 [Translated from English to Vietnamese - www.onlinedoctranslator.com](https://www.onlinedoctranslator.com/en/?utm_source=onlinedoctranslator&utm_medium=docx&utm_campaign=attribution)

Hướng dẫn Matlab



### MATLABTUTORIAL

*Đơn giản là học dễ dàng bởi tutorialspoint.com*

hướng dẫn.com

tôi

**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 viên 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 nâng mình lên các cấp độ tiếp theo.

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 đồ họa 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 vi 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

[matlabHướng dẫn2](#_bookmark0)

[khán giả2](#_bookmark1)

[Điều kiện tiên quyết2](#_bookmark2)

[Bản quyền & Tuyên bố miễn trừ trách nhiệmThông báo2](#_bookmark3)

[Tổng quan10](#_bookmark4)

[Sức mạnh của MATLABToán Tin Học10](#_bookmark5)

[tính năng củaMATLAB11](#_bookmark6)

[Sử dụngMATLAB11](#_bookmark7)

[Môi trường12](#_bookmark8)

[Địa phươngCài đặt môi trường12](#_bookmark9)

[hiểu vềMôi trường MATLAB:13](#_bookmark10)

[Thiết lập GNUQuãng tám16](#_bookmark11)

[Cú pháp cơ bản17](#_bookmark12)

[taytrên Thực hành17](#_bookmark13)

[Sử dụng dấu chấm phẩy (;) trongMATLAB18](#_bookmark14)

[ThêmBình luận18](#_bookmark15)

[Toán tử thường được sử dụng vàKý tự đặc biệt18](#_bookmark16)

[Các biến đặc biệt vàhằng số19](#_bookmark17)

[đặt tênBiến19](#_bookmark18)

[tiết kiệm của bạnCông việc20](#_bookmark19)

[Biến21](#_bookmark20)

[NhiềuBài tập22](#_bookmark21)

[tôi đã quênBiến!22](#_bookmark22)

[DàiBài tập23](#_bookmark23)

[định dạngLệnh23](#_bookmark24)

[TạoVectors24](#_bookmark25)

[TạoMa trận25](#_bookmark26)

[Lệnh26](#_bookmark27)

[Các lệnh để quản lý mộtPhiên26](#_bookmark28)

[Các lệnh để làm việc vớiHệ thống26](#_bookmark29)

[Đầu vào vàRa Lệnh27](#_bookmark30)

[Vector, Ma trận và MảngLệnh28](#_bookmark31)

[âm mưuLệnh29](#_bookmark32)

[M-Tệp31](#_bookmark33)

[CácTệp M31](#_bookmark34)

[Tạo và chạyTập lệnh 31](#_bookmark35)

[Ví dụ32](#_bookmark36)

[Dữ liệu -Các loại34](#_bookmark37)

[Các loại dữ liệu có sẵntrong MATLAB34](#_bookmark38)

[Ví dụ35](#_bookmark39)

[Dữ liệuLoại chuyển đổi35](#_bookmark40)

[Xác định dữ liệuCác loại36](#_bookmark41)

[Ví dụ37](#_bookmark42)

[Người vận hành39](#_bookmark43)

[Môn số họcNgười vận hành39](#_bookmark44)

[Ví dụ40](#_bookmark45)

[chức năng choCác phép toán số học41](#_bookmark46)

[quan hệNgười vận hành44](#_bookmark47)

[Ví dụ44](#_bookmark48)

[Ví dụ45](#_bookmark49)

[Hợp lýNhà khai thác45](#_bookmark50)

[chức năng choCác phép toán logic46](#_bookmark51)

[BitwiseHoạt động48](#_bookmark52)

[Ví dụ49](#_bookmark53)

[Bố tríHoạt động50](#_bookmark54)

[Quyết định52](#_bookmark55)

[Ví dụ:54](#_bookmark56)

[Cú pháp54](#_bookmark57)

[Chảysơ đồ55](#_bookmark58)

[Ví dụ:55](#_bookmark59)

[Cú pháp56](#_bookmark60)

[Ví dụ56](#_bookmark61)

[Cú pháp56](#_bookmark62)

[Ví dụ:57](#_bookmark63)

[Cú pháp57](#_bookmark64)

[Ví dụ58](#_bookmark65)

[Cú pháp58](#_bookmark66)

[Ví dụ:59](#_bookmark67)

[vòng lặp60](#_bookmark68)

[Trong khivòng61](#_bookmark69)

[Cú pháp61](#_bookmark70)

[Ví dụ61](#_bookmark71)

[cho vòng lặp61](#_bookmark72)

[Cú pháp62](#_bookmark73)

[Thí dụ162](#_bookmark74)

[Thí dụ262](#_bookmark75)

[Thí dụ363](#_bookmark76)

[lồng vào nhauvòng63](#_bookmark77)

[Cú pháp63](#_bookmark78)

[Ví dụ64](#_bookmark79)

[VòngBáo cáo kiểm soát64](#_bookmark80)

[Chảysơ đồ65](#_bookmark81)

[Ví dụ:65](#_bookmark82)

[Chảysơ đồ66](#_bookmark83)

[Ví dụ:66](#_bookmark84)

[Vectors68](#_bookmark85)

[Chèo thuyềnVectơ:68](#_bookmark86)

[CộtVectơ:68](#_bookmark87)

[Tham khảo các phần tử của mộtVector69](#_bookmark88)

[véc tơHoạt động69](#_bookmark89)

[ma trận74](#_bookmark90)

[Tham khảo các yếu tố củamột ma trận74](#_bookmark91)

[Xóa một hàng hoặc một cột trongmột ma trận76](#_bookmark92)

[Ví dụ76](#_bookmark93)

[ma trậnHoạt động76](#_bookmark94)

[Cộng và trừcủa ma trận77](#_bookmark95)

[Ví dụ77](#_bookmark96)

[Phân chiaMa trận77](#_bookmark97)

[Ví dụ77](#_bookmark98)

[Hoạt động vô hướngcủa Matrix78](#_bookmark99)

[Ví dụ78](#_bookmark100)

[chuyển vị củamột ma trận78](#_bookmark101)

[Ví dụ78](#_bookmark102)

[nốiMa trận79](#_bookmark103)

[Ví dụ79](#_bookmark104)

[ma trậnphép nhân80](#_bookmark105)

[Ví dụ80](#_bookmark106)

[yếu tố quyết định củamột ma trận80](#_bookmark107)

[Ví dụ80](#_bookmark108)

[nghịch đảo của mộtma trận81](#_bookmark109)

[Ví dụ81](#_bookmark110)

[Mảng82](#_bookmark111)

[mảng đặc biệttrong MATLAB82](#_bookmark112)

[MộtHình vuông ma thuật83](#_bookmark113)

[đa chiềuMảng83](#_bookmark114)

[Ví dụ84](#_bookmark115)

[MảngChức năng85](#_bookmark116)

[Ví dụ86](#_bookmark117)

[Sắp xếpMảng87](#_bookmark118)

[Tế bàoMảng87](#_bookmark119)

[ở đâu,88](#_bookmark120)

[Ví dụ88](#_bookmark121)

[Truy cập dữ liệu trong ôMảng88](#_bookmark122)

[Ký hiệu dấu hai chấm90](#_bookmark123)

[Ví dụ91](#_bookmark124)

[Số92](#_bookmark125)

[Chuyển đổi sang dữ liệu số khác nhauCác loại92](#_bookmark126)

[Ví dụ92](#_bookmark127)

[Ví dụ93](#_bookmark128)

[nhỏ nhất vàSố nguyên lớn nhất93](#_bookmark129)

[Ví dụ93](#_bookmark130)

[Điểm nổi nhỏ nhất và lớn nhấtSố94](#_bookmark131)

[Ví dụ94](#_bookmark132)

[dây96](#_bookmark133)

[Ví dụ96](#_bookmark134)

[Ký tự hình chữ nhậtMảng97](#_bookmark135)

[Ví dụ97](#_bookmark136)

[Ví dụ98](#_bookmark137)

[Kết hợp các chuỗi vào một ôMảng98](#_bookmark138)

[Ví dụ98](#_bookmark139)

[Hàm chuỗitrong MATLAB98](#_bookmark140)

[VÍ DỤ100](#_bookmark141)

[ĐỊNH DẠNGSTRINGS100](#_bookmark142)

[THAM GIASTRINGS100](#_bookmark143)

[TÌM VÀTHAY THẾ DÂY CHUYỀN100](#_bookmark144)

[SO SÁNHCHUỖI101](#_bookmark145)

[Chức năng102](#_bookmark146)

[Ví dụ102](#_bookmark147)

[Vô danhChức năng103](#_bookmark148)

[Ví dụ103](#_bookmark149)

[Sơ đẳngvà các chức năng con104](#_bookmark150)

[Ví dụ104](#_bookmark151)

[lồng vào nhauChức năng104](#_bookmark152)

[Ví dụ105](#_bookmark153)

[Chức năng riêng105](#_bookmark154)

[Ví dụ105](#_bookmark155)

[Toàn cầuBiến106](#_bookmark156)

[Ví dụ106](#_bookmark157)

[Dữ liệunhập khẩu107](#_bookmark158)

[Thí dụ1107](#_bookmark159)

[Thí dụ2108](#_bookmark160)

[Thí dụ3109](#_bookmark161)

[toán họcthật đơn giản109](#_bookmark162)

[Cấp thấpTệp I/O109](#_bookmark163)

[Nhập tệp dữ liệu văn bản vớiI/O110 cấp thấp](#_bookmark164)

[Ví dụ110](#_bookmark165)

[Xuất dữ liệu113](#_bookmark166)

[Ví dụ113](#_bookmark167)

[Viết nhật kýTệp114](#_bookmark168)

[Xuất dữ liệu sang tệp dữ liệu văn bản vớiI/O115 cấp thấp](#_bookmark169)

[Ví dụ115](#_bookmark170)

[âm mưu116](#_bookmark171)

[Thêm tiêu đề, nhãn, đường lưới và chia tỷ lệ trênđồ thị118](#_bookmark172)

[Ví dụ118](#_bookmark173)

[Vẽ nhiều chức năng trênCùng đồ thị119](#_bookmark174)

[Ví dụ119](#_bookmark175)

[Đặt màutrên Graph120](#_bookmark176)

[Ví dụ120](#_bookmark177)

[Cài đặtCân Trục121](#_bookmark178)

[Ví dụ121](#_bookmark179)

[đang tạoLô phụ122](#_bookmark180)

[Ví dụ122](#_bookmark181)

[Đồ Họa124](#_bookmark182)

[Đang vẽBiểu đồ thanh124](#_bookmark183)

[Ví dụ124](#_bookmark184)

[Đang vẽđường nét125](#_bookmark185)

[Ví dụ125](#_bookmark186)

[Số baÔ chiều126](#_bookmark187)

[Ví dụ127](#_bookmark188)

[Đại số128](#_bookmark189)

[Giải phương trình đại số cơ bảntrong MATLAB128](#_bookmark190)

[Giải phương trình đại số cơ bảntrong Octave129](#_bookmark191)

[Giải phương trình bậc haitrong MATLAB129](#_bookmark192)

[Giải phương trình bậc hai trongQuãng tám130](#_bookmark193)

[Giải phương trình bậc cao trongMATLAB130](#_bookmark194)

[Giải phương trình bậc cao trongQuãng tám131](#_bookmark195)

[Giải hệ phương trìnhtrong MATLAB131](#_bookmark196)

[Giải hệ phương trìnhtrong Octave132](#_bookmark197)

[Mở rộng và thu thập phương trìnhtrong MATLAB133](#_bookmark198)

[Khai triển và thu thập phương trình trongQuãng tám133](#_bookmark199)

[Thừa số và đơn giản hóa đại sốBiểu thức134](#_bookmark200)

[Ví dụ134](#_bookmark201)

[giải tích135](#_bookmark202)

[tính toánGiới hạn135](#_bookmark203)

[Tính toán giới hạn bằng cách sử dụngQuãng tám136](#_bookmark204)

[Xác minh các thuộc tính cơ bản củaGiới hạn136](#_bookmark205)

[Ví dụ136](#_bookmark206)

[Xác minh các thuộc tính cơ bản của giới hạn bằng cách sử dụngQuãng tám137](#_bookmark207)

[Bên trái và bên phảiGiới hạn hai bên138](#_bookmark208)

[Ví dụ138](#_bookmark209)

[Vi sai140](#_bookmark210)

[Ví dụ140](#_bookmark211)

[Xác minh các quy tắc cơ bảnkhác biệt hóa140](#_bookmark212)

[QUI ĐỊNH1141](#_bookmark213)

[QUI ĐỊNH2141](#_bookmark214)

[QUI ĐỊNH3141](#_bookmark215)

[QUI ĐỊNH4141](#_bookmark216)

[QUI ĐỊNH5141](#_bookmark217)

[QUI ĐỊNH6141](#_bookmark218)

[Ví dụ141](#_bookmark219)

[Đạo hàm của hàm mũ, logarit và lượng giácHàm143](#_bookmark220)

[Ví dụ143](#_bookmark221)

[Máy tính cao hơnLệnh phái sinh145](#_bookmark222)

[Ví dụ145](#_bookmark223)

[Tìm Cực đại và Cực tiểu củamột đường cong146](#_bookmark224)

[Ví dụ146](#_bookmark225)

[giải vi phânphương trình149](#_bookmark226)

[Hội nhập151](#_bookmark227)

[Tìm Tích Phân Bất Định Bằng Cách Sử DụngMATLAB151](#_bookmark228)

[Thí dụ1152](#_bookmark229)

[Thí dụ2152](#_bookmark230)

[Tìm Tích Phân Xác Định Bằng Cách Sử DụngMATLAB154](#_bookmark231)

[Thí dụ1155](#_bookmark232)

[Thí dụ2155](#_bookmark233)

[Đa thức157](#_bookmark234)

[đánh giáĐa thức157](#_bookmark235)

[Tìm về cội nguồnĐa thức158](#_bookmark236)

[đa thứcLắp đường cong158](#_bookmark237)

[Ví dụ158](#_bookmark238)

[biến đổi160](#_bookmark239)

[LaplaceBiến đổi160](#_bookmark240)

[Ví dụ160](#_bookmark241)

[Nghịch đảo LaplaceBiến đổi161](#_bookmark242)

[Ví dụ161](#_bookmark243)

[CácBiến đổi Fourier162](#_bookmark244)

[Ví dụ162](#_bookmark245)

[Nghịch đảo Fourierbiến đổi163](#_bookmark246)

[GNUQuãng tám164](#_bookmark247)

[MATLABso với Octave164](#_bookmark248)

[TƯƠNG THÍCHVÍ DỤ164](#_bookmark249)

[KHÔNG TƯƠNG THÍCHVÍ DỤ165](#_bookmark250)

[simulink167](#_bookmark251)

[sử dụngsimulink168](#_bookmark252)

[Xây dựngNgười mẫu169](#_bookmark253)

[Ví dụ169](#_bookmark254)

CHƯƠNG

1

Tổng quan

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

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 giao diện người dùng; 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 hàm toán học giúp bạn thực hiện các phép tính toán học, tạo biểu đồ 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 và đồ họa 2-D và 3-D
* Đại số tuyến tính
* phương trình đại số
* Chức năng phi tuyến tính
* Số liệu thống kê
* Phân tích dữ liệu
* Giải tích và phương trình vi phân
* phép tính số
* Hội nhập
* biến đổi
* Lắp đường cong
* Nhiều chức năng đặc biệt khác

## 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 phát triển ứng dụng.
* Nó cũng cung cấp một môi trường tương tác để khám phá lặp đi lặp lại, thiết kế và giải quyết vấn đề.
* Nó cung cấp thư viện 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 các phương trình vi phân thông thường.
* Nó cung cấp đồ họa tích hợp để trực quan hóa dữ liệu và các công cụ để tạo các ô tùy chỉnh.
* 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 ứng dụng với giao diện đồ họa tùy chỉnh.
* 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
* Xử lý hình ảnh và video
* Hệ thống điều khiển
* Kiểm tra và Đo lường
* tài chính tính toán
* Sinh học tính toán

CHƯƠNG

2

# Môi trường

## Dùng thử tùy chọn trực tuyến

Ybạnthực sự không cần thiết lập môi trường của riêng bạn để bắt đầu học ngôn ngữ lập trình MATLAB/Octave. Lý do rất đơn giản, chúng tôi đã thiết lập môi trường Octave trực tuyến để bạn có thể thực hiện trực tuyến tất cả các ví dụ có sẵn cùng lúc khi bạn đang thực hiện công việc lý thuyết của mình. Điều này giúp bạn tin tưởng vào những gì bạn đang đọc và 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.

Hãy thử ví dụ sau bằng cách sử dụng tùy chọn Dùng thử có sẵn ở góc trên cùng bên phải của hộp mã mẫu bên dưới:

x=[12345678910];

y1=[.16.08.04.02.013.007.004.002.001.0008];

y2=[.16.07.03.01.008.003.0008.0003.00007.00002];

ký hiệu học(x,y1,'-bo;y1;',x,y2,'-kx;y2;');chức vụ('Tiêu đề cốt truyện');

xnhãn('Trục X');

ylabel('Trục Y');in-biểu đồ deps.tập phim

Đối với hầu hết các ví dụ được đưa ra trong hướng dẫn này, bạn sẽ tìm thấy tùy chọn Dùng thử, vì vậy chỉ cần tận dụng nó và tận hưởng việc học của mình.

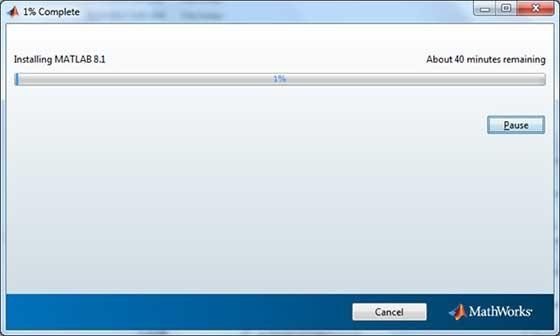
## Cài đặt môi trường cục bộ

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[nơi 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 viên. Bạn cần đăng nhập vào trang web và đợi một chút để họ phê duyệt.

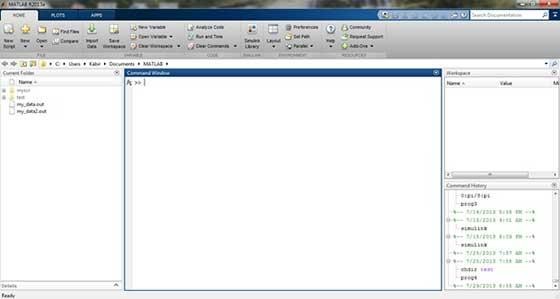
Khi bạn nhận được liên kết tải xuống, như tôi đã nói, chỉ cần vài cú nhấp chuột:





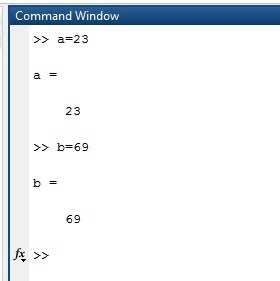
## Hiểu Môi trường MATLAB:

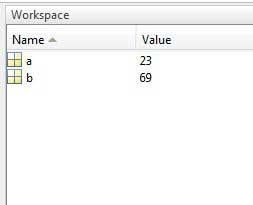
Bạn có thể khởi chạy IDE phát triển MATLAB từ biểu tượng được tạo trên màn hình của bạn. Cửa sổ làm việc chính trong MATLAB được gọi là màn hình nền. Khi bạn khởi động MATLAB, màn hình nền sẽ xuất hiện trong bố cục mặc định của nó:

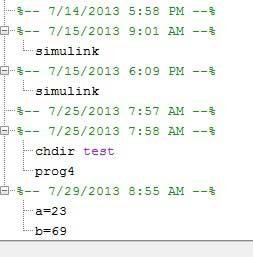


Máy tính để bàn có các bảng sau:

* **Thư mục hiện tại**- Bảng điều khiển này cho phép bạn truy cập các thư mục và tệp dự án của mình.
* **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 tệp.
  + **Lịch sử lệnh**- Bảng này hiển thị hoặc chạy lại các lệnh mà bạn đã nhập tại dòng lệnh.



## 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.

CHƯƠNG

3

Cú pháp cơ bản

mATLABenvsự ủi anhoạt 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

Nhập một biểu thức hợp lệ, ví dụ:

5+5

Và nhấn ENTER

Khi bạn nhấp vào nút Thực thi hoặc gõ Ctrl+E, MATLAB sẽ thực thi ngay lập tức và kết quả trả về là:

ans = 10

Hãy để chúng tôi lấy thêm vài ví dụ:

3^2 %3nâng lên sức mạnh của2

Khi bạn nhấp vào nút Thực thi hoặc gõ Ctrl+E, MATLAB sẽ thực thi ngay lập tức và kết quả trả về là:

ans = 9

Một vi dụ khac,

tội(số Pi/2)% sin của góc90o

Khi bạn nhấp vào nút Thực thi hoặc gõ Ctrl+E, MATLAB sẽ thực thi ngay lập tức và kết quả trả về là:

trả lời = 1

Một vi dụ khac,

7/0 %Chiaquasố không

Khi bạn nhấp vào nút Thực thi hoặc gõ Ctrl+E, MATLAB sẽ thực thi ngay lập tức và kết quả trả về là:

ans = Inf

cảnh báo: chia cho số không

Một vi dụ khac,

732\*20.3

Khi bạn nhấp vào nút Thực thi hoặc gõ Ctrl+E, MATLAB sẽ thực thi ngay lập tức và kết quả trả về là:

trả lời=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, v.v. Nan là viết tắt của 'not a number'.

## Sử dụng dấu chấm phẩy (;) trong MATLAB

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 nhấp vào nút Thực thi hoặc gõ Ctrl+E, MATLAB sẽ thực thi ngay lập tức và kết quả trả về là:

y=8

## Thêm Nhận xét

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

x=9 %gán giá trị9đến x

Bạn cũng có thể viết một khối 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.

## Các toán tử và ký tự đặc biệt thường được sử dụng

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

|  |  |
| --- | --- |
| **Nhà điều hành** | **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. |
| **:** | Đại tràng; 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** |
| **trả lời** | Câu trả lời gần đây nhất. |
| **Số tập** | Độ chính xác của độ chính xác dấu phẩy động. |
| **tôi,j** | Đơn vị ảo √-1. |
| **thông tin** | Vô cực. |
| **NaN** | Kết quả số không xác định (không phải là số). |
| **Số Pi** | 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 lưu 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,

lưu tập tin của tôi

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

tải tập tin của tôi

CHƯƠNG

4

Biến

TôiNMôi trường MATLAB, mỗi biến là một mảng hoặc ma trận. Bạn có thể gán các biến một cách đơn giản. Ví dụ,

x=3 %xác định xvàkhởi tạo nóvớimột 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. Hãy để chúng tôi kiểm tra một ví dụ khác,

x=câu hỏi(16)% xác định xvàkhởi tạo nóvớimột biểu thức

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

x =

4

Xin lưu ý rằng:

* 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 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 ans, có thể được sử dụng sau này.

Ví dụ,

câu hỏi(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/trả lời

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

ans = 1.1182e+03

Hãy xem xét một ví dụ khác:

x=7\*số 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 bài tập trên cùng một dòng. Ví dụ,

một=2;b=7;c=một\*b

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

c = 14

## Tôi đã quên Biến!

Lệnh who hiển thị tất cả các tên biến mà bạn đã sử dụng.

ai

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

biến của bạn là:

aansbcx

y

Lệnh whos hiển thị thêm một chút về các biến:

* Các biến hiện có trong bộ nhớ
* Loại của từng biến
* Bộ nhớ được phân bổ cho từng biến
* Cho dù chúng là các biến phức tạp hay không

ai

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên | Kích thước | byte | Tầng lớp | Thuộc tính |
| một | 1x1 | số 8 | gấp đôi |  |
| trả lời | 1x1 | số 8 | gấp đôi |  |
| b | 1x1 | số 8 | gấp đôi |  |
| c | 1x1 | số 8 | gấp đôi |  |
| x | 1x1 | số 8 | gấp đôi |  |
| y | 1x1 | số 8 | gấp đôi |  |

Lệnh xóa xóa tất cả (hoặc (các) biến đã chỉ định khỏi bộ nhớ.

xóa xthông thoáng

%nó sẽxóa bỏx,thắngkhông hiển thị bất cứ điều gì

% nó sẽ xóa tất cả các biến trong không gian làm việc

% một cách hòa bình và kín đáo

## bài tập dài

Các bài tập dài có thể được kéo dài sang một dòng khác bằng cách sử dụng dấu chấm lửng (...). Ví dụ,

vận tốc ban đầu=0;

sự tăng tốc=9,8;

thời gian=20;

final\_velocity=vận tốc ban đầu...

+sự tăng tốc\*thời gian

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

vận tốc cuối cùng = 196

## Lệnh định dạng

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

Lệnh dài định dạng hiển thị 16 chữ số sau số thập phân. Ví dụ:

định dạngDài

x=7+10/3+5^1.2

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

x = 17,231981640639408

Một vi dụ khac,

định dạngngắn ngủi

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 ngân hàng làm tròn số đến hai chữ số thập phân. Ví dụ,

định dạng ngân hàng daily\_wage=177,45;

daily\_wage=daily\_wage\*6

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

daily\_wage =

1064.70

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

Lệnh định dạng short e cho phép hiển thị ở dạng số mũ với bốn chữ số thập phân cộng với số mũ. Ví dụ,

định dạngngắn ngủie4.678\*4,9

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

ans = 2,2922e+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,

định dạngDàiVí dụ=số Pi

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

x =

3.141592653589793e+00

Lệnh format rat đưa ra biểu thức hữu tỷ gần nhất do một phép tính. Ví dụ,

định dạng chuột4.678\*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**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=[7891011]

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

r =

Cột 1 đến 4

7

Cột 5

11

số 8

9

10

Một vi dụ khac,

r=[7891011];

t=[2,3,4,5,6];

độ phân giải=r+t

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

độ phân giải =

Cột 1 đến 4

9

Cột 5

17

11

13

15

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

c=[7;số 8;9;10;11]

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

c =

7

số 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: chúng ta hãy tạo ma trận 3 nhân 3 như sau:

tôi=[123;456;789]

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

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

CHƯƠNG

5

lệnh

mATLAB tôiĐó 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 quản lýmột phiên

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

|  |  |
| --- | --- |
| **Chỉ huy** | **Mục đích** |
| clc | Xóa cửa sổ lệnh. |
| thông thoáng | Xóa các biến khỏi bộ nhớ. |
| hiện hữu | Kiểm tra sự tồn tại của tệp hoặc biến. |
| toàn cầu | Khai báo các biến là toàn cầu. |
| Cứu giúp | Tìm kiếm một chủ đề trợ giúp. |
| tìm kiếm | Tìm kiếm các mục trợ giúp cho một từ khóa. |
| Từ bỏ | Dừng MATLAB. |
| Ai | Liệt kê các biến hiện tại. |
| Ai | Liệt kê các biến hiện tại (hiển thị dài). |

## Các lệnh làm việc 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, v.v.

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

Cd Ngày Xóa Nhật ký Dir Đường dẫn tải Pwd Lưu Loại gì

wklread

Thay đổi thư mục hiện tại. Hiển thị ngày hiện tại.

Xóa một tập tin.

Bật/tắt ghi tập tin nhật ký. Liệt kê tất cả các tệp trong thư mục hiện tại.

Tải các biến không gian làm việc từ một tệp. Hiển thị đường dẫn tìm kiếm.

Hiển thị thư mục hiện tại.

Lưu các biến không gian làm việc trong một tệp. Hiển thị nội dung của một tập tin.

Liệt kê tất cả các tệp MATLAB trong thư mục hiện tại.

Đọc tệp bảng tính .wk1.

**Mục đích**

**Chỉ huy**

## Lệnh đầu vào và đầu ra

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Chỉ huy** | | **Mục đích** | |
|  | |  | |
| phân tán | | Hiển thị nội dung của một mảng hoặc chuỗi. | |
|  | |  | |
|  | |  | |
| fscanf | | Đọc dữ liệu được định dạng từ một tập tin. | |
|  | |  | |
|  | |  | |
| Sự sắp xếp | | Kiểm soát định dạng hiển thị màn hình. | |
|  | |  | |
|  | |  | |
| fprintf | | Thực hiện ghi định dạng vào màn hình hoặc tập tin. | |
|  | |  | |
|  | |  | |
| Đầu vào | | Hiển thị lời nhắc và chờ đầu vào. | |
|  | |  | |
|  | |  | |
| ; |  |  | 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:

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã định dạng** | | **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:

|  |  |  |  |
| --- | --- | --- | --- |
| **Chức năng định dạng** | | **hiển thị lên đến** | |
|  | |  | |
| định dạng ngắn | | Bốn chữ số thập phân (mặc định). | |
|  | |  | |
|  | |  | |
| định dạng dài | | 16 chữ số thập phân. | |
|  | |  | |
|  | |  | |
| định dạng ngắn e | | Năm chữ số cộng với số mũ. | |
|  | |  | |
|  | |  | |
| định dạng dài e | | 16 chữ số cộng với số mũ. | |
|  | |  | |
|  | |  | |
| định dạng ngân hàng | | Hai chữ số thập phân. | |
|  | |  | |
|  | |  | |
| định dạng + | | Tích cực, tiêu cực, hoặc bằng không. | |
|  | |  | |
|  | |  | |
| định dạng chuột | | xấp xỉ hợp lý. | |
|  | |  | |
|  | |  | |
| định dạng nhỏ gọn | | Loại bỏ một số nguồn cấp dữ liệu dòng. | |
|  | |  | |
|  | |  | |
| định dạng lỏng lẻo |  |  | 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:

|  |  |  |  |
| --- | --- | --- | --- |
| **Chỉ huy** | | **Mục đích** | |
|  | |  | |
| Con mèo | | Nối các mảng. | |
|  | |  | |
|  | |  | |
| Tìm thấy | | Tìm chỉ số của các phần tử khác không. | |
|  | |  | |
|  | |  | |
| Chiều dài | | Tính số phần tử. | |
|  | |  | |
|  | |  | |
| khoảng trống | | Tạo vector cách đều nhau. | |
|  | |  | |
|  | |  | |
| không gian log | | Tạo véc tơ cách đều logarit. | |
|  | |  | |
|  | |  | |
| tối đa | | Trả về phần tử lớn nhất. | |
|  | |  | |
|  | |  | |
| tối thiểu | | Trả về phần tử nhỏ nhất. | |
|  | |  | |
|  | |  | |
| sản xuất | | Sản phẩm của mỗi cột. | |
|  | |  | |
|  | |  | |
| định hình lại | | Thay đổi kích thước. | |
|  | |  | |
|  | |  | |
| Kích thước | | Tính toán kích thước mảng. | |
|  | |  | |
|  | |  | |
| Loại | | Sắp xếp từng cột. | |
|  | |  | |
|  | |  | |
| Tổng |  |  | Tính tổng từng cột. |

|  |  |
| --- | --- |
| Con mắt | Tạo một ma trận nhận dạng. |
| những cái | Tạo một mảng của những người. |
| số không | Tạo một mảng các số không. |
| Vượt qua | Tính toán các sản phẩm chéo của ma trận. |
| chấm | Tính toán các sản phẩm chấm ma trận. |
| khám phá | Tính định thức của một mảng. |
| đầu tư | Tính nghịch đảo của ma trận. |
| pinv | Tính nghịch đảo giả của ma trận. |
| Cấp | Tính hạng của một ma trận. |
| Giới thiệu | Tính toán hình thức bậc thang hàng giảm. |
| Tế bào | Tạo mảng ô. |
| tế bào | Hiển thị mảng ô. |
| cốt truyện di động | Hiển thị biểu diễn đồ họa của mảng ô. |
| số2cell | Chuyển đổi mảng số thành mảng ô. |
| Thỏa thuận | 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 để vẽ đồ thị:

|  |  |
| --- | --- |
| **Chỉ huy** | **Mục đích** |
| trục | Đặt giới hạn trục. |
| fplot | Vẽ đồ thị thông minh của các chức năng. |
| lưới | Hiển thị đường lưới. |
| kịch bản | Tạo biểu đồ xy. |
| in | In đồ thị hoặc lưu đồ thị vào một tập tin. |
| chức vụ | Đặt văn bản ở trên cùng của cốt truyện. |
| xnhãn | Thêm nhãn văn bản vào trục x. |
| ylabel | Thêm nhãn văn bản vào trục y. |

|  |  |
| --- | --- |
| trục | Tạo các đối tượng trục. |
| Thoát | Đóng cốt truyện hiện tại. |
| đóng tất cả | Đóng tất cả các lô. |
| nhân vật | Mở một cửa sổ hình mới. |
| gtext | Cho phép đặt nhãn bằng chuột. |
| giữ | Đóng băng cốt truyện hiện tại. |
| huyền thoại | Vị trí huyền thoại bằng chuột. |
| Làm tươi | Vẽ lại cửa sổ hình hiện tại. |
| bố trí | Chỉ định các thuộc tính của các đối tượng như trục. |
| tình tiết phụ | Tạo các ô trong các cửa sổ con. |
| chữ | Đặt chuỗi trong hình. |
| quán ba | Tạo biểu đồ thanh. |
| loglog | Tạo cốt truyện log-log. |
| vùng cực | Tạo âm mưu cực. |
| Tạo biểu đồ semilog. (logarit trục hoành). |  |
| ký hiệu học | Tạo biểu đồ semilog. (tọa độ logarit). |
| cầu thang | Tạo cốt truyện cầu thang. |
| thân cây | Tạo biểu đồ gốc. |

CHƯƠNG

6

Tệp M

Socho đến nay, chúng tôi đã sử dụng môi trường MATLAB như một máy tính. Tuy nhiên, MATLAB cũng là một ngôn ngữ lập trình mạnh mẽ, 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 các 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:

* **kịch bản**- Các tệp script là các tệp 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.
* **Chức năng**- Các tệp chức năng cũng là các 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 hàm.

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 dòng lệnh.

## Tạo và chạy tệp Script

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 dấu nhắc lệnh
* Sử dụng IDE

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

chỉnh sửaHoặc là

chỉnh sửa<tên tệp>

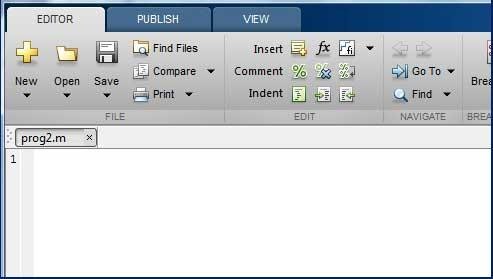
Lệnh trên sẽ tạo tệp trong thư mục MATLAB mặc định. Nếu bạn muốn lưu trữ tất cả các tệp chương trình trong một thư mục cụ thể, thì bạn sẽ phải cung cấp toàn bộ đường dẫn.

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

mkdirchuyên nghiệp %tạo thư mục progs dướivỡ nợthư mục chdirchuyên nghiệp %thay đổi thư mục hiện tại thành progs

chỉnh sửa1 .tôi %tạo một tệp m có tên prog1.tôi

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



Ngoài ra, nếu bạn đang sử dụng IDE, hãy chọn MỚI -> Tập lệnh. 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:

NoOfSinh Viên=6000;Nhân viên giảng dạy=150;Không Giảng DạyNhân Viên=20;Tổng cộng=NoOfSinh Viên+Nhân viên giảng dạy...

+Không Giảng DạyNhân Viên;phân tán(Tổng cộng);

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 của cửa sổ lệnh hiển thị kết quả:

6170

#### Thí dụ

Tạo một tệp tập lệnh và nhập mã sau:

một=5;b=7;

c=một+b

đ=c+tội(b)e=5\*đ

f=kinh nghiệm(-đ)

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

c = 12

d = 12,6570

e = 63,2849

f =

3.1852e-06

CHƯƠNG

7

Dữ liệu- Các loại

mhươu nai ATLABs 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 không gian lưu trữ mới, nếu cần.

Ví dụ,

Tổng cộng=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 nhân 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:

|  |  |
| --- | --- |
| **Loại dữ liệu** | **Sự miêu tả** |
| 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 |

|  |  |
| --- | --- |
| Độc thân | dữ liệu số chính xác duy nhất |
| gấp đôi | dữ liệu số chính xác kép |
| hợp lý | các giá trị logic là 1 hoặc 0, tương ứng là đúng và sai |
| than | dữ liệu ký tự (chuỗi được lưu trữ dưới dạng vectơ ký tự) |
| Mảng ô | mảng các ô được lập chỉ mục, mỗi ô có khả năng lưu trữ một mảng có kích thước và kiểu dữ liệu khác nhau |
| cấu trúc | Các cấu trúc giống như C, mỗi cấu trúc có các trường được đặt tên có khả năng lưu trữ một mảng có kích thước và kiểu dữ liệu khác nhau |
| chức năng xử lý | con trỏ đến một chức năng |
| lớp người dùng | các đối tượng được xây dựng từ một lớp do người dùng định nghĩa |
| các lớp java | các đối tượng được xây dựng từ một lớp Java |

#### Thí dụ

Tạo một tệp script với đoạn mã sau:

str='Chào thế giới!'N=2345

đ=gấp đôi(N)

bỏ=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 =

Chào thế giới! n =

2345

d = 2345

một = 790

rn = 5,6789e+03

c = 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 số** | **Mục đích** |
| than | Chuyển thành mảng ký tự (chuỗi) |
| int2str | Chuyển đổi dữ liệu số nguyên thành chuỗi |

|  |  |
| --- | --- |
| mat2str | Chuyển đổi ma trận thành chuỗi |
| số2str | Chuyển đổi số thành chuỗi |
| str2double | Chuyển đổi chuỗi thành giá trị chính xác kép |
| str2num | Chuyển đổi chuỗi thành số |
| bản địa2unicode | Chuyển đổi byte số thành ký tự Unicode |
| unicode2native | Chuyển đổi các ký tự Unicode thành byte số |
| cơ sở2dec | 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 |
| số2hex | Chuyển đổi đơn và đôi thành chuỗi thập lục phân IEEE |
| ô2mat | Chuyển đổi mảng ô thành mảng số |
| cấu trúc ô2 | Chuyển đổi mảng ô thành mảng cấu trúc |
| tế bàotr | Tạo mảng ô của chuỗi từ mảng ký tự |
| mat2cell | Chuyển đổi mảng thành mảng ô với các ô có kích thước khác nhau |
| số2cell | 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 số** | **Mục đích** |
| Là | phát hiện trạng thái |
| là một | 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 |

|  |  |
| --- | --- |
| sân trường | Xác định xem đầu vào có phải là trường mảng cấu trúc không |
| nổi | Xác định xem đầu vào có phải là mảng dấu phẩy động không |
| ishghandle | Đúng cho Xử lý xử lý đối tượng đồ họa |
| là số nguyên | 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 |
| phi logic | Xác định xem đầu vào có phải là mảng logic không |
| số học | Xác định xem đầu vào có phải là mảng số không |
| đối tượng | Xác định xem đầu vào có phải là đối tượng MATLAB không |
| là thật | Kiểm tra xem đầu vào có phải là mảng thực không |
| vô hướng | Xác định xem đầu vào có vô hướng hay không |
| isstr | Xác định xem đầu vào có phải là mảng ký tự không |
| cấu trúc | Xác định xem đầu vào có phải là mảng cấu trúc không |
| isvector | Xác định xem đầu vào có phải là vectơ không |
| tầng lớp | Xác định lớp đối tượng |
| xác thực thuộc tính | Kiểm tra tính hợp lệ của mảng |
| ai | Liệt kê các biến trong không gian làm việc, với kích thước và loại |

#### Thí dụ

Tạo một tệp script với đoạn mã sau:

x=3là số nguyên(x)nổi(x)isvector(x)vô hướng(x)số học(x)

x=23,54

là số nguyên(x)nổi(x)isvector(x)vô hướng(x)số học(x)

x=[123]

là số nguyên(x)nổi(x)isvector(x)vô hướng(x)

x='Xin chào'là số nguyên(x)nổi(x)

isvector(x)vô hướng(x)số học(x)

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

x =

3

trả lời = 0

trả lời = 1

trả lời = 1

trả lời = 1

trả lời = 1

x = 23,5400

trả lời = 0

trả lời = 1

trả lời = 1

trả lời = 1

trả lời = 1

x =

1

trả lời = 0

trả lời = 1

trả lời = 1

trả lời = 0

x = Xin chào ans =

0

trả lời = 0

trả lời = 1

trả lời = 0

trả lời = 0

2

3

CHƯƠNG

số 8

nhà điều hành

MộtN nhà điều hành tôisa ký hiệu báo cho 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
* Hoạt động theo bit
* Đặt hoạt động

## toán tử số học

MATLAB cho phép hai loại phép toán 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 thao tác trên 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:

|  |  |
| --- | --- |
| **Nhà điều hành** | **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. Thông báo cảnh báo sẽ hiển thị nếu A không đúng tỷ lệ 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. |

#### Thí 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:

một=10;

b=20;

c=một+bđ=một-be=một\*bf=một/bg=một \ bx=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 số** | **Sự miêu tả** |
| cộng(a) | cộng một ngôi; tăng thêm một lượng a |
| cộng (a,b) | Thêm; trả về a + b |
| umus(a) | Phép trừ đơn phương; giảm đi một lượng a |
| trừ (a, b) | Dấu trừ; trả về a - b |
| lần(a, b) | Nhân mảng; trả về a.\*b |
| mtimes(a, b) | Phép nhân ma trận; trả về a\* b |
| chia hết(a, b) | Chia mảng bên phải; trả về a ./ b |
| chia hết(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 |
| sức mạnh(a, b) | Công suất mảng; trả về a.^b |
| sức mạnh (a, b) | Ma trận quyền lực; trả về a^b |
| cuprod(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, mờ) | Trả về tích lũy dọc theo chiều mờ. |
| tích(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, mờ) | trả về tổng tích lũy của các phần tử dọc theo chiều dim. |
| khác biệt(X) | Vi 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,:)] |
| khác biệt(X,n) | Áp dụng đệ quy diff n lần, dẫn đến sự khác biệt thứ n. |
| khác biệt(X,n,mờ) | 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. |
| sản phẩm(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 |
| sản phẩm(A,mờ) | 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. |
| sản xuất ( ,loại dữ liệu) | nhân vào và trả về một mảng trong lớp được chỉ định bởi kiểu dữ liệu. |
| tổng(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 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. |
| tổng(A,mờ) | Tính tổng theo chiều của A được chỉ định bởi scalar dim. |
| tổng (..., 'gấp đôi')  tổng (..., mờ,'gấp đôi') | 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. |
| tổng (..., 'bản địa')  tổng (..., mờ, 'bản địa') | 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. |
| trần(A) | Tròn về hướng dương vô cùng; làm tròn các phần tử của A thành số nguyên gần nhất lớn hơn hoặc bằng A. |
| sửa(A) | Làm tròn về không |
| tầng(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, 'tròn') | Các thương phân số được làm tròn đến số nguyên gần nhất. |
| idivide(A, B, 'tầng') | Các thương phân số được làm tròn về phía âm vô cực đến các số nguyên gần nhất. |
| idivide(A, B, 'trần') | Các thương phân số được làm tròn về phía vô cực đến các số nguyên gần nhất. |
| chế độ (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:

|  |  |
| --- | --- |
| **Nhà điều hành** | **Sự miêu tả** |
| **<** | Ít hơn |
| **<=** | Ít hơn hoặc bằng |
| **>** | Lớn hơn |
| **>=** | Lớn hơn hoặc bằng |
| **==** | Tương đương với |
| **~=** | không bằng |

#### Thí dụ

Tạo một tệp script và nhập mã sau:

một=100;

b=200;

nếu(một>=b)tối đa=mộtkhác

tối đa=bkết thúc

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

tối đa = 200

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

|  |  |
| --- | --- |
| **Hàm số** | **Sự miêu tả** |
| đẳng thức(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 |
| nó (a, b) | Kiểm tra xem a có nhỏ hơn b không |
| không (a, b) | Kiểm tra xem a có bằng b không |
| bình đẳng | Kiểm tra mảng cho bình đẳng |
| bằng nhau | Kiểm tra sự bằng nhau của các mảng, coi các giá trị NaN là bằng nhau |

#### Thí dụ

Tạo một tệp script và nhập mã sau:

%so sánh hai giá trị a=100;

b=200;

nếu(ge(một,b))tối đa=mộtkhác

tối đa=bkết thúc

%so sánh hai giá trị khác nhau a=340;

b=520;

nếu(le(một,b))

phân tán(' a nhỏ hơn hoặc bằng b')khác

phân tán(' a lớn hơn b')kết thúc

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

tối đa = 200

a nhỏ hơn hoặc bằng 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 tương ứng của mảng logic.
  + Đoản mạch - các toán tử này hoạt động trên các biểu thức logic, vô hướng.

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.

#### Thí dụ

Tạo một tệp script và nhập mã sau:

một=5;

b=20;

nếu(một&&b)

phân tán('Dòng 1 - Điều kiện là đúng');

kết thúc

nếu(một||b)

phân tán('Dòng 2 - Điều kiện là đúng');

kết thúc

%hãy thay đổi giá trịofa vàbmột=0;

b=10;

nếu(một&&b)

phân tán('Dòng 3 - Điều kiện là đúng');

khác

phân tán('Dòng 3 - Điều kiện không đúng');

kết thúc

nếu(~(một&&b))

phân tán('Dòng 4 - Điều kiện là đúng');

kết thúc

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

Dòng 1 - Điều kiện đúng Dòng 2 - Điều kiện đúng Dòng 3 - Điều kiện không đúng Dòng 4 - Điều kiện đúng

#### 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 số** | **Sự miêu tả** |
| và (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. |
| không (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. |
| hoặc (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, 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 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. |
| 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) | Các thử nghiệm dọc theo chiều của A được chỉ định bởi độ mờ vô hướng. |
| 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) | Các thử nghiệm dọc theo chiều của A được chỉ định bởi độ mờ vô hướng. |
| 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. |

## Hoạt động theo bit

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 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, v.v.

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

|  |  |
| --- | --- |
| **Hàm số** | **Mục đích** |
| 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, 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 một dịch chuyển sang trái k bit, tương đương với nhân với 2k. 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 |

#### Thí dụ

Tạo một tệp script và nhập mã sau:

một = 60; % 60 = 0011 1100

b = 13; % 13 = 0000 1101

c = bitand(a, b)

% 12 = 0000 1100

c = bitor(a, b)

% 61 = 0011 1101

c = bitxor(a, b)

% 49 = 0011 0001

c = dịch chuyển bit(a, 2)

% 240 = 1111 0000 \*/

c = dịch chuyển bit(a,-2)

% 15 = 0000 1111 \*/

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

c = 12

c = 61

c = 49

c = 240

c = 15

## Đặt hoạt động

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

|  |  |
| --- | --- |
| **Hàm số** | **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 viên(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 (false) nếu 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) bằng nhau. |
| 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 |
| Độc nhất | Các giá trị duy nhất trong mảng |

#### Thí dụ

Tạo một tệp script và nhập mã sau:

một=[723141591282435]

b=[25781416253527]

bạn=liên hiệp(một,b)

tôi=giao nhau(một,b)S=thiết lập(một,b)

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

bạn =

Cột 1 đến 11

2578

9

12

14

15

16

23

24

Cột 12 đến 14

252735

tôi =

781435

s =

912152324

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| một = |  | | | | | | | | |
|  | 7 | 23 | 14 | 15 | 9 | 12 | số 8 | 24 | 35 |
| b = |  |  |  |  |  |  |  |  |  |
|  | 2 | 5 | 7 | số 8 | 14 | 16 | 25 | 35 | 27 |

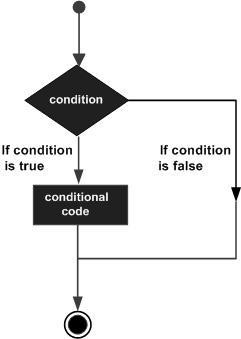
CHƯƠNG

9

quyết định

Đ.eCác cấu trúc tạo câu lệnh yêu cầu lập trình viên chỉ định một hoặc nhiều điều kiện để chương trình đánh giá hoặc kiểm tra, cùng với một hoặc nhiều câu lệnh sẽ được thực thi nếu điều kiện được xác định là đúng và tùy chọn, các câu lệnh khác sẽ được thực thi nếu điều kiện đó được xác định là sai.

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:



MATLAB cung cấp các loại tuyên bố ra quyết định sau đây. Nhấp vào các liên kết sau để kiểm tra chi tiết của họ:

|  |  |
| --- | --- |
| **Tuyên bố** | **Sự miêu tả** |
| [nếu ... câu lệnh kết thúc](http://www.tutorialspoint.com/matlab/if_end_statement_matlab.htm) | Câu lệnh if ... end bao gồm một biểu thức boolean theo sau bởi một hoặc nhiều câu lệnh. |
| [câu lệnh if...else...end](http://www.tutorialspoint.com/matlab/if_else_statement_matlab.htm) | 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. |
| [Câu lệnh if...elseif...elseif...else...end](http://www.tutorialspoint.com/matlab/if_elseif_else_statement.htm) | Một câu lệnh if có thể được theo sau bởi một (hoặc nhiều) câu lệnh khác tùy chọn...và một câu lệnh khác, rất hữu ích để kiểm tra các điều kiện khác nhau. |
| [câu lệnh if lồng nhau](http://www.tutorialspoint.com/matlab/nested_if_statements_matlab.htm) | Bạn có thể sử dụng một câu lệnh if hoặc elseif bên trong (các) câu lệnh if hoặc otherif khác. |
| [tuyên bố chuyển đổi](http://www.tutorialspoint.com/matlab/switch_statement_matlab.htm) | Một câu lệnh chuyển đổi cho phép một biến được kiểm tra sự bằng nhau đối với một danh sách các giá trị. |
| [câu lệnh chuyển đổi lồng nhau](http://www.tutorialspoint.com/matlab/nested_switch_statements_matlab.htm) | Bạn có thể sử dụng một câu lệnh swicth bên trong (các) câu lệnh chuyển mạch khác. |

## nếu … câu lệnh kết thúc

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

Cú pháp của câu lệnh if trong MATLAB là:

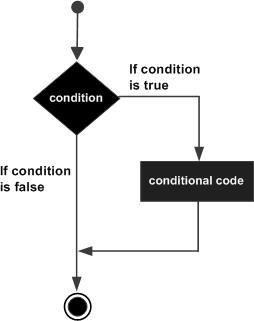
nếu<biểu thức>

%tuyên bố(S)sẽ thực hiệnnếucácbooleansự diễn đạtlà đúng

<tuyên bố>kết thúc

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.

#### Sơ đồ dòng chảy:



Thí dụ:

Tạo một tệp script và nhập mã sau:

một=10;

%kiểm tra tình trạngsử dụng nếutuyên bốnếumột<20

%nếuđiều kiệnlà đúng rồi infprintf sau đây('a nhỏ hơn 20\n');

kết thúc

fprintf('giá trị của a là : %d\n',một);

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

a nhỏ hơn 20 giá trị của a là : 10

## câu lệnh if … other … kết thúc

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:

nếu<biểu thức>

%tuyên bố(S)sẽ thực hiệnnếucácbooleansự diễn đạtlà đúng

<tuyên bố(S)>khác

<tuyên bố(S)>

%tuyên bố(S)sẽ thực hiệnnếucácbooleansự diễn đạtlà sai kết thúc

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.

#### MATLAB if...else statementSơ đồ dòng chảy:

Thí dụ:

Tạo một tệp script và nhập mã sau:

một=100;

%kiểm trabooleanđiều kiệnnếumột<20

%nếuđiều kiệnlà đúng rồi infprintf sau đây('a nhỏ hơn 20\n');

khác

%nếuđiều kiệnlà sai sau đó infprintf sau đây('a không nhỏ hơn 20\n');

kết thúc

fprintf('giá trị của a là : %d\n',một);

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

a không nhỏ hơn 20 giá trị của a là : 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, rất hữu ích để kiểm tra các điều kiện khác nhau.

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

* Một if có thể có số 0 hoặc một số khác và nó phải đứng sau bất kỳ if nào khác.
* Một if có thể có từ 0 đến nhiều otherif và chúng phải xuất hiện trước other.
* Khi một cái khác nếu thành công, không cái nào khác của cái khác hoặc cái khác sẽ được kiểm tra.

#### Cú pháp:

nếu<sự diễn đạt1>

%thực hiệnkhi nàocách diễn đạt1là đúng

<tuyên bố(S)>

khác<sự diễn đạt2>

%thực hiệnkhi nàocácbooleansự diễn đạt2là đúng

<tuyên bố(S)>

khác<sự diễn đạt3>

%thực hiệnkhi nàocácbooleansự diễn đạt3là đúng

<tuyên bố(S)>khác

% thực hiệnkhi nàokhông có điều kiện nào ở trênlà đúng

<tuyên bố(S)>kết thúc

Thí dụ

Tạo một tệp script và nhập mã sau vào đó:

một=100;

%kiểm trabooleanđiều kiệnnếumột==10

%nếuđiều kiệnlà đúng rồi infprintf sau đây('Giá trị của a là 10\n');

khác(một==20)

%nếu khác nếuđiều kiệnlà đúngfprintf('Giá trị của a là 20\n');

khác nếu một==30

%nếu khác nếuđiều kiệnlà đúngfprintf('Giá trị của a là 30\n');

khác

%nếukhông có điều kiệnlà đúng'fprintf('Không cótrong số các giá trị phù hợp\n');

fprintf('Chính xácgiá trị của mộtLà: %d\n', một ); kết thúc

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

Không có giá trị nào khớp với Giá trị chính xác của a là: 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:

Cú pháp của câu lệnh if lồng nhau như sau:

nếu<sự diễn đạt1>

%thực hiệnkhi nàocácbooleansự diễn đạt1là đúng nếu<sự diễn đạt2>

%thực hiệnkhi nàocácbooleansự diễn đạt2là đúng

kết thúc kết thúc

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

#### Thí dụ:

Tạo một tệp script và nhập mã sau vào đó:

một=100;

b=200;

%kiểm trabooleanđiều kiệnnếu(một==100)

%nếuđiều kiệnlà sự thật sau đóKiểm tra lượt theo dõinếu(b==200)

%nếuđiều kiệnlà đúng rồi infprintf sau đây('Giá trị của a là 100 và b là 200\n');

kết thúc

kết thúc

fprintf('Giá trị chính xác của a là : %d\n',một);fprintf('Giá trị chính xác của b là : %d\n',b);

Khi bạn chạy tệp, nó sẽ hiển thị:

Giá trị của a là 100 và b là 200 Giá trị chính xác của a là : 100 Giá trị chính xác của b là : 200

## Tuyên bố chuyển đổi

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 tuyên bố trường hợp.

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 hàm 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 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

Cú pháp của câu lệnh switch trong MATLAB là:

công tắc<switch\_expression>trường hợp<case\_expression>

<tuyên bố>

trường hợp<case\_expression>

<tuyên bố>

...

...

nếu không thì

<tuyên bố>

kết thúc

#### Thí dụ

Tạo một tệp script và nhập mã sau vào đó:

lớp='B';công tắc(lớp)trường hợp'MỘT'

fprintf('Tuyệt vời!\n');trường hợp'B'

fprintf('Làm tốt lắm\n');trường hợp'C'

fprintf('Làm tốt lắm\n');trường hợp'D'

fprintf('Bạn đã qua\n');

trường hợp'F'

fprintf('Thử lại\n');

nếu không thì

fprintf('Điểm không hợp lệ\n');kết thúc

Khi bạn chạy tệp, nó sẽ hiển thị:

Tốt lắm

lớp của bạnlà B

## Câu lệnh chuyển đổi 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:

công tắc(ch1)trường hợp'MỘT'

fprintf('Cái A này là một phần của công tắc bên ngoài');công tắc(ch2)

trường hợp'MỘT'

fprintf('Cái A này là một phần của công tắc bên trong');trường hợp'B'

fprintf('B này là một phần của công tắc bên trong');kết thúc

trường hợp'B'

fprintf('B này là một phần của công tắc bên ngoài');kết thúc

#### Thí dụ:

Tạo một tệp script và nhập mã sau vào đó:

một=100;

b=200;

công tắc(một)

trường hợp100

fprintf('Đây là một phần của công tắc bên ngoài %d\n',một);công tắc(b)

trường hợp200

fprintf('Đây là một phần của công tắc bên trong %d\n',một);

kết thúc

kết thúc

fprintf('Giá trị chính xác của a là : %d\n',một);fprintf('Giá trị chính xác của b là : %d\n',b);

Khi bạn chạy tệp, nó sẽ hiển thị:

Đây là một phần của công tắc bên ngoài 100 Đây là một phần của công tắc bên trong 100 Giá trị chính xác của a là: 100

Giá trị chính xác của b là: 200

# vòng lặp

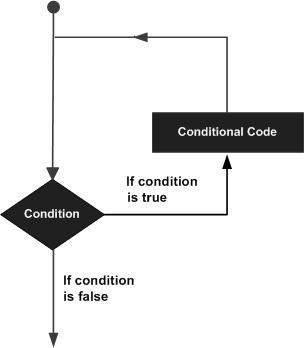
CHƯƠNG

10

tnơi đâycó thể là một 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, v.v.

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

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



MATLAB cung cấp các loại vòng lặp sau để xử lý các yêu cầu lặp. Nhấp vào các liên kết sau để kiểm tra chi tiết của họ:

|  |  |
| --- | --- |
| **Loại vòng lặp** | **Sự miêu tả** |

|  |  |
| --- | --- |
| [trong khi lặp lại](http://www.tutorialspoint.com/matlab/matlab_while_loop.htm) | 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. |
| [cho vòng lặp](http://www.tutorialspoint.com/matlab/matlab_for_loop.htm) | Thực hiện một chuỗi các câu lệnh nhiều lần và viết tắt mã quản lý biến vòng lặp. |
| [vòng lồng nhau](http://www.tutorialspoint.com/matlab/matlab_nested_loops.htm) | Bạn có thể sử dụng một hoặc nhiều vòng lặp bên trong bất kỳ vòng lặp nào khác. |

## Trong khi lặp lại

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

#### Cú pháp:

Cú pháp của vòng lặp while trong MATLAB là:

trong khi<biểu thức>

<tuyên bố>kết thúc

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.

#### Thí dụ

Tạo một tệp script và nhập mã sau:

một=10;

%trong khithực hiện vòng lặptrong khi(một<20)

fprintf('giá trị của a: %d\n',một);một=một+1;

kết thúc

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

giá trị của a: 10 giá trị của a: 11 giá trị của a: 12 giá trị của a: 13 giá trị của a: 14 giá trị của a: 15 giá trị của a: 16 giá trị của a: 17 giá trị của a: 18 giá trị của a: 19

## cho vòng lặp

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:

Cú pháp của vòng lặp for trong MATLAB là:

vìmục lục=giá trị

<báo cáo chương trình>

...

kết thúc

*giá trị*có một trong các dạng sau:

|  |  |
| --- | --- |
| **Sự sắp xếp** | **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* | tạo chỉ mục vectơ cột từ các cột tiếp theo của mảng valArrayon 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 1

Tạo một tệp script và nhập mã sau:

vìmột=10:20

fprintf('giá trị của a: %d\n',một);kết thúc

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

giá trị của a: 10 giá trị của a: 11 giá trị của a: 12 giá trị của a: 13 giá trị của a: 14 giá trị của a: 15 giá trị của a: 16 giá trị của a: 17 giá trị của a: 18 giá trị của a: 19 giá trị của một: 20

#### ví dụ 2

Tạo một tệp script và nhập mã sau:

vìmột=1.0: -0,1:0,0

phân tán(một)kết thúc

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 3

Tạo một tệp script và nhập mã sau:

vìmột= [24,18,17,23,28]

phân tán(một)kết thúc

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:

vìtôi=1:j

vìN=1:k

<tuyên bố>;

kết thúc

kết thúc

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

trong khi<biểu thức1>trong khi<biểu thức2>

<tuyên bố>

kết thúc kết thúc

#### Thí 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:

vìtôi=2:100

vìj=2:100nếu(~chế độ(tôi,j))

nghỉ; %nếuyếu tố được tìm thấy,không phảinguyên tố

kết thúc kết thúc

nếu(j> (tôi/j))

fprintf('%d là số nguyên tố\n',tôi);

kết thúc

kết thúc

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

1. là thủ tướng
2. là thủ tướng

5 là số nguyên tố

7 là số nguyên tố

11 là số nguyên tố

13 là số nguyên tố

17 là số nguyên tố

19 là số nguyên tố

23 là số nguyên tố

29 là số nguyên tố

31 là số nguyên tố

37 là số nguyên tố

41 là số nguyên tố

43 là số nguyên tố

47 là số nguyên tố

53 là số nguyên tố

59 là số nguyên tố

61 là số nguyên tố

67 là số nguyên tố

71 là số nguyên tố

73 là số nguyên tố

79 là số nguyên tố

83 là số nguyên tố

89 là số nguyên tố

97 là số nguyên tố

## 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 vi, tất cả các đối tượng tự động được tạo trong phạm vi đó sẽ bị hủy.

MATLAB hỗ trợ các câu lệnh điều khiển sau. Nhấp vào các liên kết sau để kiểm tra chi tiết của họ.

|  |  |
| --- | --- |
| **Tuyên bố kiểm soát** | **Sự miêu tả** |
| [tuyên bố phá vỡ](http://www.tutorialspoint.com/matlab/matlab_break_statement.htm) | Kết thúc câu lệnh vòng lặp và chuyển thực thi sang câu lệnh ngay sau vòng lặp. |
| [tiếp tục tuyên bố](http://www.tutorialspoint.com/matlab/matlab_continue_statement.htm) | 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. |

## tuyên bố phá vỡ

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.

#### MATLAB break statementSơ đồ dòng chảy:

Thí dụ:

Tạo một tệp script và nhập mã sau:

một=10;

%trong khithực hiện vòng lặptrong khi(một<20)

fprintf('giá trị của a: %d\n',một);một=một+1;

nếu(một>15)

%chấm dứt vòng lặpsử dụng phá vỡtuyên bốnghỉ;

kết thúc

kết thúc

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

giá trị của a: 10 giá trị của a: 11 giá trị của a: 12 giá trị của a: 13 giá trị của a: 14 giá trị của a: 15

## tiếp tục tuyên bố

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, 'tiếp tục' buộc lần lặp tiếp theo của vòng lặp diễn ra, bỏ qua bất kỳ mã nào ở giữa.

#### MATLAB continue statementSơ đồ dòng chảy:

Thí dụ:

Tạo một tệp script và nhập mã sau:

một = 10;

%while thực thi vòng lặp while a < 20

nếu một == 15

% bỏ qua bước lặp a = a + 1; tiếp tục;

kết thúc

fprintf('gia tri cua a: %d\n', a); a = a + 1;

kết thúc

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

giá trị của a: 10 giá trị của a: 11 giá trị của a: 12 giá trị của a: 13 giá trị của a: 14 giá trị của a: 16 giá trị của a: 17 giá trị của a: 18 giá trị của a: 19

# vectơ

CHƯƠNG

11

Mộtvector 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 =

Cột 1 đến 4

7

Cột 5

11

số 8

9

10

## 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 cách các phần tử.

c=[7;số 8;9;10;11]

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

c =

7

số 8

9

10

11

## Tham khảo các yếu tố củamột 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];v(3)

%tạo một vectơ cột của6yếu tố

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

trả lời = 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];v(:)

%tạo một vectơ cột của6yếu tố

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

trả lời = 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 sub\_rv.

rv=[123456789];

sub\_rv=rv(3:7)

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

phụ\_rv =

34

5

6

7

## Phép toán vectơ

Trong phần này, chúng ta hãy thảo luận về các phép toán vectơ sau:

* [Phép cộng và phép trừ vectơ](http://www.tutorialspoint.com/matlab/matlab_vector_add_subtract.htm)
* [Phép nhân vô hướng của vectơ](http://www.tutorialspoint.com/matlab/matlab_vector_scalar_multiplication.htm)
* [Chuyển vị của một vectơ](http://www.tutorialspoint.com/matlab/matlab_vector_transpose.htm)
* [nối các vectơ](http://www.tutorialspoint.com/matlab/matlab_vector_appending.htm)
* [Độ lớn của một vectơ](http://www.tutorialspoint.com/matlab/matlab_vector_magnitude.htm)
* [Vector Chấm Sản Phẩm](http://www.tutorialspoint.com/matlab/matlab_vector_dot_product.htm)
* [Vectơ với các phần tử cách đều nhau](http://www.tutorialspoint.com/matlab/matlab_vector_uniformly_spaced.htm)

## 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.

#### Thí dụ

Tạo một tệp script với đoạn mã sau:

Một= [7,11,15,23,9];

b= [2,5,13,16,20];C=Một+b;

Đ.=Một-b;

phân tán(C);

phân tán(Đ.);

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.

#### Thí dụ

Tạo một tệp script với đoạn mã sau:

v= [12 34 10 8];

tôi=5\*v

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

m =

60170

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 (').

#### Thí dụ

Tạo một tệp script với đoạn mã sau:

r= [1 2 3 4];

tr=r';

v = [1;2;3;4];

truyền hình = v';

phân tán(tr);phân tán(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 trận:

c= [c1,c2]

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

#### Thí 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;số 8];

c= [c1;c2]cMat= [c1,c2]

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

r =

1

rMat =

1

5

c =

1

2

3

4

5

6

7

số 8

2

6

3

7

4

số 8

2

3

4

5

6

7

số 8

cMat =

1

2

3

4

5

6

7

số 8

## Độ lớn của một vectơ

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

|v| = √(v12+ v22+ v32+ … + vn2)

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 véc tơ sv, có các phần tử là bình phương của các phần tử của véc tơ v.

sv = v.\*v;

1. 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= tổng(sv);

1. Sử dụng hàm sqrt để lấy căn bậc hai của tổng cũng là độ lớn của vectơ v. mag = sqrt(s);

#### Thí dụ

Tạo một tệp script với đoạn mã sau:

v= [1:2:20];

sinh viên=v.\*v;% véc tơvớiyếu tố

%nhưhình vuông của vyếu tố của

đp =sum(sv); % tổng bình phương -- tích vô hướng mag =sqrt(dp);% kích cỡ

loại bỏ ('Kích cỡ:'); phân tán (mag);

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

Kích cỡ:

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.

dấu chấm(một,b);

#### Thí dụ

Tạo một tệp script với đoạn mã sau:

v1= [2 3 4];

v2= [1 2 3];

dp=dấu chấm(v1,v2);

phân tán('Chấm sản phẩm:');phân tán(dp);

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

Sản phẩm chấm:

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:tôi]

#### Thí dụ

Tạo một tệp script với đoạn mã sau:

v= [1:2:20];

câu hỏi=v.^2;phân tán(v);phân tán(câu hỏi);

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

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 |
| 1 | 9 | 25 | 49 | 81 | 121 | 169 | 225 | 289 | 361 |

# ma trận

Mộttôiatrix tôisa mảng hai chiều của các số.

CHƯƠNG

12

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 4 nhân 5 a:

một=[12345;23456;34567;45678]

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| một = |  | | | | |
|  | 1 | 2 | 3 | 4 | 5 |
|  | 2 | 3 | 4 | 5 | 6 |
|  | 3 | 4 | 5 | 6 | 7 |
|  | 4 | 5 | 6 | 7 | số 8 |

## Tham khảo các yếu tố củamột ma trận

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

mx(tôi,N);

Ví dụ, để tham chiếu đến phần tử ở hàng thứ 2 và cột thứ 5 của ma trận a như đã tạo ở phần trước, ta gõ:

một=[12345;23456;34567;45678];một(2,5)

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

trả lời = 6

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

Hãy để chúng tôi tạo một vectơ cột v, từ các phần tử của hàng thứ 4 của ma trận a:

một=[12345;23456;34567;45678];

v=một(:,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, chúng tôi viết:

một(:,tôi:N)

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

một=[12345;23456;34567;45678];một(:,2:3)

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

|  |  |
| --- | --- |
| 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.

một=[12345;23456;34567;45678];một(:,2:3)

trả lời =

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

|  |  |
| --- | --- |
| 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:

trả lời =

345

456

Để làm điều này, hãy viết:

một=[12345;23456;34567;45678];sa=một(2:3,2:4)

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

sa =

3

4

5

456

## 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:

một=[12345;23456;34567;45678];một(4,:)=[]

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

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

Tiếp theo, chúng ta hãy xóa cột thứ năm của a:

một=[12345;23456;34567;45678];một(:,5)=[]

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

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

#### Thí dụ

Trong ví dụ này, chúng ta hãy tạo ma trận 3 nhân 3 m, sau đó chúng ta sẽ sao chép hàng thứ hai và thứ ba của ma trận này hai lần để tạo ma trận 4 nhân 3.

Tạo một tệp script với đoạn mã sau:

một=[123;456;789];

new\_mat=một([2,3,2,3], :)

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

|  |  |  |
| --- | --- | --- |
| 4 | 5 | 6 |
| 7 | số 8 | 9 |
| 4 | 5 | 6 |
| 7 | số 8 | 9 |

## Hoạt động ma trận

new\_mat=

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

* [Phép cộng và phép trừ ma trận](http://www.tutorialspoint.com/matlab/matlab_matrix_add_subtract.htm)

c =

d =

* [Phép chia ma trận](http://www.tutorialspoint.com/matlab/matlab_matrix_division.htm)
* [Hoạt động vô hướng của ma trận](http://www.tutorialspoint.com/matlab/matlab_matrix_scalar_operation.htm)
* [Hoán vị của một ma trận](http://www.tutorialspoint.com/matlab/matlab_matrix_transpose.htm)
* [ma trận nối](http://www.tutorialspoint.com/matlab/matlab_matrix_concatenation.htm)
* [Phép nhân ma trận](http://www.tutorialspoint.com/matlab/matlab_matrix_multiplication.htm)
* [Định thức của một ma trận](http://www.tutorialspoint.com/matlab/matlab_matrix_determinant.htm)
* [Nghịch đảo của ma trận](http://www.tutorialspoint.com/matlab/matlab_matrix_inverse.htm)

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.

Thí dụ

Tạo một tệp script với đoạn mã sau:

một= [1 2 3;4 5 6;7 8 9];

b= [7 5 6;2 0 8;5 7 1];

c=một+bd=một-b

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

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

#### Thí dụ

Tạo một tệp script với đoạn mã sau:

một= [1 2 3;4 5 6;7 8 9];

b= [7 5 6;2 0 8;5 7 1];

c=một/bd=một \ b

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

|  |  |  |
| --- | --- | --- |
| -0,52542 | 0,68644 | 0,66102 |
| -0,42373 | 0,94068 | 1.01695 |
| -0,32203 | 1.19492 | 1.37288 |

-3.27778-1.05556 -4.86111

-0.111110.11111 -0,27778

3.055561.27778 4.30556

## 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.

#### Thí dụ

Tạo một tệp script với đoạn mã sau:

một= [10 12 23;14 8 6;27 8 9];

b=2;

c=một+bd=một-thì là ở=một\*bf=một/b

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| c = | 12 | 14 | 25 | |
|  | 16 | 10 | số 8 | |
| d = | 29  số 8 | 10  10 | 11  21 | |
|  | 12 | 6 | 4 | |
| e = | 25  20 | 6  24 | 7  46 | |
|  | 28 | 16 | 12 | |
| f = | 54 | 16 | 18 | |
| 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 (').

#### Thí dụ

Tạo một tệp script với đoạn mã sau:

một= [10 12 23;14 8 6;27 8 9]

b=một'

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

|  |  |  |  |
| --- | --- | --- | --- |
| một = |  | | |
|  | 10 | 12 | 23 |
|  | 14 | số 8 | 6 |
|  | 27 | số 8 | 9 |

|  |  |  |  |
| --- | --- | --- | --- |
| b = |  | | |
|  | 10 | 14 | 27 |
|  | 12 | số 8 | số 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:

* + nối ngang
  + nối dọc

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 nối dọc.

#### Thí dụ

Tạo một tệp script với đoạn mã sau:

một= [10 12 23;14 8 6;27 8 9]

b= [12 31 45;8 0-9;45 2 11]

c= [một,b]

đ= [một;b]

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

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

Mỗi phần tử ở vị trí 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 \*.

#### Thí dụ

Tạo một tệp script với đoạn mã sau:

một= [1 2 3;2 3 4;1 2 5]

b= [2 1 3;5 0-2;2 3-1]

sản xuất=một\*b

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

|  |  |  |  |
| --- | --- | --- | --- |
| một = | 1 | 2 | 3 |
|  | 2 | 3 | 4 |
| b = | 1 | 2 | 5 |
| 2 | | 1 | 3 |
| 5 | | 0 | -2 |
| 2  sản phẩm =  18 | | 3  10 | -1  -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).

#### Thí dụ

Tạo một tệp script với đoạn mã sau:

một= [1 2 3;2 3 4;1 2 5]

thám tử(một)

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

|  |  |  |  |
| --- | --- | --- | --- |
| một = |  | | |
|  | 1 | 2 | 3 |
|  | 2 | 3 | 4 |
|  | 1 | 2 | 5 |

một =

một

trả lời =

-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=Một-1Một=1

Ma trận nghịch đảo không phải lúc nào cũng tồn tại. Nếu định thức của ma trận bằng 0 thì không tồn tại ma trận nghịch đảo 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).

#### Thí dụ

Tạo một tệp script và nhập mã sau:

một= [1 2 3;2 3 4;1 2 5]

inv(một)

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

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | |
| 2 | 3 | 4 | |
| 1 | 2 | 5 | |
| s =  -3.50002.00000.5000 | | | |
| 3.0000 | -1.0000 | | -1.0000 |
| -0,5000 | 0 | | 0,5000 |

# Mảng

CHƯƠNG

13

TôiNMATLAB tất cả các biến thuộc mọi kiểu dữ liệu đều là mảng nhiều chiều. Vectơ là mảng một chiều và ma trận là 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.

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

số không(5)

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

trả lời =

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

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

những cái(4,3)

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

trả lời =

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

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

con mắt(4)

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

trả lời =

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |

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

rand(3,5)

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

trả lời =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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.

ma thuật(4)

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

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

## Mảng đa chiều

trả lời =

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.

một=[795;619;432]

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

|  |  |  |  |
| --- | --- | --- | --- |
| một = |  | | |
|  | 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:

một(:,:,2)=[123;456;789]

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

|  |  |  |
| --- | --- | --- |
| a(:,:,1) = 7 | 9 | 5 |
| 6 | 1 | 9 |
| 4 | 3 | 2 |
| a(:,:,2) = 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | số 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) =  0,0344 | 0,7952 | 0,6463 |
| 0,4387 | 0,1869 | 0,7094 |
| 0,3816 | 0,4898 | 0,7547 |
| 0,7655 | 0,4456 | 0,2760 |
| b(:,:,2) =  0,6797 | 0,4984 | 0,2238 |
| 0,6551 | 0,9597 | 0,7513 |
| 0,1626 | 0,3404 | 0,2551 |
| 0,1190 | 0,5853 | 0,5060 |

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=con mèo(lờ mờ,A1,A2...)

Ở đâu,

* *b*là mảng mới được tạo
* *A1*, A2, ... là các mảng cần nối
* *lờ mờ*là thứ nguyên để nối các mảng

#### Thí dụ

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

một=[987;654;321];

b=[123;456;789];

c=con mèo(3,một,b,[231;478;390])

Khi bạn chạy tệp, nó sẽ hiển thị:

|  |  |  |  |
| --- | --- | --- | --- |
| c(:,:,1) | = |  | |
| 9 |  | số 8 | 7 |
| 6 |  | 5 | 4 |
| 3 |  | 2 | 1 |
| c(:,:,2) | = |  |  |
| 1 |  | 2 | 3 |
| 4 |  | 5 | 6 |
| 7 |  | số 8 | 9 |
| c(:,:,3) | = |  |  |
| 2 |  | 3 | 1 |
| 4 |  | 7 | số 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 số** | **Mục đích** |
| Chiều dài | Độ dài của vectơ hoặc kích thước mảng lớn nhất |
| Ndim | Số kích thước mảng |
| con số | Số phần tử mảng |
| Kích thước | Kích thước mảng |
| iscolumn | Xác định xem đầu vào có phải là vectơ cột không |
| trống rỗng | Xác định xem mảng có trống không |
| ismatrix | Xác định xem đầu vào có phải là ma trận hay không |
| mũi tên | Xác định xem đầu vào có phải là vectơ hàng không |
| vô hướng | 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 |
| vòng quay | Dịch chuyển mảng tròn |
| ctranspose | chuyển vị liên hợp phức tạp |
| chẩn đoán | 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 |
| lộn ngược | Lật ma trận lên xuống |

|  |  |
| --- | --- |
| bất biến | Kích thước hoán vị nghịch đảo của mảng ND |
| hoán vị | Sắp xếp lại kích thước của mảng ND |
| lặp lại | Sao chép và gạch mảng |
| định hình lại | định hình lại mảng |
| thối90 | Xoay ma trận 90 độ |
| shiftdim | Kích thước thay đổi |
| phân loại | Xác định xem các phần tử của tập hợp có được sắp xếp theo thứ tự không |
| loại | Sắp xếp các phần tử mảng tăng dần hoặc giảm dần |
| sắp xếp | Sắp xếp các hàng theo thứ tự tăng dần |
| vắt kiệt | Xóa kích thước đơn lẻ |
| hoán vị | chuyển vị |
| véc tơ hóa | Vector hóa biểu thức |

#### ví dụ

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 đó:

x=[7.1,3.4,7.2,28/4,3.6,17,9.4,8,9];

chiều dài(x)%độ dài của véc tơ x y=rand(3,4,5,2);

dấu vết(y)%khôngkích thướcTrongmảng y

S=['Zara','không','Shamim','Riz','Shadab'];chữ số(S)%khôngcủa các yếu tốTrongS

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

ans = 8

trả lời = 4

ans = 23

Dịch chuyển vòng tròn các phần tử mảng:

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

một=[123;456;789]%mảng ban đầu a

b=vòng quay(một,1)% dịch chuyển vòng giá trị thứ nguyên đầu tiên xuốngqua1.

c=vòng quay(một,[1-1])%các giá trị thứ nguyên đầu tiên của dịch chuyển tròn%xuốngqua1

%vàgiá trị chiều thứ hai ở bên trái%qua1.

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

|  |  |  |  |
| --- | --- | --- | --- |
| một = | 1 | 2 | 3 |
|  | 4 | 5 | 6 |
|  | 7 | số 8 | 9 |
| b = | 7 | số 8 | 9 |
|  | 1 | 2 | 3 |
|  | 4 | 5 | 6 |
| c = | số 8 | 9 | 7 |
|  | 2 | 3 | 1 |
|  | 5 | 6 | 4 |

## Sắp xếp mảng

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

v=[2345129501917]%sắp xếp véc tơ theo chiều ngang(v)%phân loại v

tôi=[264;539;201]% sắp xếp mảng hai chiều(tôi,1)%sắp xếp m dọc theo hàng sắp xếp(tôi,2)%sắp xếp m dọc theo cột

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| v = | 23 | 45 | 12 | 9 | 5 | 0 | 19 | 17 |
| trả lời | = |  |  |  |  |  |  |  |
|  | 0 | 5 | 9 | 12 | 17 | 19 | 23 | 45 |
| m = |  |  |  |  | | | | |
|  | 2 | 6 | 4 |
|  | 5 | 3 | 9 |
|  | 2 | 0 | 1 |
| trả lời | = |  |  |
|  | 2 | 0 | 1 |
|  | 2 | 3 | 4 |
|  | 5 | 6 | 9 |
| trả lời | = |  |  |
|  | 2 | 4 | 6 |
|  | 3 | 5 | 9 |
|  | 0 | 1 | 2 |

## 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 ô được sử dụng để tạo một mảng ô. Cú pháp cho hàm ô là:

C=tế bào(lờ mờ)

C=tế bào(mờ1,...,mờN)Đ.=tế bào(đối tượng)

#### Ở đâu,

* *C*là mảng ô;
* *lờ mờ*là một số nguyên vô hướng hoặc vectơ của các số nguyên xác định kích thước của mảng ô C;
* *mờ1, ... , mờN*là các số nguyên vô hướng xác định kích thước của C;
* *đối tượng*là một trong những điều sau đây:
* Mảng hoặc đối tượng Java
* Mảng .NET kiểu System.String hoặc System.Object

#### Thí dụ

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

c=tế bào(2,5);

c={'Màu đỏ','Màu xanh da trời','Màu xanh lá','Màu vàng','Trắng';12345}

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

c =

'Màu đỏ'

[1]

'Màu xanh da trời'

'Màu xanh lá'

'Màu vàng'

'Trắng'

[

2]

[

3]

[

4]

[

5]

## 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 đơn đầu tiên (), để chỉ các tập hợp ô
  + Đặt các chỉ số trong dấu ngoặc nhọn {}, để chỉ dữ liệu trong các ô riêng lẻ Khi bạn đặt các chỉ số trong dấu ngoặc đơn đầu tiên, nó đề cập đến tập hợp các ô.

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

c={'Màu đỏ','Màu xanh da trời','Màu xanh lá','Màu vàng','Trắng';12345};c(1:2,1:2)

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

ans = 'Đỏ'

[1]

'Màu xanh da trời'

[

2]

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

c={'Màu đỏ','Màu xanh da trời','Màu xanh lá','Màu vàng','Trắng';12345};

c{1,2:4}

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

trả lời =

Màu xanh da trời

trả lời =

Màu xanh lá

ans = Vàng

# Ký hiệu dấu hai chấm

CHƯƠNG

14

thec**ôi thôi(:)**tôilà 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:

trả lời = 1

2

3

4

5

6

7

số 8

9

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:

ans = 100

95

90

85

80

75

70

65

60

55

50

Hãy để chúng tôi lấy một ví dụ khác:

0:số Pi/số 8:số Pi

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

trả lời =

Cột 1 đến 7

00.39270.7854

1.1781

1.5708

1.9635

2.3562

Cột 8 đến 9

2.74893.1416

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):

|  |  |
| --- | --- |
| **Sự sắp xếp** | **Mục đích** |
| **A(:,j)** | là cột thứ j của A. |
| **Một (tôi, :)** | là hàng thứ i của A. |
| **MỘT(:,:)** | 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)** | là kthứ tựtrang 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. |
| **MỘT(:)** | 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. |

Thí dụ

Tạo một tệp script và nhập mã sau vào đó:

Một=[1234;4567;78910]

Một(:,2)%cột thứ hai của A

Một(:,2:3)%thứ haivàcột thứ ba của A

Một(2:3,2:3)%thứ haivàhàng thứ bavàthứ haivàcột thứ ba

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

một =

1

4

7

2

5

số 8

3

6

9

4

7

10

trả lời = 2

5

số 8

trả lời = 2

5

số 8

3

6

9

trả lời = 5

số 8

6

9

# số

CHƯƠNG

15

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

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:

|  |  |
| --- | --- |
| **Hàm số** | **Mục đích** |
| Gấp đôi | Chuyển đổi thành số chính xác gấp đôi |
| Đơn | Chuyển đổi thành số chính xác duy nhất |
| int8 | Chuyển đổi thành số nguyên có dấu 8 bit |
| int16 | Chuyển đổi thành số nguyên có dấu 16 bit |
| int32 | Chuyển đổi thành số nguyên có chữ ký 32 bit |
| int64 | Chuyển đổi thành số nguyên có chữ ký 64 bit |
| uint8 | Chuyển đổi thành số nguyên không dấu 8 bit |
| uint16 | Chuyển đổi thành số nguyên không dấu 16 bit |
| uint32 | Chuyển đổi thành số nguyên không dấu 32 bit |
| uint64 | Chuyển đổi thành số nguyên không dấu 64 bit |

#### Thí dụ

Tạo một tệp script và nhập mã sau:

x=Độc thân([5.323.476.28]).\*7,5

x=gấp đôi([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:

x = 39.9000

x =

26.0250

47.1000

39.9000

x =

26.0250

47.1000

38

x =

23

45

382345

x =

38

23

45

x =

38

23

45

#### Thí dụ

Hãy để chúng tôi mở rộng ví dụ trước hơn một chút. 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=số2cell(x)

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

x =

38

23

45

x =

38

23

45

x =

[38]

[23]

[45]

## 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.

#### Thí 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:

%hiển thị nhỏ nhấtvàlớn nhấtđã kýdữ liệu số nguyên str='Phạm vi cho int8 là:\n\t%d đến %d ';

chạy nước rút(str,intmin('int8'),intmax('int8'))str='Phạm vi cho int16 là:\n\t%d đến %d ';

chạy nước rút(str,intmin('int16'),intmax('int16'))str='Phạm vi cho int32 là:\n\t%d tới %d ';chạy nước rút(str,intmin('int32'),intmax('int32'))str='Phạm vi cho int64 là:\n\t%d tới %d ';chạy nước rút(str,intmin('int64'),intmax('int64'))

%hiển thị nhỏ nhấtvàlớn nhấtchưa kýdữ liệu số nguyên str='Phạm vi cho uint8 là:\n\t%d tới %d ';

chạy nước rút(str,intmin('uint8'),intmax('uint8'))str='Phạm vi cho uint16 là:\n\t%d tới %d ';chạy nước rút(str,intmin('uint16'),intmax('uint16'))str='Phạm vi cho uint32 là:\n\t%d tới %d ';chạy nước rút(str,intmin('uint32'),intmax('uint32'))str='Phạm vi cho uint64 là:\n\t%d tới %d ';chạy nước rút(str,intmin('uint64'),intmax('uint64'))

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

trả lời =

Phạm vi cho int8 là:

-128 đến 127

trả lời =

Phạm vi cho int16 là:

-32768 đến 32767

trả lời =

Phạm vi cho int32 là:

-2147483648 đến 2147483647

trả lời =

Phạm vi cho int64 là:

-9223372036854775808 đến 9223372036854775807

trả lời =

Phạm vi cho uint8 là:

0 đến 255

trả lời =

Phạm vi cho uint16 là:

0 đến 65535

trả lời =

Phạm vi cho uint32 là:

0 đến 4294967295

trả lời =

Phạm vi cho uint64 là:

0 đến 1.844674e+19

## 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.

#### Thí 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:

%hiển thị nhỏ nhấtvàđĩa đơn lớn nhất-độ chính xác

%số điểm nổi

str='Phạm vi cho đơn là:\n\t%g đến %g và\n\t %gđến %g';chạy nước rút(str,-realmax('Độc thân'),-dân tộc thiểu số('Độc thân'),...

dân tộc thiểu số('Độc thân'),realmax('Độc thân'))

%hiển thị nhỏ nhấtvàlớn nhấtgấp đôi-độ chính xác

%số điểm nổi

str='Phạm vi gấp đôi là:\n\t%g đến %g và\n\t %gđến %g';chạy nước rút(str,-realmax('gấp đôi'),-dân tộc thiểu số('gấp đôi'),...

dân tộc thiểu số('gấp đôi'),realmax('gấp đôi'))

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

trả lời =

Phạm vi cho đơn là:

-3.40282e+38 đến -1.17549e-38 và 1.17549e-38to3.40282e+38

trả lời =

Phạm vi cho gấp đôi là:

-1.79769e+308 đến -2.22507e-308 và 2.22507e-308đến1.79769e+308

# Dây

CHƯƠNG

16

CViệc đánh lại 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=Điểm của 'Hướng dẫn'

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

my\_string = Điểm hướng dẫn

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:

ai

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

Đặt tên my\_string

Kích thước 1x16

Lớp byte

32 ký tự

Thuộc tính

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:

Thí dụ

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

my\_string=Điểm của 'Hướng dẫn';

str\_ascii=uint8(my\_string)%số 8-giá trị ascii bit str\_back\_to\_char=than(str\_ascii)

str\_16bit=uint16(my\_string)%16-giá trị ascii bit str\_back\_to\_char=than(str\_16bit)

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

str\_ascii =

Cột 1 đến 14

84117116111114105

Cột 15 đến 16

110116

str\_back\_to\_char = Điểm hướng dẫn str\_16bit =

Cột 1 đến 10

97

108391153280111105

841171161111141059710839115

Cột 11 đến 16

3280111105110116

str\_back\_to\_char = Điểm hướng dẫn

## 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 ký tự.

#### Thí dụ

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

doc\_profile=['Zara Ali' Sr. Bác sĩ phẫu thuật

';...

';...

'Trung tâm Nghiên cứu Tim mạch RN Tagore']doc\_profile=than('Zara Ali','Sr. Bác sĩ phẫu thuật',...'Trung tâm Nghiên cứu Tim mạch RN Tagore')

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

doc\_profile = Zara Ali

bác sĩ phẫu thuật

Trung tâm Nghiên cứu Tim mạch RN Tagore doc\_profile =

Bác sĩ phẫu thuật Zara Ali Sr.

Trung tâm Nghiên cứu Tim mạch RN Tagore

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 tồn mọi khoảng trắng ở cuối trong các mảng đầu vào.
  + Sử dụng hàm nối chuỗi, strcat. Phương pháp này loại bỏ dấu cách trong đầu vào

#### Thí dụ

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

Tên='Zara AliChức vụ='Sr. Bác sĩ phẫu thuật

';

';

làm việc tại='Trung tâm Nghiên cứu Tim mạch RN Tagore';Hồ sơ=[Tên', 'Chức vụ', 'làm việc tại]

Hồ sơ=strcat(Tên,', ',Chức vụ,', ',làm việc tại)

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

hồ sơ = Zara Ali

Hồ sơ Trung tâm Nghiên cứu Tim mạch Tagore =

, bác sĩ phẫu thuật

, RN

Zara Ali,Sr. Bác sĩ phẫu thuật, Trung tâm Nghiên cứu Tim mạch RN Tagore

## Kết hợp các chuỗi thànhmột mảng di động

Từ cuộc thảo luận trước đây của chúng ta, rõ ràng là việc kết hợp các chuỗi có độ dài khác nhau có thể gây khó khăn vì tất cả các chuỗi trong mảng phải có cùng độ dài. Chúng tôi đã sử dụng khoảng trống ở cuối chuỗi để cân bằng độ dài của chúng.

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.

#### Thí dụ

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

Tên='Zara AliChức vụ='Sr. Bác sĩ phẫu thuật

';

';

làm việc tại='Trung tâm Nghiên cứu Tim mạch RN Tagore';Hồ sơ=than(Tên,Chức vụ,làm việc tại);

Hồ sơ=tế bàotr(Hồ sơ);phân tán(Hồ sơ)

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

'Zara Ali' 'Sr. Bác sĩ phẫu thuật'

'Trung tâm Nghiên cứu Tim mạch RN Tagore'

## 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:

|  |  |
| --- | --- |
| **Hàm số** | **Mục đích** |
| **Các hàm lưu trữ văn bản trong mảng ký tự, kết hợp mảng ký tự, v.v.** | |

|  |  |
| --- | --- |
| 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:

Một=số Pi\*1000\*những cái(1,5);

chạy nước rút(' %f \n %.2f \n %+.2f \n %12.2f \n %012.2f \n',Một)

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

ans = 3141,592654

3141.59

+3141.59

3141.59

000003141.59

#### THAM GIA CHUỖI

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

%mảng ô của chuỗi

str\_array={'màu đỏ','màu xanh da trời','màu xanh lá','màu vàng','trái cam'};

%Tham giadâyTrongMảng ôvào trongĐộc thânchuỗistr1=liên kết("-",str\_array)

str2=liên kết(",",str\_array)

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

str1 =

đỏ xanh lục vàng cam str2 =

đỏ, xanh dương, xanh lá cây, vàng, cam

#### TÌM VÀ THAY THẾ DÂY

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

sinh viên={'Zara Ali','Neha Bhatnagar',...'Mica Malik','Madhu Gautam',...

'Madhu Sharma','Bhawna Sharma',...'Nuha Ali','Reva Dutta',...'Sunaina Ali','Sofia Kabir'};

%Cácstrrephàm sốtìm kiếmvàthay thếphụ-chuỗi.sinh viên mới=strrep(sinh viên(số 8),'Tái sinh','Pulomi')

%Trưng bàytên đầu tiên first\_names=sải bước(sinh viên)

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

new\_student=

'Poulomi Dutta' first\_names =

Cột 1 đến 6

'Zara''Neha''Monica'

'Madhu'

'Madhu'

'Bhawna'

Cột 7 đến 10

'Nuha''Reva''Sunaina''Sofia'

#### SO SÁNH CHUỖI

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

str1='Đây là bài kiểm tra'str2='Đây là văn bản'nếu(strcmp(str1,str2))

chạy nước rút('%s và %s bằng nhau',str1,str2)khác

chạy nước rút('%s và %s không bằng nhau',str1,str2)kết thúc

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

str1 =

Đây là thử nghiệm str2 =

Đây là văn bản ans =

Đây là bài kiểm tra và Đây là văn bản không bằng nhau

# Chức năng

CHƯƠNG

17

Mộtniềm vuictionlà một nhóm các câu lệnh cùng thực hiện một nhiệm vụ. Trong MATLAB, các hàm được định nghĩa trong các tệp 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, tách biệt với 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:

hàm số[ra1,ra2,...,raN]=Niềm vui của tôi(trong 1,trong 2,trong 3,...,nhà trọ)

## Thí 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:

hàm sốtối đa=mymax(n1,n2,n3,n4,n5)

%Cái nàyhàm sốtính giá trị lớn nhất của

%năm số đã chonhưđầu vào tối đa= n1;

nếu(n2>tối đa)

tối đa=n2;

kết thúc

nếu(n3>tối đa)tối đa=n3;

kết thúc

nếu(n4>tối đa)

tối đa=n4;

kết thúc

nếu(n5>tối đa)

tối đa=n5;

kết thúc

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:

giúp mymax

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

Hàm này tính toán giá trị lớn nhất trong số năm số được cung cấp làm đầu vào

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:

ans = 89

## 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=@ (người lập luận)sự diễn đạt

#### Thí 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.

Tạo một tệp script và nhập mã sau vào đó:

sức mạnh=@ (x,N)x.^N;kết quả1=sức mạnh(7,3)kết quả2=sức mạnh(49,0,5)kết quả3=sức mạnh(10,-10)kết quả4=sức mạnh(4,5,1,5)

Khi bạn chạy tệp, nó sẽ hiển thị:

kết quả1 = 343

kết quả2 =

7

kết quả3 =

1.0000e-10

kết quả4 =

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.

#### Thí 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 đĩa hàm con tính toán biệt thức.

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

hàm số[x1,x2]=bậc hai(một,b,c)

%Chức năng nàytrả lại gốc rễ của

%một phương trình bậc hai.

%Nónhận3đối số đầu vào

%đó là đồng-hiệu quả của x2,xvàcác

%tình trạng ổn định

%Nótrả về gốc d=đĩa(một,b,c);

x1=(-b+đ)/(2\*một);

x2=(-b-đ)/(2\*một);kết thúc%kết thúccủa bậc hai

hàm sốdis=đĩa(một,b,c)

%hàm sốtính toán phân biệt dis=câu hỏi(b^2-4\*một\*c);

kết thúc%kết thúccủaphụ-hàm số

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

bậc hai(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 vi của một hàm khác và chúng chia sẻ quyền truy cập vào không gian làm việc của hàm chứa.

Hàm lồng nhau tuân theo cú pháp sau:

hàm sốx=Một(p1,p2)

...b(p2)

hàm sốy=b(p3)

...

kết thúc

...

kết thúc

#### Thí 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:

hàm số[x1,x2]=bậc hai2(một,b,c)hàm sốđĩa %lồng vào nhauhàm số

đ=câu hỏi(b^2-4\*một\*c);kết thúc%kết thúccủahàm sốđĩa đĩa;

x1=(-b+đ)/(2\*một);

x2=(-b-đ)/(2\*một);

kết thúc%kết thúccủahàm sốbậc hai2

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

bậc hai2(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 riêng tư. Chúng chỉ hiển thị với các chức năng trong thư mục mẹ.

#### Thí 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:

hàm sốdis=đĩa(một,b,c)

%hàm sốtính toán phân biệt dis=câu hỏi(b^2-4\*một\*c);

kết thúc%kết thúccủaphụ-hàm số

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

hàm số[x1,x2]=bậc hai3(một,b,c)

%Chức năng nàytrả lại gốc rễ của

%một phương trình bậc hai.

%Nónhận3đối số đầu vào

%đó là đồng-hiệu quả của x2,xvàcác

%tình trạng ổn định

%Nótrả về gốc d=đĩa(một,b,c);

x1=(-b+đ)/(2\*một);

x2=(-b-đ)/(2\*một);kết thúc%kết thúccủa bậc hai3

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

bậc hai3(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 cục trong tất cả các hàm.

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.

#### Thí dụ

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

hàm sốtrung bình=Trung bình cộng(con số)toàn cầuTOÀN BỘ

trung bình=Tổng(con số)/TOÀN BỘ;kết thúc

Tạo một tệp script và nhập mã sau vào đó:

toàn cầuTOÀN BỘ;TOÀN BỘ=10;

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

av=Trung bình cộng(N)

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

av = 35.5000

# Nhập dữ liệu

CHƯƠNG

18

Tôinhập khẩudữ liệu trong MATLAB có nghĩa là tải dữ liệu từ tệp bên ngoài. Hàm importdata 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 | **A = 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 | **A = nhập dữ liệu ( , delimiterIn, headerlinesIn)**  Tải dữ liệu từ tệp ASCII, tên tệp hoặc khay nhớ tạm, đọc dữ liệu số bắt đầu từ lineheaderlinesIn+1. |
| 5 | **[A, delimiterOut, headerlinesOut] = importdata( )**  Ngoài ra, trả về ký tự dấu phân cách được phát hiện cho tệp ASCII đầu vào trong delimiterOut và số lượng 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:

tên tập tin='nụ cười.jpg';

Một=nhập dữ liệu(tên tập tin);hình ảnh(Một);

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:

Chủ NhậtThứ HaiNgàyThứ BaNgàyThứ TưNgàyThứ NămNgàyThứ SáuNgàySatureNgày

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 95.0176.2161.54 | 40,57 | 55,79 | 70.28 | 81,53 |
| 73.1145.6579.19 | 93,55 | 75.29 | 69,87 | 74,68 |
| 60.6841.8592.18 | 91,69 | 81,32 | 90,38 | 74,51 |
| 48.6082.1473.82 | 41.03 | 0,99 | 67.22 | 93.18 |
| 89.1344.4757.63 | 89,36 | 13,89 | 19,88 | 46,60 |

Tạo một tệp script và nhập mã sau vào đó:

tên tập tin='dữ liệu hàng tuần.txt';dấu phân cáchTrong='';tiêu đềTrong=1;

Một=nhập dữ liệu(tên tập tin,dấu phân cáchTrong,tiêu đềTrong);

%Khung nhìndữ liệuvìk=[1:7]

phân tán(Một.những người gom góp{1,k})phân tán(Một.dữ liệu(:,k))phân tán('')

kết thúc

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

Chủ nhật

95.0100

73.1100

60.6800

48.6000

89.1300

Thứ hai

76.2100

45.6500

41.8500

82.1400

44.4700

Thứ ba

61.5400

79.1900

92.1800

73.8200

57.6300

Thứ tư

40.5700

93.5500

91.6900

41.0300

89.3600

Thứ năm

55.7900

75.2900

81.3200

0,9900

13.8900

Thứ sáu

70.2800

69.8700

90.3800

67.2200

19.8800

ngày thứ bảy

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:

Một=nhập dữ liệu('-pastespecial')

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

một =

'Toán học là đơn giản'

## 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:

|  |  |
| --- | --- |
| **Hàm số** | **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 tệp văn bản hoặc tệp ASCII.
* 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.
* Hàm fread đọc một luồng dữ liệu ở cấp độ byte hoặc bit.

#### Thí 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:

Dữ liệu lượng mưa

Tháng: Tháng Sáu, Tháng Bảy, Tháng Tám

M=3 12:00:00

Tháng 6 năm 2012

17.2128.5239.7816.55 23,67

19.150.3517.57NaN12.01

17.9228.4917.4017.06 11.09

|  |  |  |  |
| --- | --- | --- | --- |
| 9.599.33 | NaN | 0,31 | 0,23 |
| 10.4613.17 | NaN | 14,89 | 19.33 |
| 20.9719.50 | 17,65 | 14,45 | 14.00 |
| 18.2310.34 | 17,95 | 16,46 | 19.34 |
| 09:10:02 |  |  |  |
| Tháng 7-2012 |  |  |  |
| 12.7616.94 | 14,38 | 11,86 | 16,89 |
| 20.4623.17 | NaN | 24,89 | 19.33 |
| 30.9749.50 | 47,65 | 24,45 | 34.00 |
| 18.2330.34 | 27,95 | 16,46 | 19.34 |
| 30.4633.17 | NaN | 34,89 | 29.33 |
| 30.9749.50 | 47,65 | 24,45 | 34.00 |
| 28.6730.34 | 27,95 | 36,46 | 29,34 |
| 15:03:40 |  |  |  |
| Tháng 8-2012 |  |  |  |
| 17.0916.55 | 19,59 | 17,25 | 19.22 |
| 17.5411.45 | 13,48 | 22,55 | 24.01 |
| NaN21.19 | 25,85 | 25.05 | 27.21 |
| 26.7924.98 | 12.23 | 16,99 | 18,67 |
| 17.5411.45 | 13,48 | 22,55 | 24.01 |
| NaN21.19 | 25,85 | 25.05 | 27.21 |
| 26.7924.98 | 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:

1. Mở tệp bằng chức năng fopen và lấy mã định danh tệp.
2. 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 số dấu phẩy động.
3. Để bỏ qua các ký tự chữ trong tệp, hãy đưa chúng vào phần mô tả định dạng. Để bỏ qua một trường dữ liệu, hãy sử dụng dấu hoa thị ('\*') trong bộ xác 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);

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.
2. chúng tôisẽ 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 - mảng thời gian, tháng và dữ liệu mưa.

Tạo một tệp script và nhập mã sau vào đó:

tên tập tin='/data/myfile.txt';hàng=7;

cols=5;

%Mở tập tin

fid=mở cửa(tên tập tin);

%đọc các tiêu đề tập tin,tìm M(số tháng)

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

%đọc từngbố trísố đovìN=1:m

dữ liệu của tôi(N).thời gian=fscanf(fid,'%S',1);dữ liệu của tôi(N).tháng=fscanf(fid,'%S',1);

%fscanf điền vào mảngTrongthứ tự cột,

%vì vậy hãy chuyển đổi kết quả mydata(N).dữ liệu mưa =...

fscanf(fid,'%f',[hàng,cols]);

kết thúc

vìN=1:m

phân tán(dữ liệu của tôi(N).thời gian),phân tán(dữ liệu của tôi(N).tháng)phân tán(dữ liệu của tôi(N).dữ liệu mưa)

kết thúc

%đóng tệp fclose(fid);

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

12:00:00

Tháng 6 năm 2012

09:10:02

Tháng 7-2012

15:03:40

Tháng 8-2012

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

CHƯƠNG

19

Đ.ata exHải cảngtrong MATLAB có nghĩa là ghi vào tệp. MATLAB cho phép bạn sử dụng dữ liệu của mình trong một ứng dụng khác đọc các tệp 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 đầu ra văn bản kết quả.
  + Tệp ASCII chuyên dụng sử dụng các chức năng cấp thấp như fprintf.
  + MEX để truy cập quy trình C/C++ hoặc Fortran ghi vào một định dạng tệp văn bản cụ thể.

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 vòng loại -ASCII
  + Sử dụng hàm dlmwrite Cú pháp để sử dụng hàm lưu là:

lưu dữ liệu của tôi.ngoàisố\_mảng-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 công cụ xác định.

Cú pháp sử dụng hàm dlmwrite là:

dlmwrite('my\_data.out',số\_mảng,'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.

Thí dụ

Ví dụ sau minh họa khái niệm này. Tạo một tệp script và nhập mã sau:

số\_mảng= [1 2 3 4;4 5 6 7;7 8 9 0];lưu mảng\_data1.ngoàisố\_mảng-ASCII;gõ mảng\_data1.ngoài

dlmwrite('mảng\_data2.out',số\_mảng,'');gõ mảng\_data2.ngoài

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

1.0000000e+002.0000000e+003.0000000e+004.0000000e+00

4.0000000e+005.0000000e+006.0000000e+007.0000000e+00

7.0000000e+008.0000000e+009.0000000e+000.0000000e+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 hàm cell2mat
  + Hoặc xuất mảng ô bằng các hàm I/O tệp cấp thấp.

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='xin chào';

lưu dữ liệu văn bản.ngoàih-dữ liệu văn bản loại ascii.ngoài

MATLAB thực hiện các câu lệnh trên và hiển thị kết quả sau:

1.0400000e+021.0100000e+021.0800000e+021.0800000e+021.1100000e+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:

Nhật ký

Theo tùy chọn, bạn có thể cung cấp tên của tệp nhật ký, chẳng hạn như:

logdata nhật ký.ngoài

Để tắt chức năng nhật ký:

tắt nhật ký

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:

#### Thí dụ

Tạo một tệp script và nhập mã sau vào đó:

%tạo ma trận y,vớihai hàng x=0:10:100;

y= [x;đăng nhập(x)];

%mở tệp tinvìviết

fid=mở cửa('logtable.txt','w');

%Tiêu đề bảngfprintf(fid,'Nhật ký

Chức năng\n\n');

%ingiá trịTrongthứ tự cột

%hai giá trị xuất hiện trên mỗi hàng của tệp fprintf(fid,'%f%f\n' ,y);fđóng(fid);

%hiển thị tệp đã tạo loại logtable.txt

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

Nhật ký

Hàm số

0,000000-Inf

10.0000002.302585

20.0000002.995732

30.0000003.401197

40.0000003.688879

50.0000003.912023

60.0000004.094345

70.0000004.248495

80.0000004.382027

90.0000004.499810

100.0000004.605170

# âm mưu

to kịch bản đồ thị của hàm số, bạn cần thực hiện các bước sau:

CHƯƠNG

20

1. Xác định x, bằng cách chỉ định phạm vi giá trị cho biến x, mà hàm sẽ được vẽ
2. Xác định hàm, y = f(x)
3. Gọi lệnh cốt truyện, như cốt truyện (x, y)

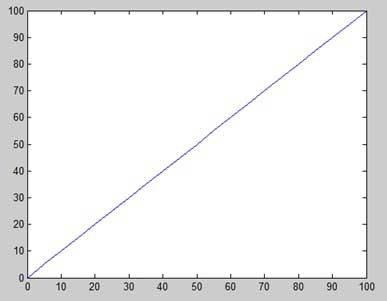
Ví dụ sau đây sẽ chứng minh khái niệm này. Chúng ta hãy vẽ đồ thị hàm đơn giản y = x cho phạm vi giá trị của x từ 0 đến 100, với số gia là 5.

Tạo một tệp script và nhập mã sau:

x=[0:5:100];

y=x;kịch bản(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 = x2. 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.

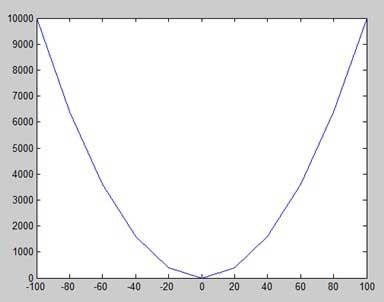
Tạo một tệp script và nhập mã sau:

x=[12345678910];

x=[-100:20:100];

y=x.^2;kịch bản(x,y)

Khi bạn chạy tệp, MATLAB sẽ hiển thị biểu đồ sau:

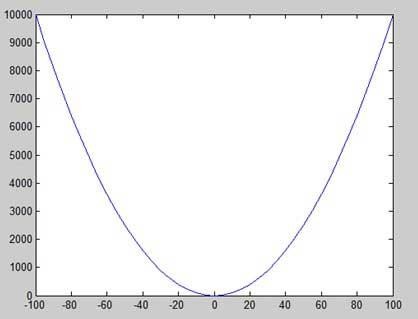


Thay đổi tệp mã một chút, giảm số gia xuống 5:

x=[-100:5:100];

y=x.^2;kịch bản(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.

* Lệnh xlabel và ylabel tạo nhãn dọc theo trục x và trục y.
* Lệnh title cho phép bạn đặt tiêu đề trên biểu đồ.
* Lệnh grid on cho phép bạn đặt các đường lưới trên biểu đồ.
* Lệnh 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 ô vuông.

#### Thí dụ

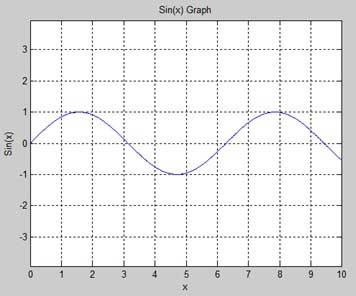
Tạo một tệp script và nhập mã sau:

x=[0:0,01:10];

y=tội(x);

kịch bản(x,y),xnhãn('x'),ylabel('Tội lỗi(x)'),chức vụ('Đồ thị Sin(x)'),lưới điện trên,trục bằng

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:

#### Thí dụ

Tạo một tệp script và nhập mã sau:

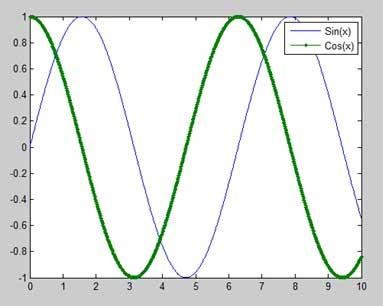
x=[0:0,01:10];

y=tội(x);

g=cos(x);

kịch bản(x,y,x,g,'.-'),huyền thoại('Tội lỗi(x)','Co(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** |
| Màu đen | **k** |
| Màu xanh da trời | **b** |
| Màu đỏ | **r** |
| lục lam | **c** |
| Màu xanh lá | **g** |
| đỏ tươi | **tôi** |
| Màu vàng | **y** |

#### Thí dụ

Hãy vẽ đồ thị của haiđa thức 1.f(x) = 3x 4+ 2x3+ 7x2 + 2x+9 và

2.g(x) = 5x 3+ 9x + 2

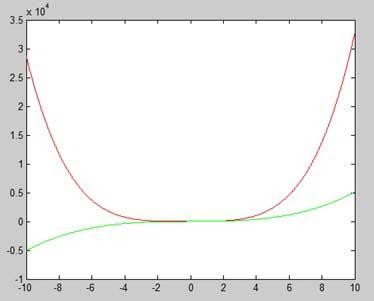
Tạo một tệp script và nhập mã sau:

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;

kịch bản(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:

trục([xmin xmax ymin ymax])

Ví dụ sau đây cho thấy điều này:

#### Thí dụ

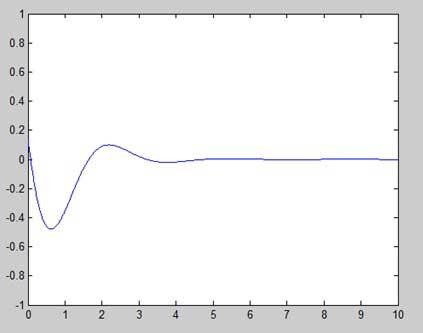
Tạo một tệp script và nhập mã sau:

x=[0:0,01:10];

y=kinh nghiệm(-x).\*tội(2\*x+3);

kịch bản(x,y),trục([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 các ô con.

Cú pháp của lệnh là:

tình tiết phụ(tôi,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 thể hiện khái niệm:

#### Thí dụ

Hãy để chúng tôi tạo ra hai lô:

y = e−1,5 lầnsin(10x) y = e−2xtội lỗi(10x)

Tạo một tệp script và nhập mã sau:

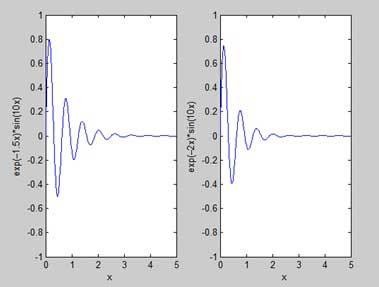
x=[0:0,01:5];

y=kinh nghiệm(-1,5\*x).\*tội(10\*x);tình tiết phụ(1,2,1)

kịch bản(x,y),xnhãn('x'),ylabel('exp(–1.5x)\*sin(10x)'),trục([05-11])y=kinh nghiệm(-2\*x).\*tội(10\*x);

tình tiết phụ(1,2,2)kịch bản(x,y),xnhãn('x'),ylabel('exp(–2x)\*sin(10x)'),trục([05-11])

Khi bạn chạy tệp, MATLAB tạo biểu đồ sau:



# đồ họa

CHƯƠNG

21

tChàoChương này sẽ tiếp tục khám phá khả năng vẽ đồ thị và vẽ đồ thị của MATLAB. Chúng ta sẽ thảo luận:

* Vẽ biểu đồ thanh
* Vẽ đường viền
* Đồ thị ba chiều

## 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.

#### Thí dụ

Hãy để chúng tôi có một lớp học tưởng tượng với 10 sinh viên. Chúng ta biết phần trăm số điểm mà những sinh viên 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:

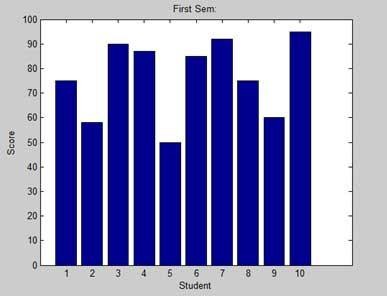
x=[1:10];

y=[75,58,90,87,50,85,92,75,60,95];

quán ba(x,y),xnhãn('Sinh viên'),ylabel('Điểm'),chức vụ('Học kỳ đầu tiên:')

in-biểu đồ deps.tập phim

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.

#### Thí 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 gồm các phần tử cung cấp phạm vi trên x và y cùng với thông số kỹ thuật về số gia trong từng 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:

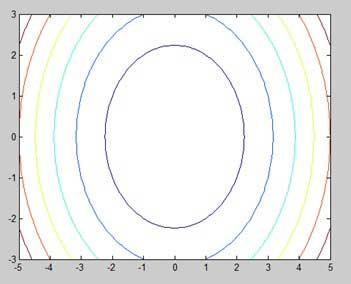
[x,y]=lưới điện(–5:0,1:5,–3:0,1:3);

Cuối cùng, chúng ta cần gán chức năng. Giả sử hàm của chúng ta là: x2 + y2 Tạo một tệp script và nhập đoạn mã sau:

[x,y]=lưới điện(-5:0,1:5,-3:0,1:3);%biến độc lập g=x.^2+y.^2;%chức năng của chúng tôi

viền(x,y,g)%gọi đường viềnchức năng in-biểu đồ deps.tập phim

Khi bạn chạy tệp, MATLAB hiển thị bản đồ đường viền sau:

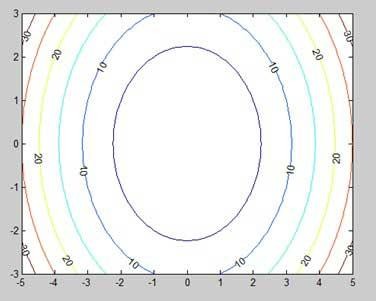


Hãy để chúng tôi sửa đổi mã một chút để tạo bản đồ:

[x,y]=lưới điện(-5:0,1:5,-3:0,1:3);%biến độc lập g=x.^2+y.^2;%chức năng của chúng tôi

[C,h]=viền(x,y,g);%gọi đường viềnbộ chức năng(h,'Hiển thị văn bản','trên','Bước văn bản',được(h,'Bước cấp')\*2)in-biểu đồ deps.tập phim

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 themeshgrid. 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:

#### Thí dụ

Hãy để chúng tôi tạo bản đồ bề mặt 3D cho hàm g = xe-(x2 + y2)

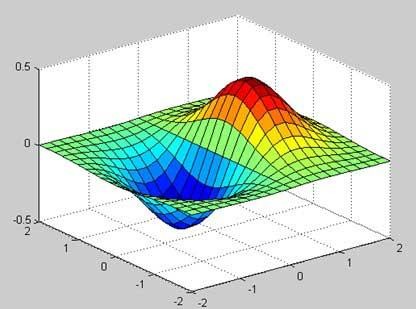
Tạo một tệp script và nhập mã sau:

[x,y]=lưới điện(-2:.2:2);

g=x.\*kinh nghiệm(-x.^2-y.^2);lướt sóng(x,y,g)

in-biểu đồ deps.tập phim

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

CHƯƠNG

22

Socho đến nay, chúng ta đã thấy rằng tất cả các ví dụ đều hoạt động trong MATLAB cũng như GNU của nó, còn được gọi là Octave. 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 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

giải quyết('x-5=0')

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

trả lời = 5

Bạn cũng có thể gọi hàm giải quyết là:

y=giải quyết('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:

giải quyết('x-5')

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

trả lời = 5

Nếu phương trình bao gồm nhiều ký hiệu, thì MATLAB theo mặc định giả định rằng bạn đang giải tìm x, tuy nhiên, lệnh giải có dạng khác:

giải quyết(phương trình,Biến đổi)

ở đâu, bạn cũng có thể đề cập đến biến.

Ví dụ, chúng ta hãy giải phương trình v – u – 3t2 = 0, cho v. Trong trường hợp này, chúng ta nên viết:

giải quyết('vu-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 Octave

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

rễ([1,-5])

Octave sẽ thực thi câu lệnh trên và trả về kết quả như sau:

trả lời = 5

Bạn cũng có thể gọi hàm giải quyết là:

y=rễ([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 x2 -7x +12 = 0. Tạo một tệp tập lệnh và nhập mã sau:

tương đương='x^2 -7\*x + 12 = 0';S=giải quyết(tương đương);

phân tán('Gốc đầu tiên là:'),phân tán(S(1));

phân tán('Gốc thứ hai là:'),phân tán(S(2));

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

Gốc đầu tiên là:

3

Gốc thứ hai là:

4

## Giải phương trình bậc hai trong quãng tám

Ví dụ sau đây giải phương trình bậc hai x2 -7x +12 = 0 trong Octave. Tạo một tệp script và nhập mã sau:

S=rễ([1,-7,12]);

phân tán('Gốc đầu tiên là:'),phân tán(S(1));

phân tán('Gốc thứ hai là:'),phân tán(S(2));

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

Gốc đầu tiên là:

4

Gốc thứ hai là:

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

giải quyết('(x-3)^2\*(x-7)=0')

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

trả lời = 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 x4−7x3 + 3x2−5x + 9

= 0.

Tạo một tệp script và nhập mã sau:

tương đương='x^4 - 7\*x^3 + 3\*x^2 - 5\*x + 9 = 0';S=giải quyết(tương đương);

phân tán('Gốc đầu tiên là:'),phân tán(S(1));

phân tán('Gốc thứ hai là:'),phân tán(S(2));

phân tán('Gốc thứ ba là:'),phân tán(S(3));

phân tán('Gốc thứ tư là:'),phân tán(S(4));

%chuyển đổi rễ thànhgấp đôiloại

phân tán('Giá trị số của gốc đầu tiên'),phân tán(gấp đôi(S(1)));phân tán('Giá trị số của căn thứ hai'),phân tán(gấp đôi(S(2)));phân tán('Giá trị số của căn thứ ba'),phân tán(gấp đôi(S(3)));phân tán('Giá trị số của căn thứ tư'),phân tán(gấp đôi(S(4)));

Khi bạn chạy tệp, nó sẽ trả về kết quả như sau:

Gốc đầu tiên là: 6.630396332390718431485053218985

Gốc thứ hai là:

1.0597804633025896291682772499885

Gốc thứ ba là:

- 0,34508839784665403032666523448675 - 1,0778362954630176596831109269793\*i

Gốc thứ tư là:

- 0,34508839784665403032666523448675 + 1,0778362954630176596831109269793\*i

Giá trị số của gốc đầu tiên 6.6304

Giá trị số của căn thứ hai 1,0598

Giá trị số của căn bậc ba

-0,3451 - 1,0778i

Giá trị số của căn bậc tư

-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 giải phương trình bậc 4 x4 − 7x3 + 3x2 − 5x + 9 = 0. Tạo một tệp tập lệnh và nhập mã sau:

v=[1,-7,3,-5,9];

S=rễ(v);

%chuyển đổi rễ thànhgấp đôiloại

phân tán('Giá trị số của gốc đầu tiên'),phân tán(gấp đôi(S(1)));phân tán('Giá trị số của căn thứ hai'),phân tán(gấp đôi(S(2)));phân tán('Giá trị số của căn thứ ba'),phân tán(gấp đôi(S(3)));phân tán('Giá trị số của căn thứ tư'),phân tán(gấp đôi(S(4)));

Khi bạn chạy tệp, nó sẽ trả về kết quả như sau:

Giá trị số của gốc đầu tiên 6.6304

Giá trị số của căn thứ hai

-0,34509 + 1,07784i

Giá trị số của căn bậc ba

-0,34509 - 1,07784i

Giá trị số của căn bậc tư 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=giải quyết('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

trả lời =

-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 Octave

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:

Một=[5,9;3,-6];

b=[5;4];

một \ b

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

trả lời =

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

## Mở rộng 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:

biểu tượng x%biến tượng trưng x sym y%biến tượng trưng x

%mở rộng phương trình mở rộng((x-5)\*(x+9))

mở rộng((x+2)\*(x-3)\*(x-5)\*(x+7))mở rộng(tội(2\*x))mở rộng(cos(x+y))

%thu thập phương trình thu thập(x^3\*(x-7))sưu tầm(x^4\*(x-3)\*(x-5))

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

trả lời =

x^2 + 4\*x - 45 ans =

x^4 + x^3 - 43\*x^2 + 23\*x + 210 ans =

2\*cos(x)\*sin(x)trả lời =

cos(x)\*cos(y) - sin(x)\*sin(y) ans =

x^4 - 7\*x^3 ans =

x^6 - 8\*x^5 + 15\*x^4

## Mở rộng và thu thập các phương trình trong Octave

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:

%trước hết tảibưu kiện,đảm bảo nó đã được cài đặt.pkg tải tượng trưng

%làm biểu tượngmô-đunký hiệu có sẵn

%xác định các biến tượng trưng x=ký hiệu('x');

y=ký hiệu('y');

z=ký hiệu('z');

%mở rộng phương trình mở rộng((x-5)\*(x+9))

mở rộng((x+2)\*(x-3)\*(x-5)\*(x+7))mở rộng(Tội(2\*x))mở rộng(cos(x+y))

%thu thập phương trình thu thập(x^3\*(x-7),z)sưu tầm(x^4\*(x-3)\*(x-5),z)

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

trả lời =

-45.0+x^2+(4.0)\*x

trả lời =

210.0+x^4-(43.0)\*x^2+x^3+(23.0)\*x

trả lời =

tội lỗi((2.0)\*x)trả lời =

cos(y+x)trả lời =

x^(3.0)\*(-7.0+x)

trả lời =

(-3.0+x)\*x^(4.0)\*(-5.0+x)

## Thừa số và đơn giản hóa các biểu thức đại số

Lệnh thừa số phân tích biểu thức và lệnh đơn giản hóa đơn giản hóa biểu thức. Ví dụ sau minh họa khái niệm:

#### Thí dụ

Tạo một tệp script và nhập mã sau:

sym x sym y

hệ số(x^3-y^3)hệ số([x^2-y^2,x^3+y^3])đơn giản hóa((x^4-16)/(x^2-4))

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

trả lời =

(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

CHƯƠNG

23

mATLAB chuyên nghiệpcung cấp nhiều cách khác nhau để giải các bài toán về vi phân và tích phân, giải các phương trình vi phân ở bất kỳ mức độ nào và 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 Vi 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 vi 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) = (x3 + 5)/(x4 + 7), vì x có xu hướng bằng không.

sym x giới hạn((x^3+5)/(x^4+7))

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

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. Cái đầu tiên là biểu thức và cái thứ hai là số mà x tiến đến, đây là a.

Ví dụ: chúng ta hãy tính giới hạn của hàm f(x) = (x-3)/(x-1), vì x có xu hướng tiến tới 1.

giới hạn((x-3)/(x-1),1)

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

trả lời =

NaN

Hãy lấy một ví dụ khác,

giới hạn(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 tải biểu tượng tượng trưng x=ký hiệu("x");

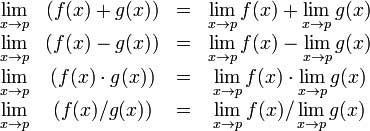
dự bị((x^3+5)/(x^4+7),x,0)

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 giới hạn

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.

#### Thí dụ

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

biểu tượng x

f=(3\*x+5)/(x-3);g=x^2+1;

l1=giới hạn(f,4)l2=giới hạn(g,4)

lThêm=giới hạn(f+g,4)lSub=giới hạn(f-g,4)lĐa=giới hạn(f\*g,4)lDiv=giới hạn(f/g,4)

Khi bạn chạy tệp, nó sẽ hiển thị:

l1 = 17

l2 = 17

lThêm = 34

lSub = 0

lĐa = 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:

biểu tượng ký hiệu tải pkg

x=ký hiệu("x");

f=(3\*x+5)/(x-3);g=x^2+1;

l1=dự bị(f,x,4)

l2=dự bị(g,x,4)lThêm=dự bị(f+g,x,4)lSub=dự bị(f-g,x,4)

lĐa=dự bị(f\*g,x,4)lDiv=dự bị(f/g,x,4)

Octave sẽ thực thi câu lệnh trên và trả về kết quả như sau:

l1 =

17,0

l2 =

17,0

lThêm =

34,0

lSub =

0,0

lĐa =

289.0

lDiv =

1.0

#### Giới hạn bên trái và bên phải

Khi một hàm có sự gián đoạn đối với một số giá trị cụ thể của biến, giới hạn không tồn tại tại điểm đó. Nói cách khác, giới hạn của hàm số f(x) gián đoạn tại x = a, khi giá trị của giới hạn khi x tiến đến x từ vế trái không bằng giá trị của giới hạn khi x tiến dần từ vế phải.

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|

Chúng tôi sẽ chỉ ra rằng limx->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 không liên tục
* Bằng cách tính toán các giới hạn và chỉ ra rằng cả hai đều khác nhau.

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.

#### Thí dụ

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

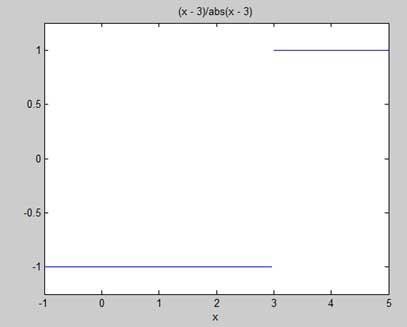
f=(x-3)/cơ bụng(x-3);

ezplot(f,[-1,5])

tôi=giới hạn(f,x,3,'bên trái')

r=giới hạn(f,x,3,'đúng')

Khi bạn chạy tệp, MATLAB sẽ vẽ biểu đồ sau,



và hiển thị đầu ra sau:

tôi =

-1

r = 1

# sự khác biệt

CHƯƠNG

24

mATLAB chuyên nghiệpvides lệnh diff để tính toán các dẫn xuất tượng trưng. Ở dạng đơn giản nhất, bạn chuyển hàm mà bạn muốn phân biệt thành lệnh khác làm đối số.

Ví dụ: chúng ta hãy tính đạo hàm của hàm f(t) = 3t2 + 2t-2

Thí dụ

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

biểu tượng t

f=3\*t^2+2\*t^(-2);khác biệt(f)

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

trả lời =

6\*t - 4/t^3

Sau đây là Octave tương đương với phép tính trên:

biểu tượng ký hiệu tải pkg

t=ký hiệu("t");

f=3\*t^2+2\*t^(-2);phân biệt(f,t)

Octave thực thi mã và trả về kết quả sau:

trả lời =

-(4.0)\*t^(-3.0)+(6.0)\*t

## Xác minh các quy tắc khác biệt cơ bản

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)/g2

QUY TẮC 5

Quy tắc lũy thừa đa thức hoặc cơ bản 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)

## Thí dụ

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

sym x sym t

f=(x+2)\*(x^2+3)der1=khác biệt(f)

f=(t^2+3)\*(câu hỏi(t)+t^3)der2=khác biệt(f)

f=(x^2-2\*x+1)\*(3\*x^3-5\*x^2+2)der3=khác biệt(f)

f=(2\*x^2+3\*x)/(x^3+1)thứ 4=khác biệt(f)

f=(x^2+1)^17thứ 5=khác biệt(f)

f=(t^3+3\*t^2+5\*t-9)^(-6)thứ 6=khác biệt(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)

chết3 =

(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)

chết4 =

(4\*x + 3)/(x^3 + 1) - (3\*x^2\*(2\*x^2 + 3\*x))/(x^3 + 1)^2

f =

(x^2 + 1)^17

chết5 =

34\*x\*(x^2 + 1)^16

f =

1/(t^3 + 3\*t^2 + 5\*t - 9)^6

chết6 =

-(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 tải biểu tượng tượng trưng x=ký hiệu("x");

t=ký hiệu("t");

f=(x+2)\*(x^2+3)

der1=phân biệt(f,x)f=(t^2+3)\*(t^(1/2)+t^3)der2=phân biệt(f,t)

f=(x^2-2\*x+1)\*(3\*x^3-5\*x^2+2)der3=phân biệt(f,x)

f=(2\*x^2+3\*x)/(x^3+1)thứ 4=phân biệt(f,x)f=(x^2+1)^17

thứ 5= phân biệt(f,x)f=(t^3+3\*t^2+5\*t-9)^(-6)thứ 6=phân biệt(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:

|  |  |
| --- | --- |
| **Hàm số** | **Phát sinh** |
| **ccây rìu** | ccây rìu.ln ca (ln là logarit tự nhiên) |
| **ex** | ex |
| **ln x** | 1/x |
| **lncx** | 1/x.ln c |
| **xx** | xx.(1 + lnx) |
| **tội lỗi(x)** | cos(x) |
| **cos(x)** | -sin(x) |
| **rám nắng(x)** | giây2(x) hoặc 1/cos2(x) hoặc 1 + tan2(x) |
| **cũi(x)** | -csc2(x) hoặc -1/sin2(x) hoặc -(1 + nôi2(x)) |
| **giây(x)** | giây(x).tan(x) |
| **csc(x)** | -csc(x).cot(x) |

#### Thí dụ

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

biểu tượng x

y=kinh nghiệm(x)khác biệt(y)

y=x^9khác biệt(y)

y=tội(x)khác biệt(y)

y=rám nắng(x)khác biệt(y)

y=cos(x)khác biệt(y)

y=đăng nhập(x)khác biệt(y)

y=log10(x)khác biệt(y)

y=tội(x)^2khác biệt(y)

y=cos(3\*x^2+2\*x+1)khác biệt(y)

y=kinh nghiệm(x)/tội(x)khác biệt(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:

biểu tượng ký hiệu tải pkg

x=ký hiệu("x");y=kinh nghiệm(x)

phân biệt(y,x)

y=x^9phân biệt(y,x)

y=Tội(x)phân biệt(y,x)

y=Tân(x)phân biệt(y,x)

y=cos(x)phân biệt(y,x)

y=Nhật ký(x)phân biệt(y,x)

%gói tượng trưng khôngkhông phảicócái nàyủng hộ

%y=nhật ký10(x)

%phân biệt(y,x)

y=Tội(x)^2phân biệt(y,x)

y=cos(3\*x^2+2\*x+1)phân biệt(y,x)

y=kinh nghiệm(x)/Tội(x)phân biệt(y,x)

## Tính toán các công cụ phái sinh bậc cao hơn

Để tính các đạo hàm bậc cao 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\*kinh nghiệm(-3\*x);khác biệt(f,2)

MATLAB thực thi mã và trả về kết quả sau:

trả lời =

9\*x\*exp(-3\*x) - 6\*exp(-3\*x)

Sau đây là Octave tương đương với phép tính trên:

biểu tượng ký hiệu tải pkg

x=ký hiệu("x");

f=x\*kinh nghiệm(-3\*x);

phân biệt(f,x,2)

#### Thí 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 đó:

biểu tượng x

y=3\*tội(x)+7\*cos(5\*x);%xác địnhhàm số

lhs=khác biệt(y,2)+y;%đánh giá lhs của phương trình rhs=-5\*cos(2\*x);%rhs của phương trìnhnếu(bình đẳng(lhs,rhs))

phân tán('Vâng, phương trình đúng');khác

phân tán('Không, phương trình không đúng');

kết thúc

phân tán('Giá trị của LHS là: '),phân tán(lhs);

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

Không, phương trình không đúng Giá trị của LHS là:

-168\*cos(5\*x)

Sau đây là Octave tương đương với phép tính trên:

biểu tượng ký hiệu tải pkg

x=ký hiệu("x");

y=3\*Tội(x)+7\*cos(5\*x);%xác địnhhàm số

lhs=phân biệt(y,x,2)+y;%đánh giá lhs của phương trình rhs=-5\*cos(2\*x);%rhs của phương trình

nếu(lhs==rhs)

phân tán('Vâng, phương trình đúng');khác

phân tán('Không, phương trình không đúng');

kết thúc

phân tán('Giá trị của LHS là: '),phân tán(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 vi giá 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.

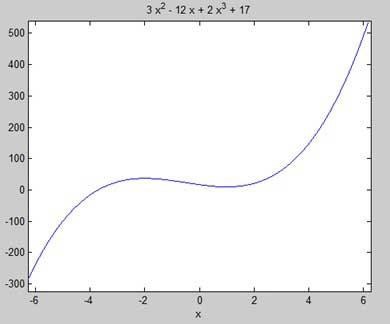
#### Thí dụ

Hãy để chúng tôi tìm các điểm dừng của hàm f(x) = 2x3 + 3x2 − 12x + 17 Thực hiện các bước sau:

1. Trước tiên, chúng ta hãy nhập hàm và vẽ đồ thị của nó:

biểu tượng x

y=2\*x^3+3\*x^2-12\*x+17;%xác địnhhàm sốezplot(y)

MATLAB thực thi mã và trả về biểu đồ sau:

Đây là mã tương đương Octave cho ví dụ trên:

biểu tượng ký hiệu tải pkg

x=ký hiệu('x');

y=nội tuyến("2\*x^3 + 3\*x^2 - 12\*x + 17");

ezplot(y)

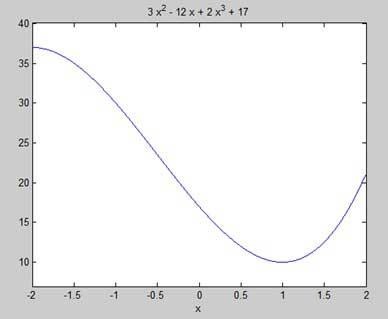
in-biểu đồ deps.tập phim

1. Mục đích của chúng ta là tìm một số cực đại và cực tiểu cục bộ trên biểu đồ, vì vậy chúng ta hãy tìm cực đại và cực tiểu cục bộ cho khoảng [-2, 2] trên biểu đồ.

biểu tượng x

y=2\*x^3+3\*x^2-12\*x+17;%xác địnhhàm sốezplot(y,[-2,2])

MATLAB thực thi mã và trả về biểu đồ sau:



Đây là mã tương đương Octave cho ví dụ trên:

biểu tượng ký hiệu tải pkg

x=ký hiệu('x');

y=nội tuyến("2\*x^3 + 3\*x^2 - 12\*x + 17");

ezplot(y,[-2,2])in-biểu đồ deps.tập phim

1. Tiếp theo, chúng ta hãy tính đạo hàm

g=khác biệt(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:

biểu tượng ký hiệu tải pkg

x=ký hiệu("x");

y=2\*x^3+3\*x^2-12\*x+17;g=phân biệt(y,x)

1. Chúng ta hãy giải hàm đạo hàm, g, để nhận các giá trị tại đó nó trở thành 0.

S=giải quyết(g)

MATLAB thực thi mã và trả về kết quả sau:

s =

1

-2

Sau đây là Octave tương đương với phép tính trên:

biểu tượng ký hiệu tải pkg

x=ký hiệu("x");

y=2\*x^3+3\*x^2-12\*x+17;g=phân biệt(y,x)rễ([6,6,-12])

1. 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 lệnh subs.

dự bị(y,1),dự bị(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:

biểu tượng ký hiệu tải pkg

x=ký hiệu("x");

y=2\*x^3+3\*x^2-12\*x+17;g=phân biệt(y,x)

rễ([6,6,-12])

dự bị(y,x,1),dự bị(y,x,-2)

Do đó, Các giá trị nhỏ nhất và lớn nhất của hàm số f(x) = 2x3 + 3x2 − 12x + 17, trong khoảng [-2,2] là 10 và 37.

## Giải phương trình vi phân

MATLAB cung cấp lệnh dsolve để giải các phương trình vi phân một cách tượng trưng.

Dạng cơ bản nhất của lệnh dsolve để tìm nghiệm cho một phương trình là:

giải quyết('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 các điều kiện ban đầu và điều kiện biên cho vấn đề, dưới dạng danh sách được phân cách bằng dấu phẩy theo phương trình như sau:

giải quyết('eqn','điều kiện1','điều kiện2',…)

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 vi phân cấp một: y' = 5y.

S=giải quyết('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 vi phân cấp hai như sau: y" - y = 0, y(0) = -1, y'(0) = 2.

giải quyết('D2y - y = 0','y(0) = -1','Dy(0) = 2')

MATLAB thực thi mã và trả về kết quả sau:

trả lời =

exp(t)/2 - (3\*exp(-t))/2

# Hội nhập

Tôihội nhập thỏa thuậns với hai loại vấn đề cơ bản khác nhau.

CHƯƠNG

25

* + 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 phản vi phân, hay tìm nguyên hàm, hay tìm tích phân bất định.
  + 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 định nghĩa của tích phân xác định.

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 x2 là 2x, nên chúng ta có thể nói rằng tích phân không xác định của 2x là x2.

Trong các ký hiệu:

**f'(x2) = 2 lần**, vì thế,

∫ 2xdx = x2.

Tích phân bất định không phải là duy nhất, bởi vì đạo hàm của x2 + 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 = x2 + c.

Trong đó, c được gọi là 'hằng số tùy ý'.

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 hàm, chúng tôi viết:

int(f);

Ví dụ: từ ví dụ trước của chúng tôi:

biểu tượng 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:

biểu tượng xnint(ký hiệu(x^N))

f='tội lỗi (n \* t)'

int(ký hiệu(f))biểu tượng tại

int(một\*cos(số Pi\*t))int(một^x)

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

trả lời =

từng phần([n == -1, log(x)], [n ~= -1, x^(n + 1)/(n + 1)]) f =

sin(n\*t) ans =

-cos(n\*t)/ntrả lời =

(a\*sin(pi\*t))/pitrả lời =

a^x/log(a)

#### ví dụ 2

Tạo một tệp script và nhập mã sau vào đó:

biểu tượng xnint(cos(x))

int(kinh nghiệm(x))

int(đăng nhập(x))

int(x^-1)int(x^5\*cos(5\*x))đẹp(int(x^5\*cos(5\*x)))int(x^-5)

int(giây(x)^2)

đẹp(int(1-10\*x+9\*x^2))int((3+5\*x-6\*x^2-7\*x^3)/2\*x^2)

đẹp(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

24

24 cos(5x)24 x tội lỗi (5x)12 xcos(5 x)xcos(5 x)

+-+-

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3125 |  | 625 |  | 125 |  | 5 |
| 3 |  | 5 |  |  |  |  |

4xsin(5 x)xsin(5 x)

+ 255

trả lời =

-1/(4\*x^4)

ans = tan(x)

2

x (3x- 5 x + 1)

trả lời =

- (7\*x^6)/12 - (3\*x^5)/5 + (5\*x^4)/8 + x^3/2

6543

7x3 x5 xx

--++

12582

## 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,một,b)

ExampleVí dụ, để tính giá trịcủa 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:

biểu tượng ký hiệu tải pkg

x=ký hiệu("x");f=x;

c=[1,0];

tích phân=đa giác(c);

một=đa giác(tích phân,9)-đa giác(tích phân,4);trưng bày('Diện tích: '),phân tán(gấp đôi(một));

Một giải pháp thay thế có thể được đưa ra bằng cách sử dụng hàm quad() do Octave cung cấp như sau:

biểu tượng ký hiệu tải pkg

f=nội tuyến("x");

[một,lỗi lầm,nfneval]=quad(f,4,9);

trưng bày('Diện tích: '),phân tán(gấp đôi(một));

#### ví dụ 1

Chúng ta hãy tính diện tích bao quanh giữa trục x và đường cong y = x3−2x+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;một=int(f,1,2)

trưng bày('Diện tích: '),phân tán(gấp đôi(một));

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

một = 23/4

Diện tích:

5.7500

Sau đây là Octave tương đương với phép tính trên:

biểu tượng ký hiệu tải pkg

x=ký hiệu("x");

f=x^3-2\*x+5;c=[1,0,-2,5];

tích phân=đa giác(c);

một=đa giác(tích phân,2)-đa giác(tích phân,1);trưng bày('Diện tích: '),phân tán(gấp đôi(một));

Một giải pháp thay thế có thể được đưa ra bằng cách sử dụng hàm quad() do Octave cung cấp như sau:

biểu tượng ký hiệu tải pkg

x=ký hiệu("x");

f=nội tuyến("x^3 - 2\*x +5");

[một,lỗi lầm,nfneval]=quad(f,1,2);trưng bày('Diện tích: '),phân tán(gấp đôi(một));

#### ví dụ 2

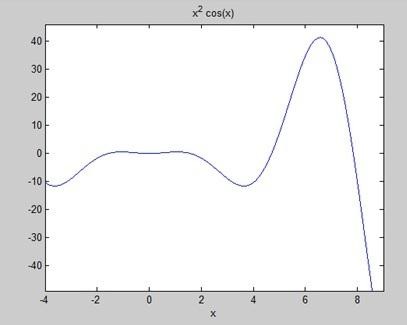
Tìm diện tích dưới đường cong: f(x) = x2 cos(x) với −4 ≤ x ≤ 9. Tạo một tệp kịch bản và viết đoạn mã sau:

f=x^2\*cos(x);

ezplot(f,[-4,9])

một=int(f,-4,9)

phân tán('Diện tích: '),phân tán(gấp đôi(một));

Khi bạn chạy tệp, MATLAB vẽ đồ thị:

và hiển thị kết quả sau:

một =

8\*cos(4) + 18\*cos(9) + 14\*sin(4) + 79\*sin(9) Diện tích:

0,3326

Sau đây là Octave tương đương với phép tính trên:

biểu tượng ký hiệu tải pkg

x=ký hiệu("x");

f=nội tuyến("x^2\*cos(x)");

ezplot(f,[-4,9])in-biểu đồ deps.tập phim

[một,lỗi lầm,nfneval]=quad(f,-4,9);

trưng bày('Diện tích: '),phân tán(gấp đôi(một));

# đa thức

CHƯƠNG

26

mđại diện ATLABgửi đa thức dưới dạng vectơ hàng chứa các hệ số được sắp xếp theo lũy thừa giảm dần. Ví dụ: phương trình P(x) = x4 + 7x3 - 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 ta, tại x = 4, hãy gõ:

P=[170-59];

đa giác(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];

đa giác(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=rễ(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. Ví dụ:

p2=nhiều(r)

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=đa năng(x,y,N)

#### Thí dụ

Tạo một tệp script và nhập mã sau:

x=[123456];y=[5.543.1128290.7498.4978.67];%dữ liệu

P=đa năng(x,y,4)%đượcđa thức

%tính toáncác giá trị của ước tính polyfit trên một phạm vi tốt hơn,

%vàvẽ đồ thị ước tính trên các giá trị dữ liệu thựcvìso sánh:x2=1:.1:6;

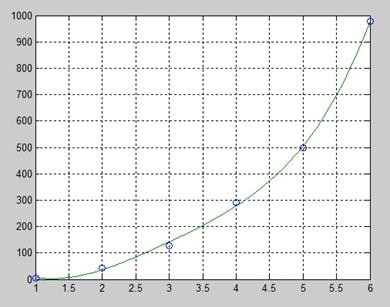
y2=đa giác(P,x2);kịch bản(x,y,'o',x2,y2)lưới điện trên

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

p =

4.1056-47.9607222.2598 -362.7453191.1250

Và vẽ đồ thị sau:



# biến đổi

CHƯƠNG

27

mATLAB chuyên nghiệplệnh vides để làm việc với các phép biến đổi, chẳng hạn như các phép biến đổi Laplace và Fourier. 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 vi 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 vi phân thành phương trình đại số. Để tính biến đổi Laplace của hàm f(t), hãy viết:

laplace(f(t))

#### Thí 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:

laplace syms(một)laplace(t^2)laplace(t^9)laplace(kinh nghiệm(-b\*t))laplace(tội(w\*t))laplace(cos(w\*t))

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)

trả lời =

w/(s^2 + w^2)

trả lời =

s/(s^2 + w^2)

## Biến đổi Laplace nghịch đảo

MATLAB cho phép chúng ta tính toán phép biến đổi Laplace nghịch đảo bằng lệnh ilaplace. Ví dụ,

ilaplace(1/S^3)

MATLAB sẽ thực hiện câu lệnh trên và hiển thị kết quả:

ans = t^2/2

#### Thí dụ

Tạo một tệp script và nhập mã sau:

syms ổn định ở ilaplace(1/S^7)ilaplace(2/(w+S))ilaplace(S/(S^2+4))ilaplace(kinh nghiệm(-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)

trả lời =

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.

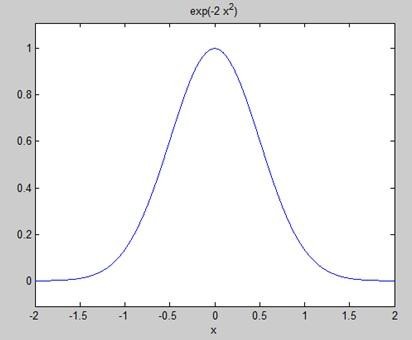
#### Thí dụ

Tạo một tệp script và nhập mã sau vào đó:

biểu tượng x

f=kinh nghiệm(-2\*x^2);%chức năng của chúng tôiezplot(f,[-2,2])%âm mưu củachức năng của chúng tôiFT=chiến binh(f)% Fourierbiến đổi

Khi bạn chạy tệp, MATLAB vẽ biểu đồ sau:



Và hiển thị kết quả sau:

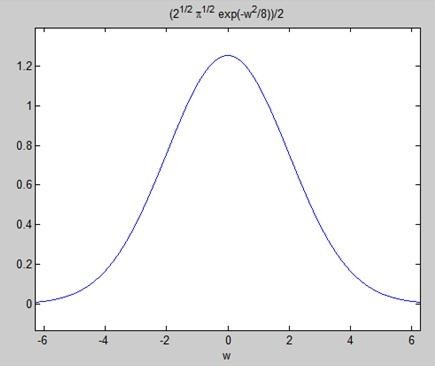
ft =

(2^(1/2)\*pi^(1/2)\*exp(-w^2/8))/2

Vẽ đồ thị biến đổi Fourier như sau:

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. Ví dụ,

f=ifourier(-2\*kinh nghiệm(-cơ bụng(w)))

MATLAB sẽ thực hiện câu lệnh trên và hiển thị kết quả:

f =

-2/(pi\*(x^2 + 1))

# GNU Quãng tám

CHƯƠNG

28

gNbạnOctave là một ngôn ngữ lập trình cấp cao giống như MATLAB và nó hầu như tương thích với MATLAB. Nó cũng được sử dụng để tính toán số.

Octave có các tính năng chung sau với MATLAB:

* ma trận là kiểu dữ liệu cơ bản
* nó có hỗ trợ tích hợp cho các số phức
* nó có các hàm và thư viện toán học tích hợp
* 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

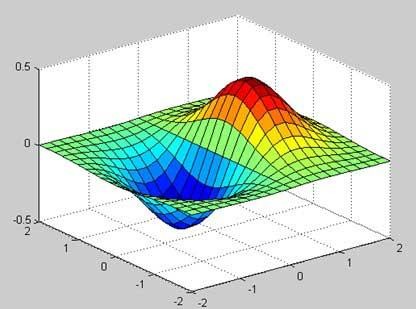
Hầu như tất cả các ví dụ trong hướng dẫn này đều tương thích với MATLAB cũng như Octave. Hãy thử ví dụ sau trong MATLAB và Octave, ví dụ này tạo ra kết quả tương tự mà không có bất kỳ thay đổi cú pháp nào:

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]=lưới điện(-2:.2:2);

g=x.\*kinh nghiệm(-x.^2-y.^2);lướt sóng(x,y,g)

in-biểu đồ deps.tập phim

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 vi 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.

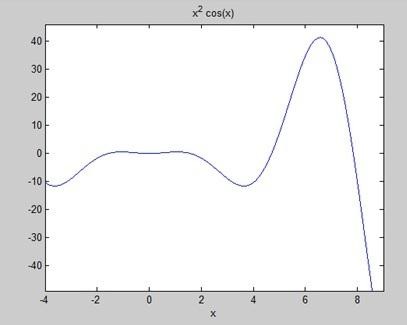
Xem xét ví dụ sau trong đó MATLAB và Octave sử dụng các hàm khác nhau để tính diện tích đường cong: f(x) = x2 cos(x) với −4 ≤ x ≤ 9. Sau đây là phiên bản mã MATLAB:

f=x^2\*cos(x);ezplot(f,[-4,9])

một=int(f,-4,9)

phân tán('Diện tích: '),phân tán(gấp đôi(một));

Khi bạn chạy tệp, MATLAB vẽ đồ thị:



và hiển thị kết quả sau:

một =

8\*cos(4) + 18\*cos(9) + 14\*sin(4) + 79\*sin(9) Diện tích:

0,3326

Nhưng để đưa ra diện tích của cùng một đường cong trong Octave, bạn sẽ phải sử dụng gói ký hiệu như sau:

biểu tượng ký hiệu tải pkg

x=ký hiệu("x");

f=nội tuyến("x^2\*cos(x)");

ezplot(f,[-4,9])in-biểu đồ deps.tập phim

[một,lỗi lầm,nfneval]=quad(f,-4,9);

trưng bày('Diện tích: '),phân tán(gấp đôi(một));

# simulink

CHƯƠNG

29

Simulink là môi trường thiết kế dựa trên mô hình và mô phỏng cho các hệ thống động và nhúng, được tích hợp với MATLAB. Simulink, cũng được phát triển bởi MathWorks, là một công cụ ngôn ngữ lập trình đồ họa luồng dữ liệu để lập mô hình, 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 thư viện khối có thể tùy chỉnh.

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:

* + thiết kế cấp hệ thống
  + mô phỏng
  + tạo mã tự động
  + kiểm tra và xác minh hệ thống nhúng

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:

* + **Dòng trạng thái**cho phép phát triển các máy trạng thái và lưu đồ.
  + **Bộ giải mã Simulink**cho phép tự động tạo mã nguồn C để triển khai hệ thống theo thời gian thực.
  + **Mục tiêu xPC**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 hệ thống vật lý.
  + **Bộ giải mã nhúng**hỗ trợ các mục tiêu nhúng cụ thể.
  + **Bộ giải mã HDL**cho phép tự động tạo VHDL và Verilog tổng hợp
  + **SimEvents**cung cấp một thư viện các khối xây dựng đồ họa để mô hình hóa các hệ thống xếp hàng

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 vi mô hình.

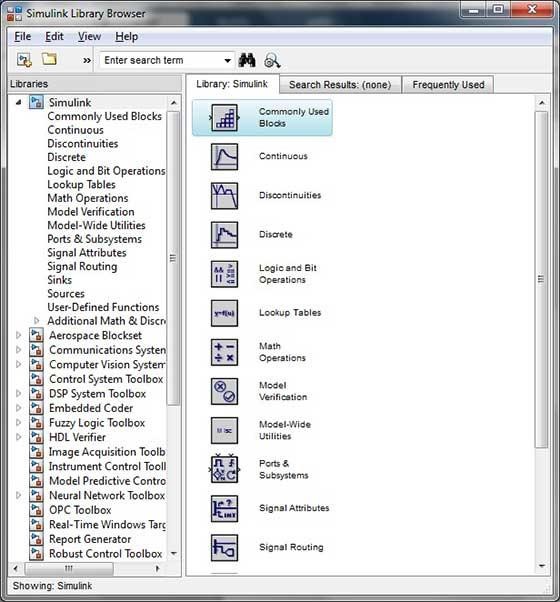
Simulink Design Verifier cho phép bạn xác định các lỗi thiết kế và tạo các kịch bản trường hợp thử nghiệm để kiểm tra mô hình.

## Sử dụng Simulink

Để mở Simulink, gõ vào không gian làm việc MATLAB:

simulink

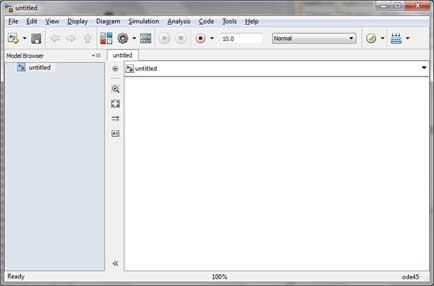
Simulink mở bằng Trình duyệt thư viện. Trình duyệt thư viện được sử dụng để xây dựng các mô hình mô phỏng.



Trên khung cửa sổ bên trái, bạn sẽ tìm thấy một số thư viện được phân loại trên cơ sở của các hệ thống khác nhau, nhấp vào từng thư viện 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 viện. 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.

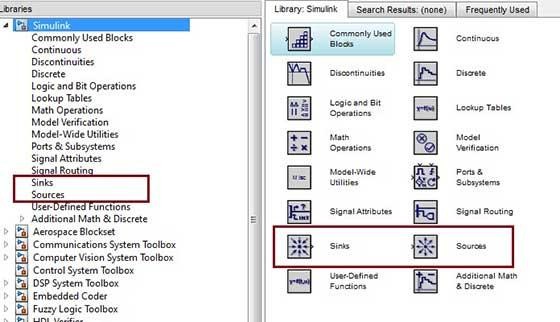
Các phần tử mô hình được thêm vào bằng cách chọn các phần tử thích hợp từ Trình duyệt thư viện và kéo chúng vào cửa sổ Mô hình.

Ngoài ra, bạn có thể sao chép các phần tử mô hình và dán chúng vào cửa sổ mô hình.

#### ví dụ

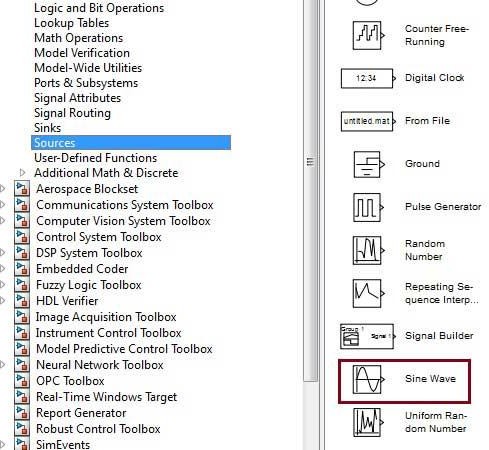
Kéo và thả các mục từ thư viện Simulink để tạo dự án của bạn.

Với mục đích của ví dụ này, 2 khối sẽ được sử dụng cho mô phỏng - Nguồn (tín hiệu) và aSink (phạm vi). 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).

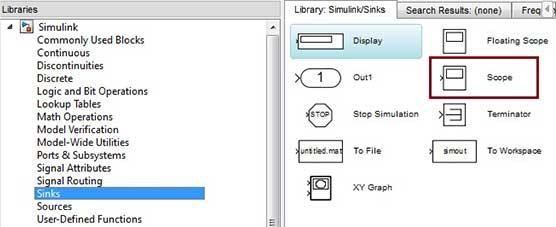


Bắt đầu bằng cách kéo các khối cần thiết từ thư viện vào cửa sổ dự án. Sau đó, kết nối các khối lại với nhau, có thể thực hiện bằng cách kéo các đầu nối từ các điểm kết nối trên một khối này sang các điểm khác.

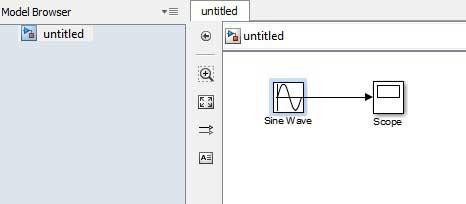
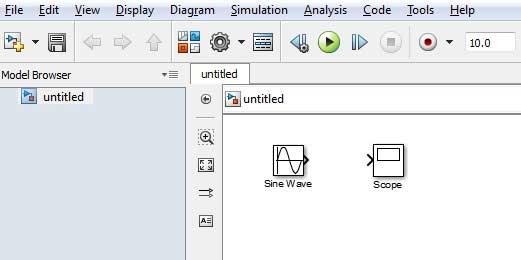
Hãy để chúng tôi kéo một khối 'Sóng hình sin' vào mô hình.



Chọn 'Sinks' từ thư viện 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 vi.

