kéo các đầu nối từ các điểm kết nối trên một khối này sang các điểm khác.

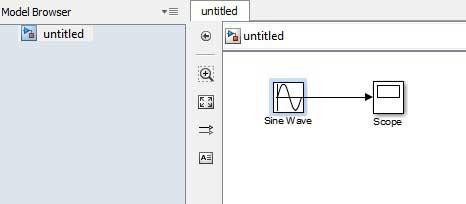
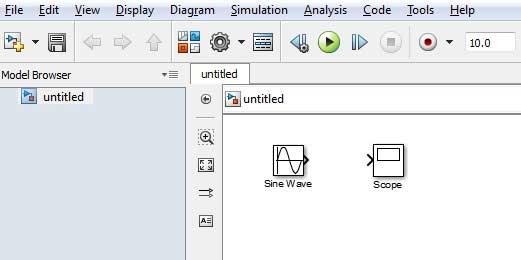
Hãy để chúng tôi kéo một khối 'Sóng hình sin' vào mô hình.



Chọn 'Sinks' từ thư víện và kéo khối 'Scope' vào mô hình.



Kéo một đường tín hiệu từ đầu ra của khối Sine Wave đến đầu vào của khối Scope.



Chạy mô phỏng bằng cách nhấn nút ' Run ', giữ tất cả các thông số mặc định (bạn có thể thay đổi chúng từ menu Mô phỏng)

Bạn sẽ nhận được biểu đồ dưới đây từ phạm ví.

