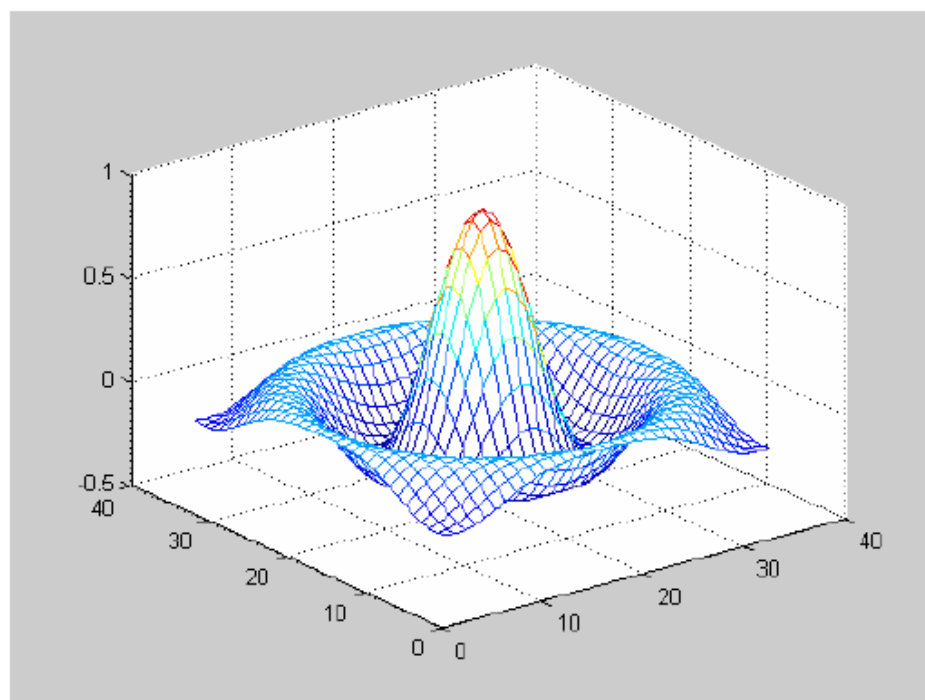


**ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**

GIÁO TRÌNH
MATLAB



Dành cho sinh viên các ngành kỹ thuật

Tác giả : PHAN - THANH - TAO

Đà Nẵng - 2004

GIỚI THIỆU

Matlab là một phần mềm toán học của hãng **Mathworks** để tính toán trên các số và có tính trực quan rất cao.

Matlab đã qua nhiều phiên bản, giáo trình này giới thiệu phiên bản 7.0 (release 14).

Matlab là viết tắt của Matrix Laboratory. **Matlab** làm việc chủ yếu với các ma trận. Ma trận cỡ $m \times n$ là bảng số chữ nhật gồm $m \times n$ số được sắp xếp thành m hàng và n cột. Trường hợp $m=1$ hoặc $n=1$ thì ma trận trở thành vector dòng hoặc cột; trường hợp $m=n=1$ thì ma trận trở thành một đại lượng vô hướng. Nói chung, **Matlab** có thể làm việc với nhiều kiểu dữ liệu khác nhau. Với xâu chữ (chuỗi ký tự) **Matlab** cũng xem là một dãy các ký tự hay là dãy mã số của các ký tự.

Matlab dùng để giải quyết các bài toán về giải tích số, xử lý tín hiệu số, xử lý đồ họa, ... mà không phải lập trình cổ điển.

Hiện nay, **Matlab** có đến hàng ngàn lệnh và hàm tiện ích. Ngoài các hàm cài sẵn trong chính ngôn ngữ, **Matlab** còn có các lệnh và hàm ứng dụng chuyên biệt trong các Toolbox, để mở rộng môi trường **Matlab** nhằm giải quyết các bài toán thuộc các phạm trù riêng. Các Toolbox khá quan trọng và tiện ích cho người dùng như toán sơ cấp, xử lý tín hiệu số, xử lý ảnh, xử lý âm thanh, ma trận thưa, logic mờ,...

Người dùng cũng có thể tạo nên các hàm phục vụ cho chuyên môn của mình, lưu vào tệp M-file để dùng về sau.

Cần tính toán bằng công thức thì có thể dùng Toolbox SYMBOLIC. Để có được $f = \cos(x)$ bằng cách lấy đạo hàm của $g = \sin(x)$ thì dùng lệnh $f = \text{diff}(\sin(x))$. Ngược lại để có g là tích phân bất định của f thì dùng lệnh $g = \text{int}(f)$.

Matlab còn có giao diện đồ họa khá đẹp mắt và dễ sử dụng. Người dùng có thể tính toán và tạo nên các hình ảnh đồ họa 2, 3 chiều cho trình ứng dụng của mình. Với các hình ảnh, nếu không chỉ định về canh trục, phối màu thì **Matlab** thực hiện tự động một cách khá phù hợp.

Vì tính mạnh mẽ để trợ giúp giải nhanh các bài toán kỹ thuật, chúng tôi cố gắng biên soạn tài liệu này để phục vụ một ít kiến thức cơ bản cho bạn đọc. Tuy nhiên, trên cơ sở đó bạn đọc có thể tự khai thác thêm các thành phần dùng riêng cho mình trong các Toolbox và Simulink.

Lần đầu xuất bản nên không thể tránh khỏi thiếu sót. Rất mong ý kiến đóng góp quý báu của bạn đọc.

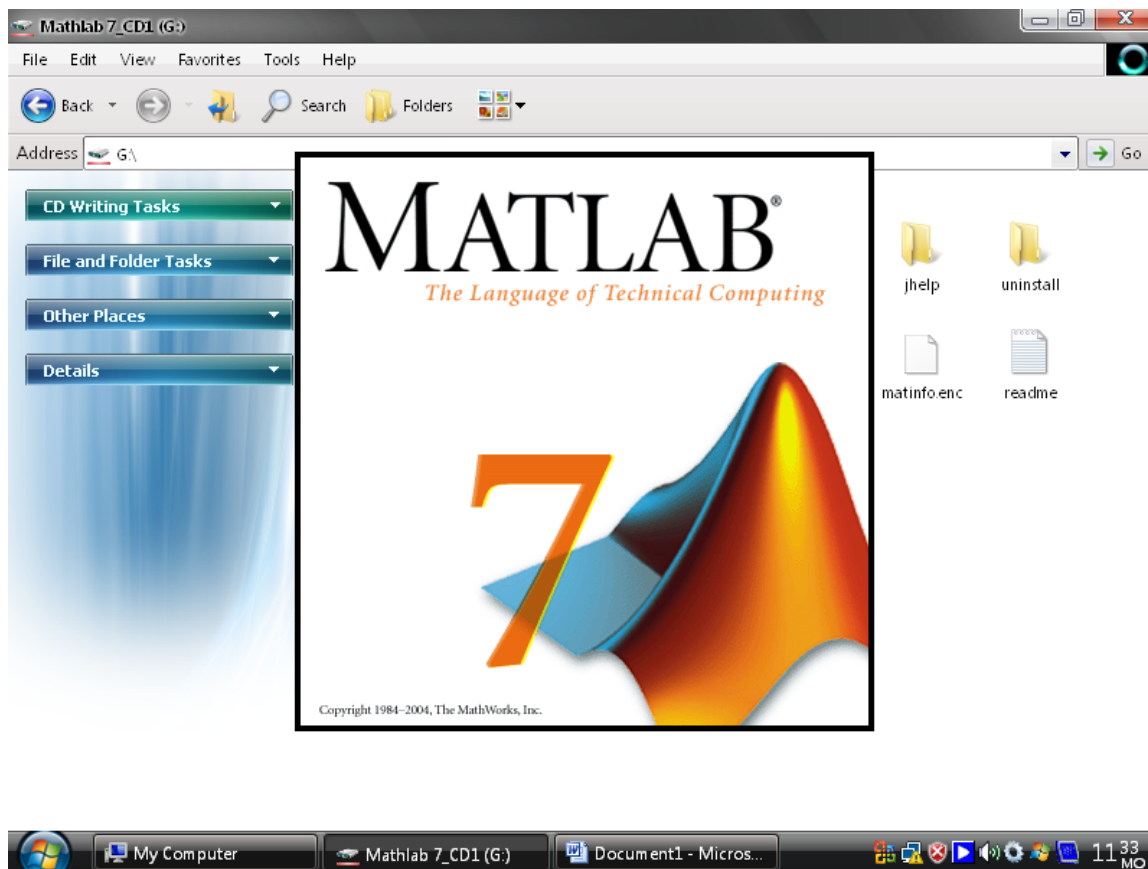
Đà Nẵng, ngày 20/02/2004

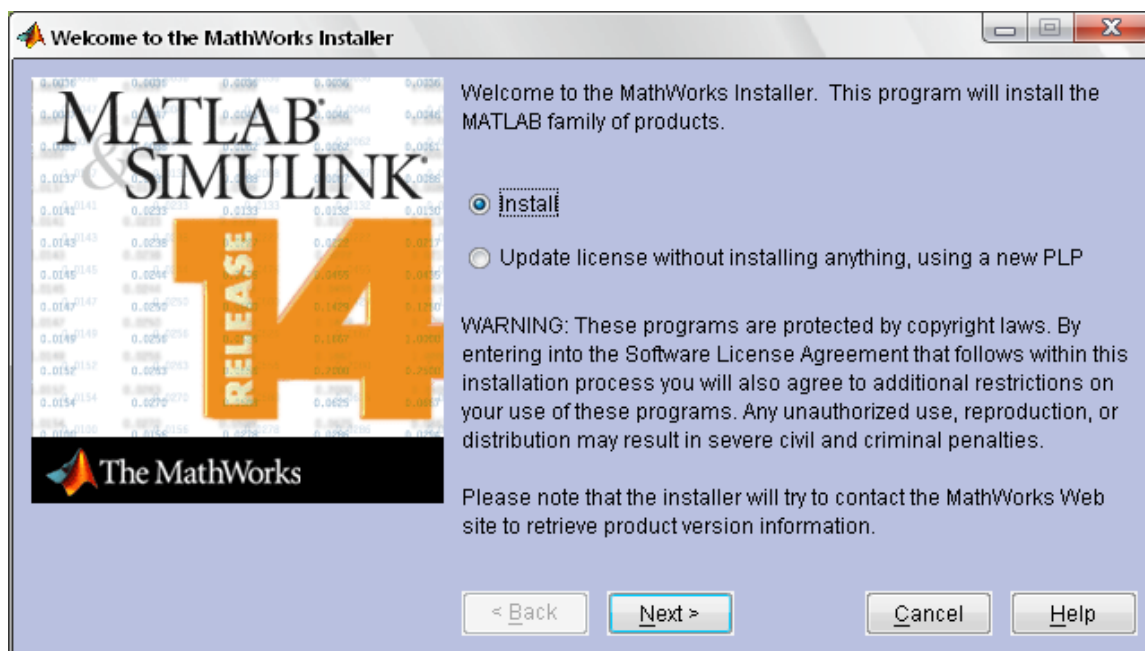
Tác giả

Phan Thanh Tao

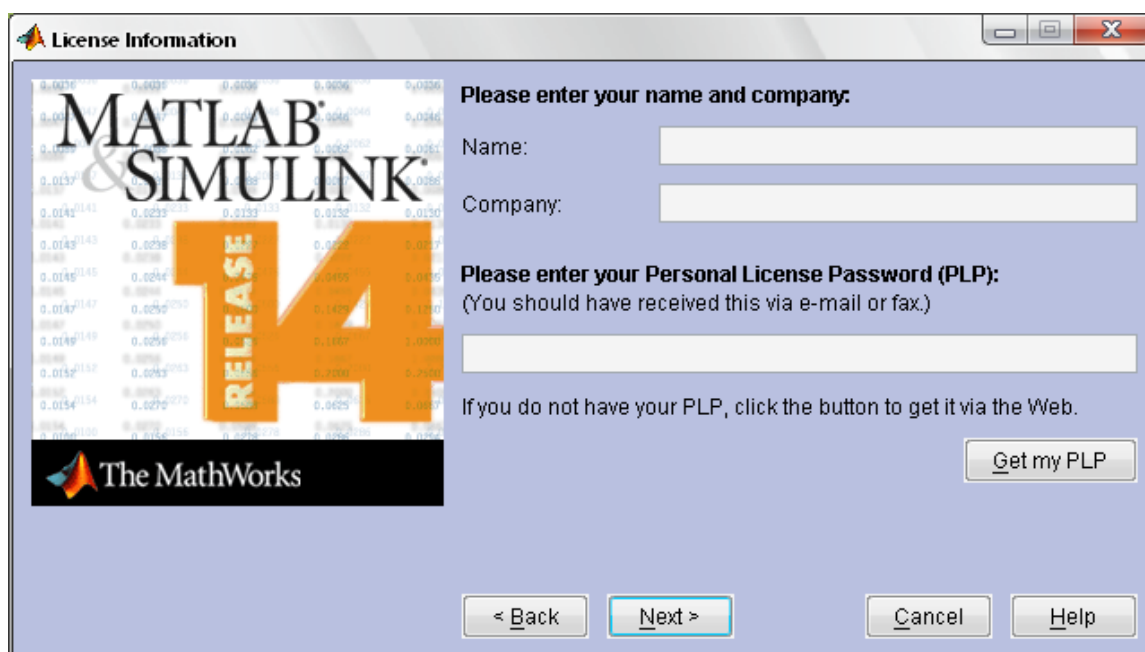
Hướng dẫn cài đặt MATLAB 7.0

Bạn hãy đưa đĩa CD vào ổ đĩa, chương trình **autorun** sẽ chạy và trên màn hình xuất hiện

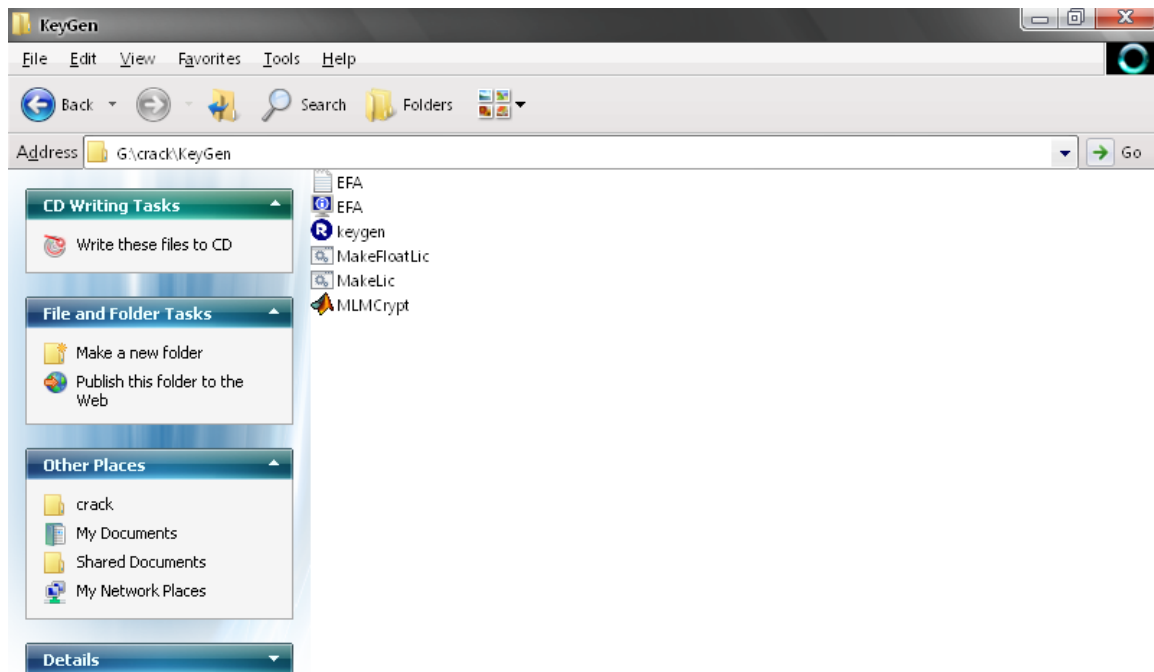




Ấn nút **Next** để tiếp tục.



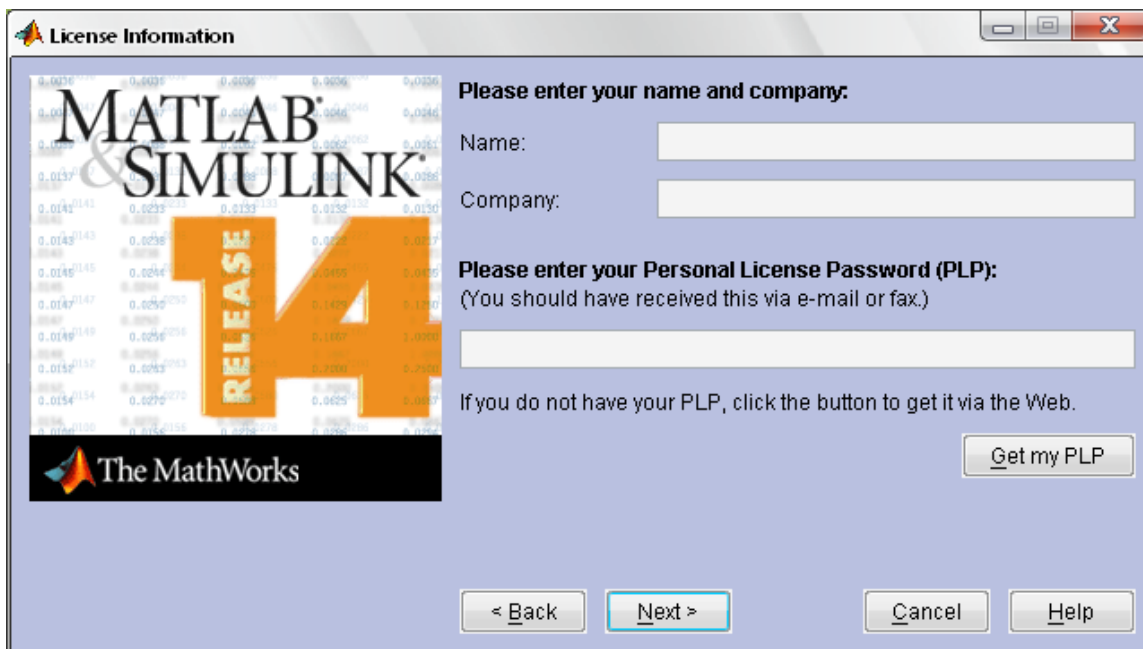
Xuất hiện màn hình yêu cầu nhập thông tin cá nhân và mật khẩu bản quyền.



Bạn gọi chương trình My Computer để mở đĩa CD và chạy chương trình \crack\keygen.exe để phát sinh mã mật khẩu.



Ấn Ctrl+C để chép mật khẩu sang Clipboard. Rồi đóng cửa sổ này lại.



The image shows the 'License Information' dialog box for MATLAB & SIMULINK. On the left is a graphic with the MATLAB & SIMULINK logo and a large orange 'RELEASE' stamp. The main area contains the following text and fields:

Please enter your name and company:

Name:

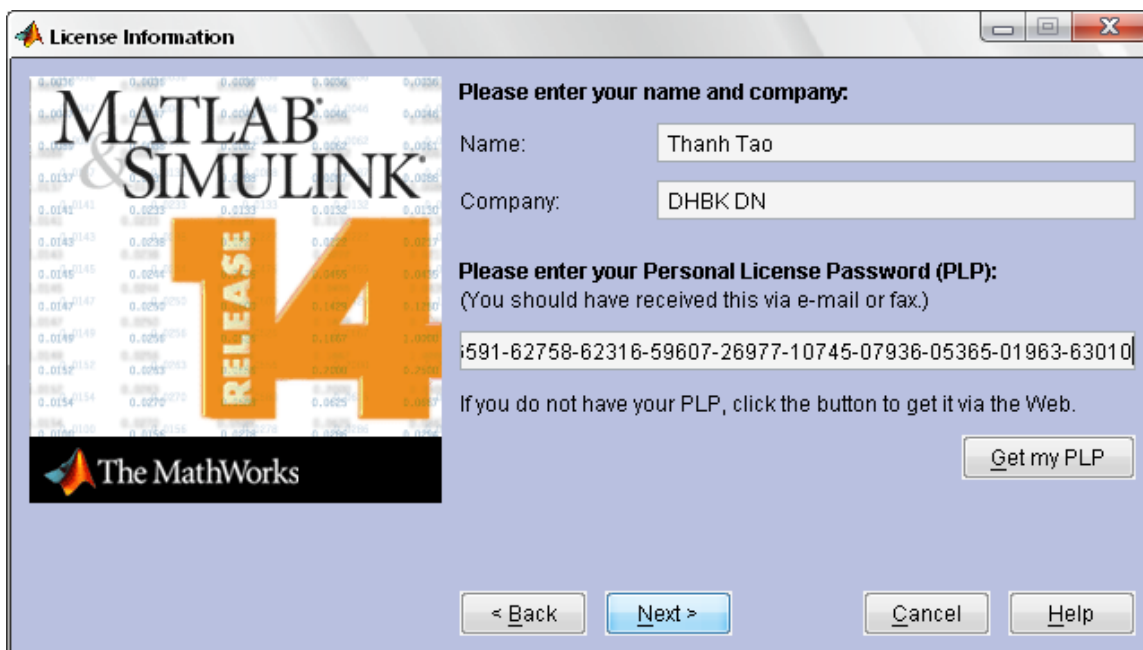
Company:

Please enter your Personal License Password (PLP):
(You should have received this via e-mail or fax.)

If you do not have your PLP, click the button to get it via the Web.

At the bottom are four buttons: '< Back', 'Next >', 'Cancel', and 'Help'.

Quay lại cửa sổ cài đặt.



This image shows the same 'License Information' dialog box, but with the fields filled out:

Please enter your name and company:

Name:

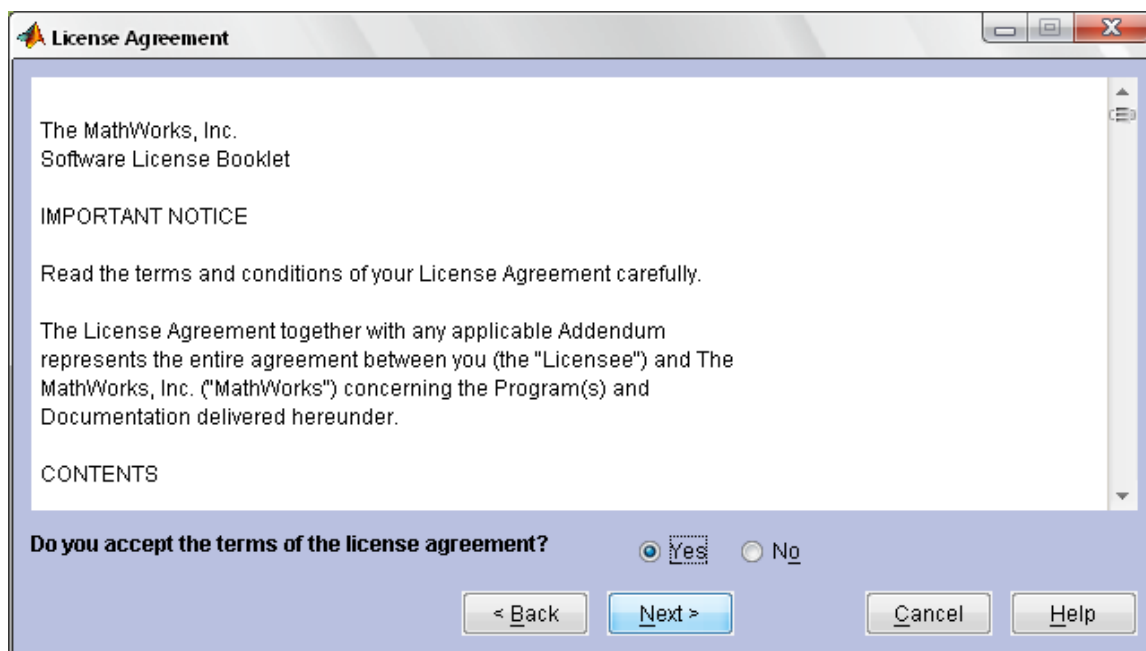
Company:

Please enter your Personal License Password (PLP):
(You should have received this via e-mail or fax.)

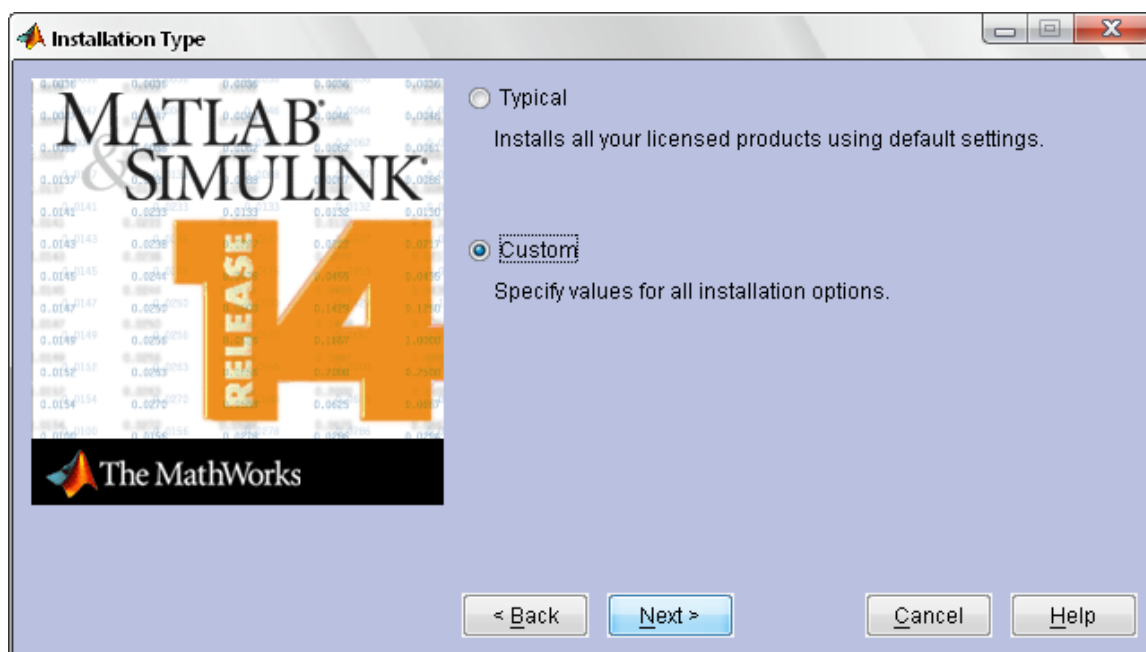
If you do not have your PLP, click the button to get it via the Web.

The bottom buttons remain the same: '< Back', 'Next >', 'Cancel', and 'Help'.

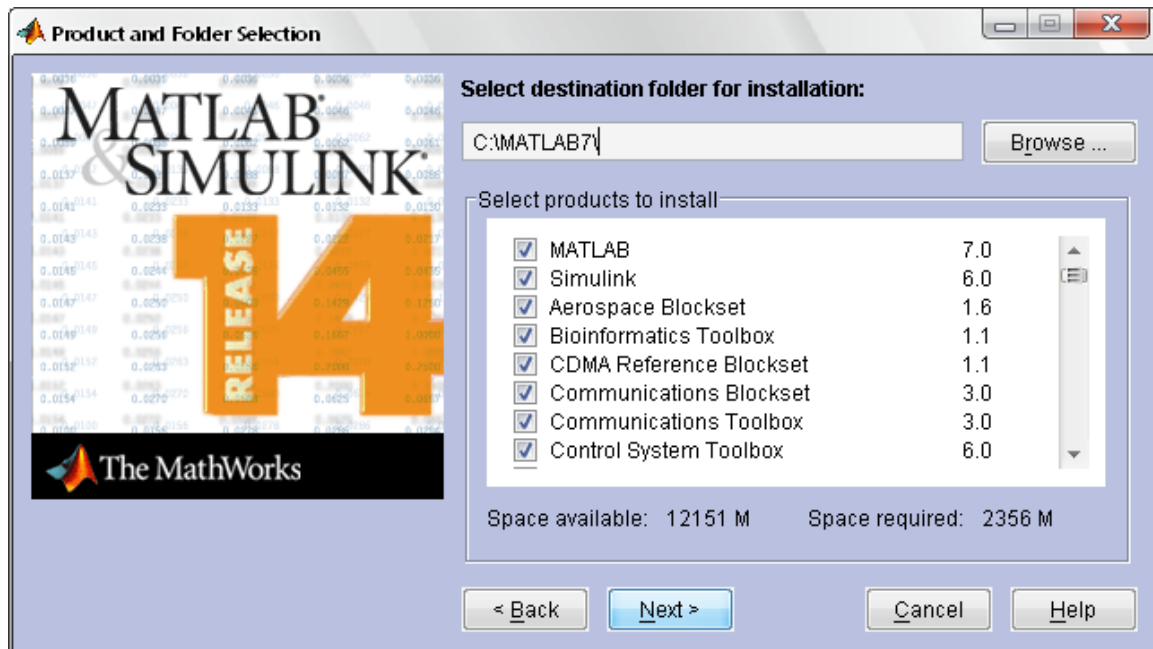
Vào ô (PLP) ấn Ctrl+V để dán mã mật khẩu vào.
Ấn nút **N**ext để tiếp tục.



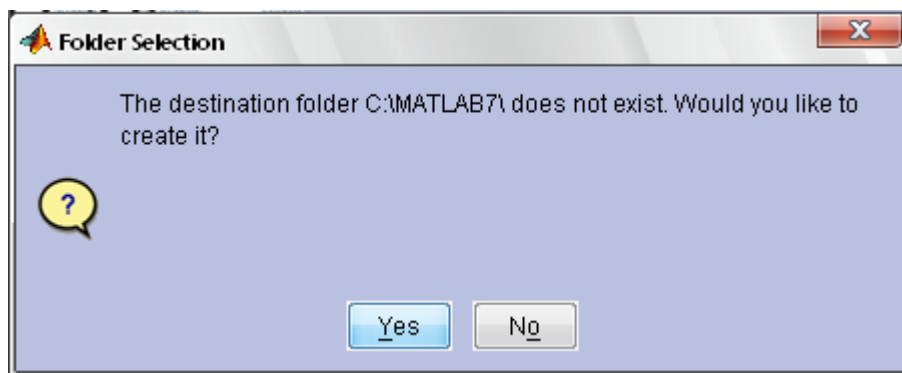
Đánh dấu **Yes** để đồng ý về bản quyền, rồi ấn nút **Next** để tiếp tục.



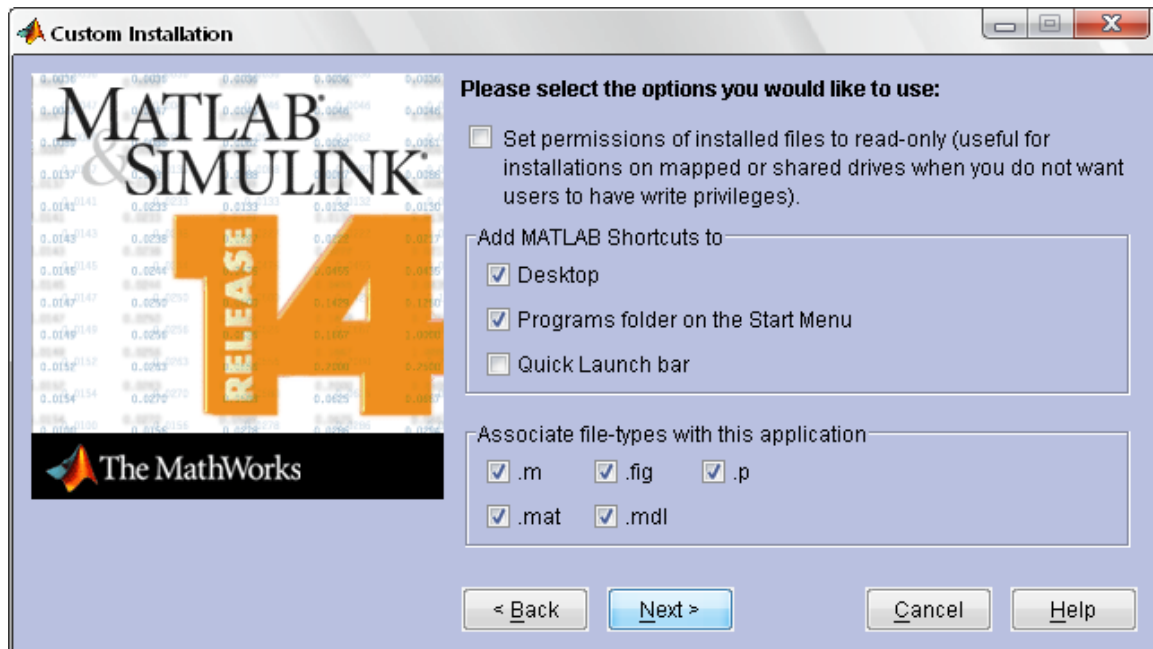
Để cài đặt đầy đủ, đánh dấu **Custom** và ấn nút **Next** để tiếp tục.



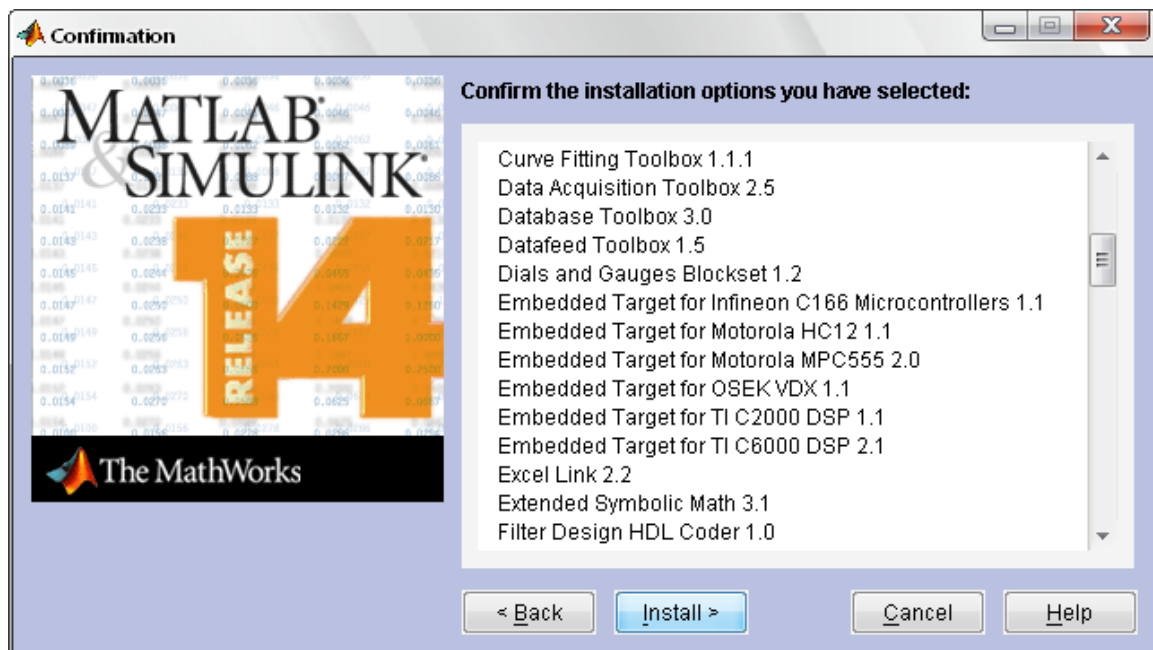
Ấn nút **N**ext để tiếp tục.



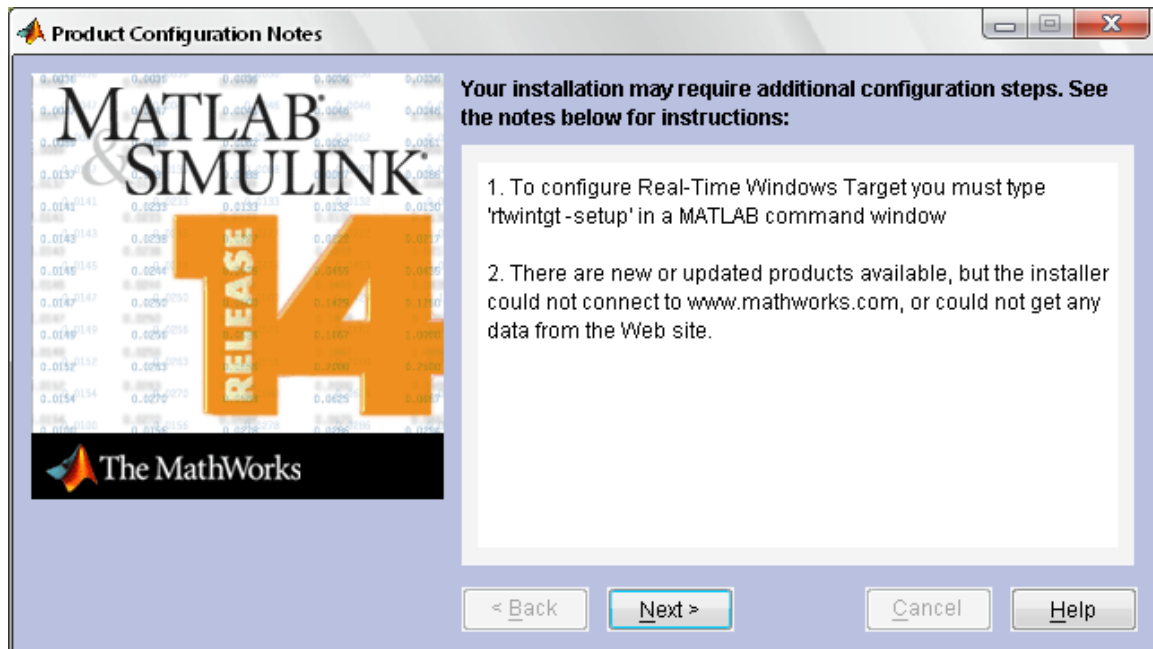
Nếu cài đặt lần đầu trên máy thì chưa có thư mục **MATLAB7**, hỏi có chấp nhận tạo thư mục mới. Ấn **Y**es để tiếp tục.



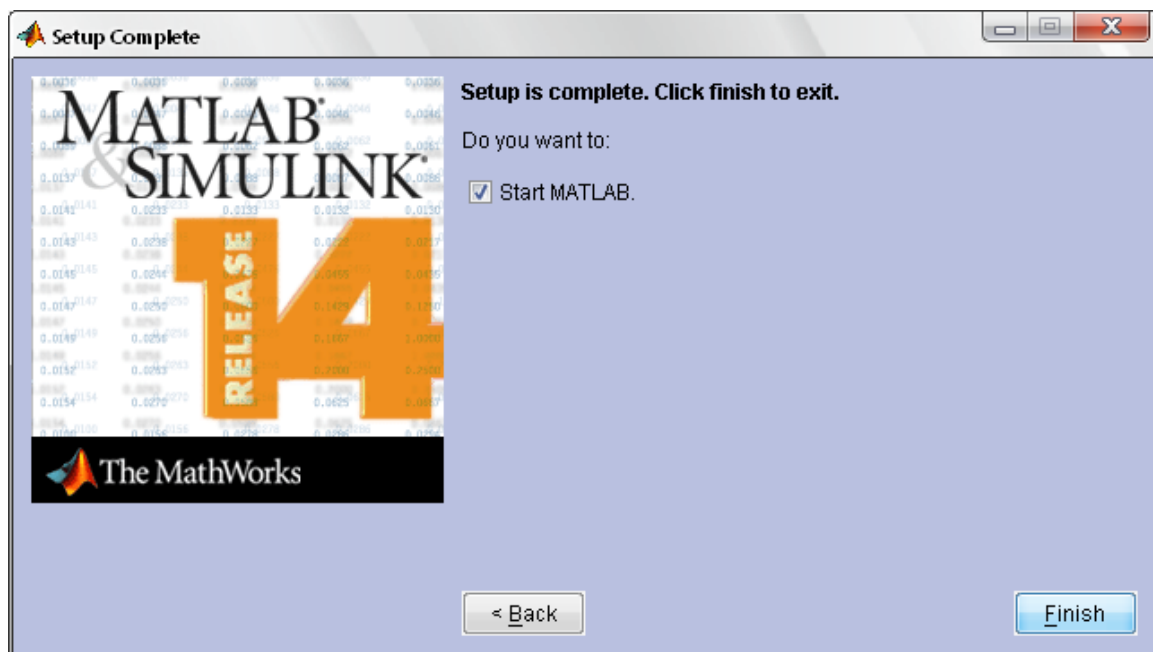
Ấn nút **N**ext để tiếp tục.



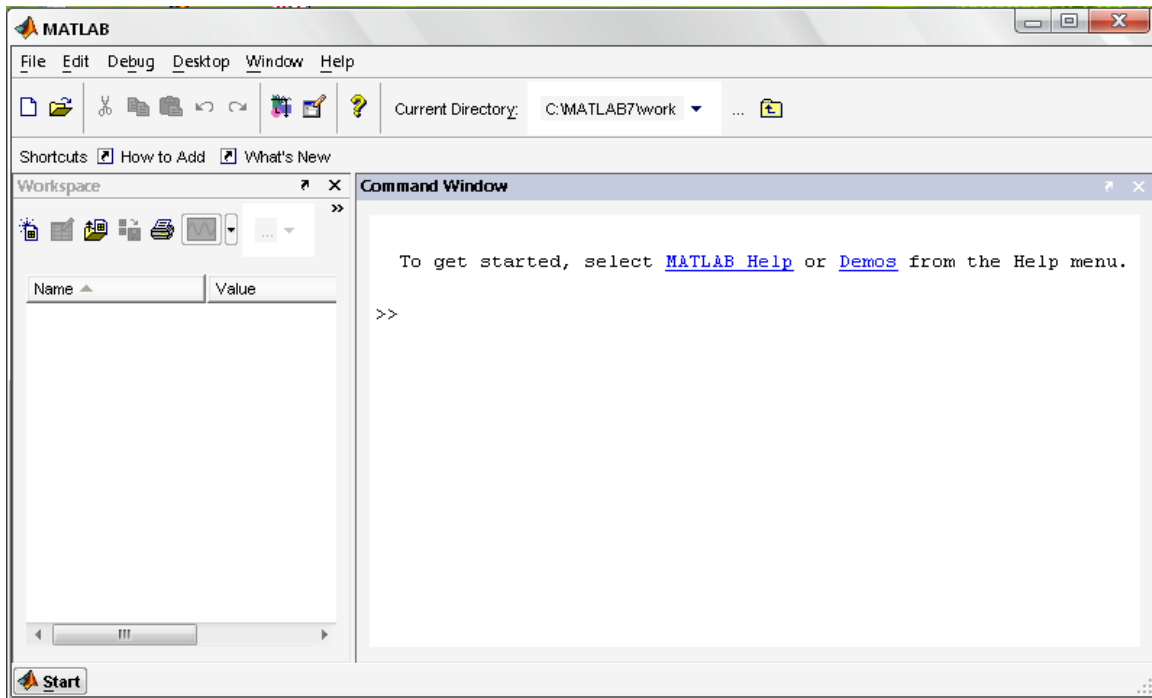
Ấn nút **I**nstall để bắt đầu cài đặt.



Xem hướng dẫn cài thêm sau này. Ấn nút **N**ext để tiếp tục.



Ấn nút **F**inish để hoàn thành việc cài đặt.



Bắt đầu vào môi trường **MATLAB**. Bạn nên chạy các chương trình mẫu để xem bằng cách ấn nút **Demos** hoặc nhập lệnh **Demo** ở dòng lệnh, sau dấu nhắc **>>**.

Chương 1. CÁC KHÁI NIỆM CƠ BẢN

MATLAB chỉ làm việc chủ yếu với các loại đối tượng là ma trận số có thể là số phức. Trong trường hợp đặc biệt, có thể là ma trận cấp 1 là các vô hướng, và các *ma trận dòng* hoặc *ma trận cột* là các vectơ.

Hãy bắt đầu với cách nhập ma trận cho **MATLAB**.

1.1. Nhập ma trận đơn giản

Ma trận có thể nhập cho **MATLAB** bằng nhiều cách:

- Nhập danh sách rõ ràng các phần tử.
- Phát sinh bằng các lệnh và hàm gắn liền.
- Tạo ra từ siêu tệp (M-file).
- Nạp từ các tệp dữ liệu bên ngoài.

Ngôn ngữ **MATLAB** không chứa các lệnh khai báo kích thước hoặc khai báo kiểu. Việc lưu trữ là tự động.

Cách dễ nhất của việc nhập ma trận là nhập danh sách rõ ràng các phần tử. Danh sách các phần tử cách nhau ký tự trống hoặc dấu phẩy, đặt trong cặp ngoặc vuông, [và], và dùng dấu chấm phẩy(;) để biểu hiện kết thúc dòng. Ví dụ, nhập lệnh

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

kết quả xuất là

```
A =  
    1    2    3  
    4    5    6  
    7    8    9
```

Ma trận **A** được lưu để sử dụng về sau .

Ma trận lớn có thể được tách ra thành nhiều dòng, sang dòng thay cho dấu chấm phẩy. Mặc dù ít cần ma trận kích thước này, nhưng ma trận trên cũng có thể tách ra thành 3 dòng nhập như sau

```
A = [ 1    2    3  
      4    5    6  
      7    8    9 ]
```

Các ma trận có thể nhập từ tệp với tên mở rộng là **".m"** . Nếu tệp có tên là **gena.m** chứa ba dòng văn bản

```
A = [ 1  2  3
      4  5  6
      7  8  9 ]
```

thì lệnh **gena** đọc tệp và phát sinh ra ma trận **A**.

Lệnh **load** có thể đọc các ma trận phát sinh từ các phần khác trước đó của **MATLAB** hoặc các ma trận ở dạng **ASCII** xuất từ các chương trình khác. Sẽ biết thêm sau này.

1.2. Các phần tử của ma trận

Các phần tử của ma trận có thể là biểu thức **MATLAB** bất kỳ; ví dụ, lệnh

```
x = [ -1.3 sqrt(3) (1+2+3)*4/5 ]
```

kết quả là

```
x =
-1.3000  1.7321  4.8000
```

Các phần tử riêng biệt của ma trận có thể được tham chiếu với các chỉ số bên trong cặp ngoặc đơn, (và). Tiếp ví dụ trên, lệnh

```
x(5) = abs(x(1))
```

cho ra

```
x =
-1.3000  1.7321  4.8000  0.0000  1.3000
```

Lưu ý rằng kích thước của **x** được tự động tăng để phù hợp với các phần tử mới, và các phần tử trong khoảng không xác định được đặt giá trị *không*.

Ma trận lớn có thể được xây dựng bằng cách dùng các ma trận nhỏ như các phần tử. Ví dụ, có thể đưa thêm một dòng khác vào ma trận **A** với lệnh

```
r = [ 10 11 12 ];
A = [ A ; r ]
```

kết quả là

```
A =
 1  2  3
 4  5  6
 7  8  9
10 11 12
```

Các ma trận nhỏ có thể được trích ra từ các ma trận lớn bằng cách dùng dấu hai chấm, : . Ví dụ, lệnh

$$\mathbf{A} = \mathbf{A}(1:3,:);$$

lấy ba dòng đầu và tất cả các cột của ma trận **A** hiện thời để đưa ma trận **A** về giá trị ban đầu. Sẽ biết thêm về dấu hai chấm sau này.

1.3. Câu lệnh và biến

MATLAB là ngôn ngữ biểu thức. Các biểu thức được đánh vào bởi người dùng, được thông dịch và ước lượng bởi hệ **MATLAB**. Các lệnh **MATLAB** thường có dạng:

$$\text{variable} = \text{expression}$$

hoặc đơn giản

$$\text{expression}$$

variable: tên biến,

expression: biểu thức.

Các biểu thức được cấu thành từ các toán tử và các ký tự đặc biệt khác, từ các hàm, và từ các tên biến. Việc ước lượng các biểu thức cho ra một ma trận, sau đó hiển thị trên màn hình và gán vào biến để sử dụng về sau. Nếu tên biến và dấu = bị bỏ qua thì một biến có tên là **ans**, viết tắt chữ "**answer**" (trả lời), được tự động tạo ra. Ví dụ, đánh vào

$$1900/81$$

cho ra **ans =**

$$23.4568$$

Một câu lệnh được kết thúc bình thường với ký tự sang dòng hay phím <Enter>. Tuy nhiên, nếu ký tự cuối cùng của câu lệnh là dấu chấm phẩy thì việc in ra kết quả được hủy, nhưng lệnh vẫn được thực hiện. Điều này là hữu ích trong các siêu tệp M-file (biết thêm sau này) và trong trường hợp kết quả đủ lớn không cần quan tâm từng số. Ví dụ, lệnh

$$\mathbf{p} = \text{conv}(\mathbf{r}, \mathbf{r});$$

tích chập các số trong **r** với chính chúng nhưng không hiển thị kết quả.

Nếu biểu thức quá phức tạp để câu lệnh không thể đặt gọn trên một dòng thì có thể dùng dấu tính lược (...) tiếp theo là ký tự sang dòng để biểu hiện câu lệnh được tiếp tục trên dòng tiếp theo. Ví dụ

$$s = 1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + 1/7 \dots \\ - 1/8 + 1/9 - 1/10 + 1/11 - 1/12;$$

tính tổng riêng của chuỗi điều hòa, gán tổng vào biến **s**, nhưng không in ra gì cả. Các ký tự trống quanh các dấu +, - là tùy chọn nhưng được đưa vào đây để dễ đọc.

Các **tên biến**, **tên hàm** được định dạng bằng một chữ viết, tiếp theo là số bất kỳ các chữ viết và chữ số (hoặc dấu nối). Chỉ có **19** ký tự đầu được nhớ.

MATLAB là ngôn ngữ nhạy cảm; nó thường phân biệt chữ hoa/chữ thường, bởi vậy **a** và **A** không phải là tên của cùng một biến. Tất cả các tên hàm phải là chữ thường; lệnh **inv(A)** sẽ lấy nghịch đảo của ma trận **A**, nhưng lệnh **INV(A)** tham chiếu đến một hàm không được định nghĩa: Tuy nhiên, lệnh **casesen** làm cho **MATLAB** không phân biệt chữ hoa/chữ thường. Trong chế độ này **INV(a)** là lấy ma trận đảo của nó.

1.4. Cách lấy thông tin vùng làm việc

Các lệnh trong các ví dụ cho đến bây giờ tạo ra các biến được lưu trong vùng làm việc của **MATLAB**. Thực hiện lệnh

who

liệt kê các biến trong vùng làm việc:

your variables are:

A ans p r s x

leaving 291636 bytes of memory free

ở đây trình bày **6** biến phát sinh bởi các ví dụ, kể cả biến **ans**. Để biết thêm chi tiết về kích thước của mỗi biến hiện thời, dùng lệnh **whos**, cũng với ví dụ, cho ra

Name	size	total	Complex
A	3 by 3	9	No
ans	1 by 1	1	No
p	1 by 5	5	No
r	1 by 3	3	No
s	1 by 1	1	No
x	1 by 5	5	No

**Grand total is (24*8) = 192 bytes,
leaving 291636 bytes of memory free.**

Mỗi phần tử của ma trận thực đòi hỏi **8** byte bộ nhớ, bởi vậy ma trận **A** cấp **3** dùng **72** byte và tất cả các biến dùng tổng cộng **192** byte. Tổng số không gian bộ nhớ tự do còn lại phụ thuộc vào từng loại máy khác nhau.

Biến **ans** cùng với một biến không liệt kê **eps** có ý nghĩa đặc biệt với **MATLAB**. Chúng là các biến cố định không thể xóa.

Biến **eps** (epsilon) dùng để xác định những giá trị gần kỳ dị (suy biến) và hạng ma trận. Giá trị khởi tạo của nó là khoảng cách từ **1.0** đến số thập phân lớn nhất tiếp theo. Đối với kỹ thuật số học **IEEE** (Institute of Electrical and Electronic Engineers) dùng trên các máy cá nhân và các máy trạm, thì

$$\text{eps} = 2^{-52}$$

khoảng 2.22×10^{-16} . **eps** có thể được đặt lại với giá trị khác, kể cả giá trị **0**.

1.5. Số và biểu thức số

Các số dùng ký pháp thập phân qui ước với dấu chấm và dấu trừ đứng trước là tùy chọn. Có thể đưa vào cuối dạng khoa học (lũy thừa 10). Sau đây là vài ví dụ về các số hợp pháp:

3	-99	0.0001
9.6397238	1.6040E-10	6.022252e23

Trên các máy dùng kỹ thuật số học chấm động **IEEE** thì độ chính xác tương đối của các số là **eps**, khoảng **16** chữ số có nghĩa. Miền giá trị khoảng **10⁻³⁰⁸** đến **10³⁰⁸**.

Các biểu thức có thể được tạo ra bằng cách dùng các phép toán số học thông thường và các qui tắc ưu tiên:

+	cộng
-	trừ
*	nhân
/	chia phải
\	chia trái
^	lũy thừa

Các phép toán trên ma trận để cho tiện có hai ký hiệu cho phép chia. Các biểu thức vô hướng **1/4** và **4\1** có cùng giá trị số, chính là **0.25**. Các cặp ngoặc đơn được dùng theo cách thông thường để xen vào việc ưu tiên của các phép toán số học.

Hầu hết các hàm toán sơ cấp thông thường trên các tính toán khoa học là các *hàm cài sẵn* của **MATLAB**, như **abs**, **sqrt**, **log**, và **sin**, ... Có thể thêm vào các hàm một cách dễ dàng với các siêu tệp M-file. Phần sau có một danh sách khá đầy đủ các hàm.

Một số các *hàm cài sẵn* đơn giản trả về các giá trị đặc biệt thường dùng. Hàm **pi** trả về số π , chương trình tính trước, đó là **4*atan(1)**. Một cách gọi khác để phát sinh số π là

$$\text{imag}(\log(-1))$$

Hàm **inf**, viết tắt chữ **infinity** (vô định), được thấy trên rất ít hệ tính toán hoặc ngôn ngữ lập trình. Trên một số máy, nó được tạo ra bởi kỹ thuật số học **IEEE** cài trong bộ đồng xử lý toán học (*coprocessor*). Trên các máy khác, phần mềm chấm động được đưa vào để mô phỏng đồng xử lý toán học. Một cách để phát sinh giá trị trả về bởi hàm **inf** là

$$s = 1/0$$

kết quả là $s =$

∞

Warning: Divide by zero.

Trên các máy với kỹ thuật số học **IEEE**, việc chia cho số không không dẫn đến điều kiện lỗi hoặc kết thúc hoạt động. Cho ra một thông báo khuyến cáo và một giá trị đặc biệt có thể xử lý trong việc tính toán sau đó.

Biến **NaN** là một số **IEEE** quan hệ với hàm **inf**, nhưng có các đặc tính khác. Nó là viết tắt chữ "**Not a Number**" (không phải là một số) và được cho ra bởi các việc tính toán như **inf/inf** hoặc **0/0**.

1.6. Số phức và ma trận phức

Số phức được dùng trong tất cả các phép toán và các hàm của **MATLAB**. Số phức được nhập bằng các hàm đặc biệt là **i** và **j**. Vài người có thể dùng

$$z = 3 + 4*i$$

trong khi người khác lại thích dùng

$$z = 3 + 4*j$$

Một ví dụ khác là

$$w = r*\exp(i*theta)$$

Có ít nhất hai cách thuận tiện để nhập ma trận phức. Chúng được minh họa bởi các lệnh

$$A = [1 \ 2; 3 \ 4] + i*[5 \ 6; 7 \ 8]$$

và

$$A = [1+5*i \ 2+6*i; 3+7*i \ 4+8*i]$$

cho ra cùng kết quả. Khi các số phức được nhập như các phân tử của ma trận bên trong cặp ngoặc vuông, thì điều quan trọng là tránh mọi khoảng trống, vì một biểu thức như $1 + 5*i$ với ký tự trống quanh dấu $+$ biểu hiện hai số riêng biệt. (Giống như thế cho số thực; một ký tự trống trước phần mũ trong $1.23 \ e-4$ gây ra lỗi).

Tên *hàm cài sẵn* có thể dùng như tên biến; trong trường hợp này hàm gốc trở nên không dùng được bên trong vùng làm việc hiện thời (hoặc hàm M-file cục bộ) cho đến khi biến bị xóa. Nếu dùng **i** và **j** là tên các biến, và đề lên các giá trị này, thì một đơn vị phức mới được phát sinh và sử dụng theo cách thông thường:

$$ii = \text{sqrt}(-1)$$

$$z = 3 + 4*ii$$

1.7. *Dạng thức xuất*

Kết quả của mọi lệnh gán của **MATLAB** được hiển thị trên màn hình, gán cho biến chỉ định hoặc cho **ans** nếu không cho biến. Dạng thức hiển thị số có thể điều khiển bằng lệnh **format**. Lệnh **format** chỉ ảnh hưởng đến cách hiển thị ma trận chứ không ảnh hưởng đến việc tính toán và lưu chúng (**MATLAB** thực hiện tất cả các tính toán theo độ chính xác kép "**double**").

Nếu tất cả các phân tử của ma trận đúng là số nguyên thì ma trận được hiển thị theo dạng không có phần thập phân. Ví dụ,

$$x = [-1 \ 0 \ 1]$$

kết quả luôn là $x =$

$$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

Nếu ít nhất một phần tử của ma trận không là số nguyên thì có một số cách có thể hiển thị. Dạng ngắn định, gọi là dạng **short**, trình bày khoảng **5** chữ số có nghĩa. Các dạng khác trình bày nhiều chữ số hơn hoặc dùng dạng khoa học. Ví dụ, giả sử

$$x = [4/3 \quad 1.2345e-6]$$

Các dạng thức, và kết quả xuất cho vector này, là:

Dạng thức short

1.3333 0.0000

Dạng thức short e

1.3333E+000 1.2345E-006

Dạng thức long

1.333333333333338 0.000001234500000

Dạng thức long e

1.333333333333338E+000 1.234500000000003E-006

Dạng thức hex

3FF555555555555 3EB4B6231AFBD271

Dạng thức +

++

Đối với các dạng **long** thì chữ số cuối cùng có thể xuất hiện không đúng, nhưng việc xuất ra đúng là một biểu hiện độ chính xác của số nhị phân lưu trong máy.

Với các dạng **short** và **long**, nếu phần tử lớn nhất của ma trận lớn hơn **1000** hoặc nhỏ hơn **0.001** thì một thừa số chung được áp dụng cho toàn bộ ma trận khi hiển thị nó. Ví dụ, lệnh

$$x = 1.e20 * x$$

nhân **x** cho **10²⁰** và kết quả hiển thị

x =

1.0E+020 *

1.3333 0.0000

Dạng thức **+** là cách cô đọng để hiển thị các ma trận lớn. Các ký hiệu **+**, **-** và ký tự trống được hiển thị cho các phần tử dương, âm và bằng không.

Lệnh cuối cùng, **format compact**, bỏ nhiều ký tự sang dòng xuất hiện giữa các hiển thị về ma trận và cho phép nhiều thông tin hiện trên màn hình.

1.8. Công cụ trợ giúp

Công cụ trợ giúp cung cấp thông tin trực tiếp về hầu hết các vấn đề của **MATLAB**. Để xem danh sách các vấn đề trợ giúp, đánh vào lệnh

help

Để lấy về một vấn đề chỉ định, đánh vào **help topic.**(*topic* là vấn đề cần trợ giúp). Ví dụ, lệnh

help eig

cung cấp thông tin về cách sử dụng hàm giá trị riêng,

help [

trình bày cách dùng các dấu ngoặc vuông để nhập ma trận, và

help help

là tham khảo chính nó, nhưng làm việc tốt đẹp.

1.9. Thoát và lưu vùng làm việc

Để thoát **MATLAB**, đánh vào lệnh **quit** hoặc **exit**. Việc kết thúc quá trình làm việc của **MATLAB** làm cho các biến trong vùng làm việc bị mất. Trước khi thoát, vùng làm việc có thể được lưu lại để dùng về sau bằng cách đánh vào lệnh

save

Lệnh này lưu tất cả các biến vào tệp có tên là **matlab.mat**. Khi gọi **MATLAB** lần sau, vùng làm việc có thể được phục hồi từ tệp **matlab.mat** bằng lệnh

load

Các lệnh **save** và **load** có thể dùng với các tên tệp khác, hoặc chỉ lưu các biến đã chọn. Lệnh **save temp** lưu các biến hiện thời vào tệp có tên là **temp.mat**. Lệnh

save temp X

chỉ lưu biến **X**, trong khi lệnh

save temp X Y Z

lưu **X**, **Y**, và **Z**.

Lệnh **load temp** lấy lại tất cả các biến từ tệp **temp.mat**. Các lệnh **load** và **save** cũng có thể dùng cho việc nhập và xuất các tệp dữ liệu dạng **ASCII**, xem phần tham khảo để biết thêm chi tiết.

1.10. Các hàm

Phần lớn tính năng của **MATLAB** nhận được từ tập hợp mở rộng của nó về các hàm. **MATLAB** có một số lớn các hàm, cho đến nay trên 500 hàm. Một số hàm là hàm nội tại hay hàm cài sẵn với chính trình xử lý **MATLAB**. Các hàm khác có thể ở thư viện các siêu tệp M-file bên ngoài cùng gói hàng của **MATLAB**(**MATLAB TOOLBOX**). Và một số được thêm vào bởi người dùng cho các trình ứng dụng đặc biệt.

Rõ ràng với người dùng thì một hàm có thể có hay không có trong trình **MATLAB** hoặc ở siêu tệp M-file. Đây là một mặt quan trọng của **MATLAB**; người dùng có thể tạo ra các hàm của riêng mình, và chúng hoạt động đúng như các hàm nội tại cài sẵn của **MATLAB** . Sẽ biết thêm về siêu tệp M-file trong phần sau.

Các phạm trù chung của các hàm toán học có thể dùng trong **MATLAB** gồm:

Toán sơ cấp
Các hàm đặc biệt
Ma trận sơ cấp
Ma trận đặc biệt
Tách và đặt thừa số ma trận
Phân tích dữ liệu
Đa thức
Giải phương trình vi phân
Phương trình phi tuyến và tối ưu phi tuyến
Tích phân số
Xử lý tín hiệu

Các phần sau sẽ giới thiệu các phạm trù khác nhau này về các hàm giải tích. Trong giáo trình này chúng tôi không đi vào chi tiết trên từng hàm; điều này được thực hiện bởi công cụ trợ giúp và trong phần tham khảo.

Cho đến bây giờ, chúng ta chỉ biết các hàm với một đối số nhập và một đối số xuất. Các hàm của **MATLAB** cũng có thể dùng với nhiều đối số. Ví dụ, lệnh

$$x = \text{sqrt}(\log(z))$$

trình bày cách dùng tổ hợp hai hàm đơn giản. Có các hàm của **MATLAB** dùng hai hoặc nhiều đối số nhập. Ví dụ,

$$\text{theta} = \text{atan2}(y, x)$$

Tất nhiên, mỗi đối số có thể là một biểu thức.

Một số hàm trả về hai hoặc nhiều đối số xuất. Các giá trị xuất được bọc quanh bởi cặp ngoặc vuông, [và], và cách nhau dấu phẩy:

$$[V,D] = \text{eig}(A)$$

$$[y,i] = \text{max}(X)$$

Hàm thứ nhất trả về hai ma trận, **V** và **D**, gồm vector riêng và các giá trị riêng tương ứng của ma trận **A**. Ví dụ thứ hai, dùng hàm **max**, trả về giá trị lớn nhất **y** và chỉ số **i** của giá trị lớn nhất trong vector **X**.

Các hàm cho phép nhiều đối số xuất có thể trả về ít đối số xuất hơn. Ví dụ, hàm **max** với một đối số xuất,

$$\text{max}(X)$$

trả về đúng giá trị lớn nhất.

Các đối số nhập hay đối số ở bên phải của một hàm không bao giờ được thay đổi. Các giá trị xuất, nếu có, của một hàm luôn trả về ở các đối số bên trái.

Chương 2. CÁC PHÉP TOÁN TRÊN MA TRẬN

Các phép toán trên ma trận là điều cơ bản của **MATLAB**; bất kỳ đâu có thể được, chúng biểu hiện như xuất hiện trên giấy, chỉ phụ thuộc vào dung lượng bộ nhớ của máy.

2.1. Chuyển vị ma trận

Ký tự đặc biệt là dấu nháy (') biểu hiện phép chuyển vị một ma trận. Các lệnh

```
A = [ 1 2 3; 4 5 6; 7 8 0 ]
```

```
B = A'
```

kết quả là

```
A =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 0
```

```
B =
```

```
1 4 7
```

```
2 5 8
```

```
3 6 0
```

và lệnh

```
x = [ -1 0 2 ]'
```

cho ra

```
x =
```

```
-1
```

```
0
```

```
2
```

Dấu nháy ' chuyển vị ma trận theo ý nghĩa hình thức; nếu **Z** là ma trận phức thì **Z'** là chuyển vị liên hợp của nó. Điều này đôi khi dẫn đến kết quả không như ý

muốn nếu dùng dữ liệu phức một cách bất cần. Đối với một chuyển vị không liên hợp thì dùng biểu thức \mathbf{Z}' hoặc hàm $\text{conj}(\mathbf{Z}')$.

2.2. Cộng và trừ ma trận

Cộng và trừ ma trận được biểu hiện bằng các ký hiệu $+$ và $-$. Các phép toán được định nghĩa cho các ma trận cùng cỡ. Ví dụ, với các ma trận trên, $\mathbf{A}+\mathbf{x}$ là không đúng, vì \mathbf{A} là ma trận vuông cấp 3 và \mathbf{x} là ma trận cỡ 3×1 . Tuy nhiên,

$$\mathbf{C} = \mathbf{A} + \mathbf{B}$$

là chấp nhận được, và kết quả là

$$\mathbf{C} = \begin{bmatrix} 2 & 6 & 10 \\ 6 & 10 & 14 \\ 10 & 14 & 0 \end{bmatrix}$$

Các phép cộng và trừ cũng được định nghĩa nếu một trong các toán hạng là đại lượng vô hướng, đó là ma trận cấp một. Trong trường hợp này, đại lượng vô hướng được cộng hoặc trừ vào tất cả các phần tử của toán hạng kia. Ví dụ

$$\mathbf{y} = \mathbf{x} - 1$$

cho ra

$$\mathbf{y} = \begin{bmatrix} -2 \\ -1 \\ 1 \end{bmatrix}$$

2.3. Nhân ma trận

Phép nhân ma trận được biểu hiện bởi ký hiệu $*$. Phép toán được định nghĩa cho các ma trận có kích thước bên trong bằng nhau, đó là $\mathbf{X}*\mathbf{Y}$ cho phép nếu số cột của ma trận \mathbf{X} bằng số hàng của ma trận \mathbf{Y} . Ví dụ, cả hai ma trận \mathbf{x} và \mathbf{y} ở trên có cỡ 3×1 , vì vậy biểu thức $\mathbf{x}*\mathbf{y}$ không được định nghĩa và kết quả là một thông báo lỗi. Tuy nhiên, vài phép nhân khác về vectơ được định nghĩa, và rất hữu ích. Thông dụng nhất là tích nội tại, cũng được gọi là tích điểm hay tích vô hướng. Đây là

kết quả là

$$\mathbf{x}' * \mathbf{y}$$

$$\mathbf{ans} =$$

$$4$$

Tất nhiên, $\mathbf{y}' * \mathbf{x}$ cho cùng kết quả. Có hai tích ngoại lai, chúng là chuyển vị của nhau.

$$\mathbf{x} * \mathbf{y}' =$$

$$\begin{bmatrix} 2 & 1 & -1 \\ 0 & 0 & 0 \\ -4 & -2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -4 & -2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} -4 & -2 & 2 \\ 0 & 0 & 0 \\ 2 & 2 & -2 \end{bmatrix}$$

$$\mathbf{y} * \mathbf{x}' =$$

$$\begin{bmatrix} 2 & 0 & -4 \\ 1 & 0 & -2 \\ -1 & 0 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & -2 \\ -1 & 0 & 2 \\ 2 & 2 & -2 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 2 \\ 2 & 2 & -2 \\ -2 & -2 & 2 \end{bmatrix}$$

Phép nhân trên từng phần tử sẽ được mô tả trong phần sau. (**MATLAB** không cung cấp đặc biệt cho việc tính toán vector qua các phép nhân. Tuy nhiên, người nào cần thì dễ dàng viết một siêu tệp M-file để tính toán chúng.)

Các phép nhân ma trận với vector là các trường hợp đặc biệt của nhân tổng quát ma trận với ma trận. Với ví dụ ma trận \mathbf{A} và vector \mathbf{x} thì

$$\mathbf{b} = \mathbf{A} * \mathbf{x}$$

là được phép và kết quả xuất là

$$\mathbf{b} =$$

$$5$$

$$8$$

$$-7$$

Một cách tự nhiên, một đại lượng vô hướng có thể nhân hoặc bị nhân với ma trận bất kỳ.

```

pi*x
ans =

    -3.1416

     0.0000

     6.2432

```

2.4. Chia ma trận

Trong **MATLAB** có hai ký hiệu "**chia ma trận**", \backslash và $/$. Nếu **A** là ma trận không suy biến, thì $\mathbf{A} \backslash \mathbf{B}$ và \mathbf{B}/\mathbf{A} tương ứng hình thức với nhân trái và nhân phải của **B** cho nghịch đảo của **A**, đó là $\mathbf{inv}(\mathbf{A}) * \mathbf{B}$ và $\mathbf{B} * \mathbf{inv}(\mathbf{A})$, nhưng kết quả nhận được trực tiếp chứ không tính toán qua phép nghịch đảo. Nói chung,

$\mathbf{X} = \mathbf{A} \backslash \mathbf{B}$ là lời giải phương trình $\mathbf{A} * \mathbf{X} = \mathbf{B}$

$\mathbf{X} = \mathbf{B}/\mathbf{A}$ là lời giải phương trình $\mathbf{X} * \mathbf{A} = \mathbf{B}$

Phép chia trái, $\mathbf{A} \backslash \mathbf{B}$, được định nghĩa với **B** có cùng số dòng với **A**. Nếu **A** là ma trận vuông, thì nó được phân tích bằng phép khử **Gauss**. Các nhân tử được dùng để giải các phương trình $\mathbf{A} * \mathbf{X}(:,j) = \mathbf{B}(:,j)$, ở đây $\mathbf{B}(:,j)$ biểu hiện cột thứ *j* của **B**. Kết quả là ma trận **X** có cùng cỡ với **B**. Nếu **A** suy biến (tùy theo ước lượng điều kiện **LINPACK** là **RCOND**) thì một thông báo lỗi được hiển thị.

Nếu **A** là ma trận vuông thì nó được phân tích bằng phương pháp trực giao **Householder** với việc định trục xoay về cột. các nhân tử được dùng để giải các phương trình xác định dưới hoặc trên theo phương pháp bình phương bé nhất. Kết quả là một ma trận **X** cỡ $m \times n$, ở đây *m* là số cột của **A** và *n* là số cột của **B**. Mỗi cột của **X** có nhiều nhất *k* thành phần khác không, với *k* là hạng thực thụ của **A**.

Phép chia phải, \mathbf{B}/\mathbf{A} , được định nghĩa theo dạng chia trái là $\mathbf{B}/\mathbf{A} = (\mathbf{A}' \backslash \mathbf{B}')'$.

Ví dụ, khi vectơ **b** được tính là $\mathbf{A} * \mathbf{x}$ thì lệnh

```
z = A\b
```

có kết quả

```

z =

    -1
     0
     2

```

Đôi lúc dùng \ và / để tính các lời giải hệ phương trình xác định dưới hoặc trên bằng phương pháp bình phương bé nhất đưa đến nhiều điều đáng ngạc nhiên. Đó là khả năng "**chia**" một vector cho vector khác. Ví dụ, với các vector \mathbf{x} và \mathbf{y} ở trên thì

$$\mathbf{s} = \mathbf{x} \backslash \mathbf{y}$$

cho ra

$$\mathbf{s} =$$

$$\mathbf{0.8000}$$

Đây là vì $\mathbf{s} = \mathbf{0.8}$ là giá trị vô hướng giải được từ phương trình $\mathbf{x}\mathbf{s} = \mathbf{y}$ theo phương pháp bình phương bé nhất. Chúng tôi đề nghị bạn đọc giải thích tại sao

$$\mathbf{S} = \mathbf{y} / \mathbf{x}$$

cho ra

$$\mathbf{S} =$$

$$\mathbf{0.0000 \quad 0.0000 \quad -1.0000}$$

$$\mathbf{0.0000 \quad 0.0000 \quad -0.5000}$$

$$\mathbf{0.0000 \quad 0.0000 \quad 0.5000}$$

2.5. Lũy thừa ma trận

Biểu thức $\mathbf{A}^{\mathbf{p}}$ nâng \mathbf{A} lên lũy thừa bậc \mathbf{p} và được định nghĩa nếu \mathbf{A} là ma trận vuông và \mathbf{p} là đại lượng vô hướng. Nếu \mathbf{p} là số nguyên lớn hơn 1 thì phép lũy thừa được tính bằng cách nhân lặp. Đối với các giá trị khác của \mathbf{p} thì việc tính toán gồm các giá trị riêng và các vector riêng, vì vậy nếu $[\mathbf{V}, \mathbf{D}] = \text{eig}(\mathbf{A})$ thì

$$\mathbf{A}^{\mathbf{p}} = \mathbf{V} * \mathbf{D}.^{\mathbf{p}} / \mathbf{V}$$

Nếu \mathbf{P} là một ma trận, và \mathbf{a} là đại lượng vô hướng thì $\mathbf{a}^{\mathbf{P}}$ nâng \mathbf{a} lên lũy thừa \mathbf{P} bằng cách dùng các giá trị riêng và các vector riêng. $\mathbf{X}^{\mathbf{P}}$, với \mathbf{X} và \mathbf{P} đều là ma trận, là một lỗi.

2.6. Các hàm sơ cấp về ma trận

Trong **MATLAB**, các biểu thức như **exp(A)** và **sqrt(A)** được xem như các phép toán về mảng, xác định trên từng phần tử của \mathbf{A} . **MATLAB** cũng có thể tính các hàm siêu việt về ma trận, như hàm mũ và hàm logarit. Các hàm đặc biệt này

chỉ được định nghĩa cho các ma trận vuông, tính toán khá khó và tốn thời gian, và đôi lúc có các tính chất toán học khá hấp dẫn.

Một hàm toán học siêu việt được thông dịch là hàm về ma trận nếu có chữ "m" nối thêm vào cuối tên hàm, như **expm(A)** và **sqrtn(A)**. Theo trọn bộ của **MATLAB** thì ba hàm sau đây được định nghĩa:

Hàm siêu việt trên ma trận	
expm	hàm mũ
logm	hàm loga
sqrtn	hàm căn bậc hai

Tuy nhiên, danh sách có thể được mở rộng bằng cách thêm vào các siêu tệp M-file, hoặc dùng lệnh **funm**. Xem các siêu tệp M-file **sqrtn**, **logm**, và **funm** trong **MATLAB TOOLBOX**, và **expm** và **funm** trong phần tham khảo.

Các hàm sơ cấp khác về ma trận gồm

Hàm sơ cấp trên ma trận	
poly	tính đa thức đặc trưng
det	tính định thức
trace	tìm vết ma trận
kron	tích tenxơ Kronecker

Xem phần tham khảo để biết thêm chi tiết.

Chương 3. CÁC PHÉP TOÁN TRÊN MẢNG

Chúng tôi dùng từ *phép toán trên mảng* để nói về các phép toán số học trên từng phần tử, thay cho các phép toán đại số tuyến tính thông thường về ma trận biểu hiện bởi các ký hiệu $*$ / \backslash $^$ $'$. Đưa vào trước phép toán dấu chấm để biểu hiện phép toán trên mảng hay phép toán trên từng phần tử.

3.1. Cộng và trừ trên mảng

Đối với các phép cộng và trừ thì phép toán trên mảng và trên ma trận là giống nhau, vì vậy $+$ và $-$ có thể được xem là các phép toán hoặc là trên ma trận, hoặc là trên mảng.

3.2. Nhân và chia trên mảng

Phép nhân trên mảng hoặc nhân từng phần tử được biểu hiện bằng $.*$. Nếu A và B cùng kích thước thì $A.*B$ biểu hiện mảng mà các phần tử của nó đơn giản là tích của từng cặp phần tử của A và B. Ví dụ, nếu

$$x = [1 \ 2 \ 3]; \ y = [4 \ 5 \ 6];$$

thì

$$z = x .* y$$

kết quả là

$$z =$$

$$4 \quad 10 \quad 18$$

Các biểu thức $A ./ B$ và $A . \backslash B$ cho ra thương của từng cặp phần tử. Vì vậy,

$$z = x ./ y$$

kết quả là

$$z =$$

$$4.0000 \quad 2.5000 \quad 2.0000$$

3.3. Lũy thừa trên mảng

Lũy thừa từng phần tử biểu hiện bởi $.^$. Sau đây là một số ví dụ, dùng các vector x và y ở trên. Đánh vào

$$z = x.^y$$

kết quả là

$$z =$$

$$\begin{bmatrix} 1 & 32 & 729 \end{bmatrix}$$

Phần mũ có thể là một đại lượng vô hướng.

$$z = 2.^{[x \ y]}$$

$$z =$$

$$\begin{bmatrix} 2 & 4 & 8 & 16 & 32 & 64 \end{bmatrix}$$

Ví dụ cuối cùng minh họa cho một trong các đặc tính hấp dẫn về cú pháp của **MATLAB**. Mặc dù khó thấy, nhưng khoảng trống giữa chữ số **2** và dấu chấm là quan trọng. Nếu không có thì dấu chấm sẽ được thông dịch là một dấu chấm thập phân quan hệ với số **2**. Rồi **MATLAB** chỉ xem dấu mũ đứng riêng và tính lũy thừa ma trận, trong trường hợp này kết quả là một thông báo lỗi vì ma trận mũ không vuông. Xen vào cặp ngoặc đơn để thực hiện cấp độ ưu tiên toán tử.

3.4. Phép toán quan hệ

Có 6 phép toán quan hệ để so sánh hai ma trận cùng cỡ.

Phép toán quan hệ	
<	Nhỏ hơn
<=	Nhỏ hơn hoặc bằng
>	Lớn hơn
>=	Lớn hơn hoặc bằng
==	Bằng
~=	Khác

Phép so sánh được thực hiện giữa các cặp phần tử tương ứng; kết quả là một ma trận gồm các số **1** và **0**, với **1** biểu hiện cho giá trị *đúng* (**TRUE**) và **0** biểu hiện cho giá trị *sai* (**FALSE**).

Ví dụ

$$2+2 \sim 4$$

đơn giản là **0**.

Các phép quan hệ có thể trình bày mẫu của các phần tử của ma trận thỏa mãn các điều kiện khác nhau. Ví dụ, sau đây là ma phương cấp 6.

A = magic(6)

A =

35	1	6	26	19	24
3	32	7	21	23	25
31	9	2	22	27	20
8	28	33	17	10	15
30	5	34	12	14	16
4	36	29	13	18	11

Ma phương cấp n là một ma trận cấp n được xây dựng từ các số nguyên từ 1 đến n^2 với tổng các dòng và các cột bằng nhau. Nếu nhìn vào ma trận đủ lớn thì có thể lưu ý thấy các phần tử bội 3 nằm trên đường chéo thứ ba. Để hiển thị điều kỳ lạ này, đánh vào

P = (rem(A,3) == 0)

Dấu = kép là toán tử kiểm tra bằng, **rem(A,3)** là ma trận gồm các số dư, 0 được mở rộng thành ma trận không, và **P** trở thành ma trận gồm các số 1 và 0.

P =

0	0	1	0	0	1
1	0	0	1	0	0
0	1	0	0	1	0
0	0	1	0	0	1
1	0	0	1	0	0
0	1	0	0	1	0

Để thấy mẫu nhỏ hơn và rõ ràng, lệnh **format +** in ma trận ở dạng cô đọng, với dấu + cho số dương, dấu - cho số âm, và ký tự trống cho số 0.

format +

P

	+				+
+					+
	+				+
		+			+
+					+
	+				+

Hàm **find** là hữu ích với các phép toán quan hệ, đó là tìm các số khác không trong ma trận **0-1**, và các phần tử dữ liệu thỏa mãn điều kiện quan hệ nào đó. Ví dụ, nếu **Y** là một vector thì hàm **find(Y < 3.0)** trả về một vector chứa các chỉ số của các phần tử trong **Y** nhỏ hơn **3.0**.

Các lệnh

```
i = find(Y > 3.0);
```

```
Y(i) = 10*ones(i);
```

thay tất cả các phần tử trong **Y** lớn hơn **3.0** với giá trị **10.0**. Nó làm việc với ngay cả **Y** là một ma trận, vì ma trận có thể được tham chiếu như mảng các vector cột.

Biểu thức quan hệ **X == NaN** luôn cho ra **NaN**, tùy theo các chỉ định số học **IEEE**, mọi phép toán trên **NaN** cho ra **NaN**. Nhưng đôi lúc cần kiểm tra các giá trị **NaN**. Vì vậy, hàm **isnan(X)** được cung cấp để trả về **1** cho các phần tử **NaN** của **X**, và **0** với các phần tử khác.

Cũng có ích là hàm **finite(x)** trả về **1** cho $-\infty < x < \infty$.

3.5. Phép toán logic

Có **3** phép toán logic làm việc với từng phần tử và thường dùng với các ma trận **0-1**.

Phép toán quan hệ	
&	Phép và
 	Phép hoặc
~	Phép phủ định

Các phép **&** và **|** so sánh hai đại lượng vô hướng, hoặc hai ma trận cùng cỡ. Đối với các ma trận thì chúng làm việc trên từng phần tử; nếu **A** và **B** là các ma trận **0-1** thì biểu thức **A & B** là một ma trận **0-1** khác biểu hiện logic **AND** của các phần tử tương ứng của **A** và **B**. Các phép toán logic xem mọi số khác **0** là đúng (**TRUE**). Chúng trả về **1** cho **TRUE** và **0** cho **FALSE**.

Phép **NOT**, hoặc bù logic, là toán tử đơn hạng. Biểu thức **~A** trả về các số **0** cho các phần tử khác **0** của **A** và **1** cho các phần tử **0**. Do đó hai biểu thức

P | (~P)

P & (~P)

trả về tất cả **1** và tất cả **0** tương ứng.

Các hàm **any** và **all** là hữu ích trong việc liên kết với các phép toán logic. Nếu **x** là vector **0-1** thì **any(x)** trả về **1** nếu phần tử bất kỳ của **x** khác không, và ngược lại trả về **0**.

Hàm **all(x)** trả về **1** chỉ nếu tất cả các phần tử của **x** khác không. Các hàm này đặc biệt hữu ích trong câu lệnh **if**,

if all(A < .5)

thực hiện các lệnh

end

Một lệnh **if** muốn trả lời cho một điều kiện đơn giản thì không phải là một vector có thể nhầm lẫn.

Đối với các đối số là ma trận thì các hàm **any** và **all** làm việc trên từng cột và trả về một vector dòng với kết quả của mỗi cột. áp dụng hàm hai lần, như **any(any(A))** luôn thu gọn ma trận về một điều kiện vô hướng.

Sau đây là bảng tóm tắt các hàm quan hệ và logic của **MATLAB**:

Hàm quan hệ và logic	
any	Điều kiện logic
all	Điều kiện logic
find	Tìm chỉ số của các điều kiện logic
exist	Kiểm tra nếu các biến tồn tại
isnan	Dò tìm các giá trị NaN
finite	Dò tìm các giá trị vô định
isempty	Dò tìm các ma trận rỗng
isstr	Dò tìm các biến xâu chữ
strcmp	So sánh các biến xâu chữ

3.6. Các hàm toán sơ cấp

Một tập hợp các hàm toán sơ cấp được áp dụng vào mảng trên cơ sở từng phần tử. Ví dụ,

A = [1 2 3; 4 5 6]

B = fix(pi*A)

C = cos(pi*B)

cho ra **A =**

1 2 3

4 5 6

B =

3 6 9

12 15 18

C =

-1 1 -1

1 -1 1

Các hàm có thể sử dụng gồm các hàm lượng giác và các hàm sơ cấp thông dụng:

Hàm lượng giác	
sin	Hàm sin
cos	Hàm cosin
tan	Hàm tang
asin	Hàm arcsin
acos	Hàm arccos
atan	Hàm arctang
atan2	Hàm arctang...
sinh	Hàm sin hyperbol
cosh	Hàm cosin hyperbol
tanh	Hàm tang hyperbol
asinh	Hàm arcsin hyperbol
acosh	Hàm arccos hyperbol
atanh	Hàm arctang hyperbol

Hàm toán sơ cấp	
abs	Trị tuyệt đối hoặc argument số phức
angle	Góc pha
sqrt	Căn bậc hai
real	Phần thực
imag	Phần ảo
conj	Liên hợp của số phức
round	Làm tròn về số nguyên gần nhất
fix	Làm tròn về phía số 0
floor	Làm tròn về phía $-\infty$
ceil	Làm tròn về phía ∞
sign	Hàm dấu
rem	Phần dư hoặc môđun
exp	Mũ cơ số e
log	Logarit tự nhiên
log10	Logarit cơ số 10

3.7. Các hàm toán học đặc biệt

Một số hàm đặc biệt cung cấp nhiều khả năng nâng cao:

Hàm đặc biệt	
bessel	Hàm Bessel
gamma	Hàm gamma và gamma bù
rat	Hàm xấp xỉ
erf	Hàm lỗi
invert	Hàm đảo lỗi
ellipk	Hàm tích phân bù elliptic loại I
ellipj	Hàm elliptic Jacôbiên

Giống như các hàm sơ cấp, chúng thực hiện trên từng phần tử khi nhập ma trận. Xem phần tham khảo để biết thêm thông tin.

Chương 4. THAO TÁC TRÊN VECTƠ VÀ MA TRẬN

Các công cụ về mô tả chỉ số của **MATLAB** cho phép thực hiện về dòng, về cột, về từng phần tử riêng biệt và từng phần của ma trận. Tâm điểm của việc mô tả chỉ số là vectơ, được phát sinh bằng cách dùng "Ký pháp Hai chấm". Vectơ và việc mô tả chỉ số là các thao tác hay dùng trong **MATLAB** và làm cho nó thực hiện các thao tác trên dữ liệu phức tạp khá hiệu lực.

4.1. Cách phát sinh vectơ

Dấu hai chấm, :, là ký tự quan trọng trong **MATLAB**. Lệnh

```
x = 1:5
```

phát sinh ra một vectơ dòng chứa các số từ 1 đến 5 theo chiều tăng đơn vị. Nó cho ra

```
x =  
1 2 3 4 5
```

Các cách tăng khác có thể dùng được.

```
y = 0:pi/4:pi
```

kết quả là

```
y =  
0.0000 0.7854 1.5708 2.3562 3.1416
```

Có thể thay đổi theo đơn vị âm.

```
z = 6:-1:1
```

cho ra

```
z =  
6 5 4 3 2 1
```

Ký pháp hai chấm cho phép phát sinh các bảng một cách dễ dàng. Để lấy một bảng sắp xếp theo chiều đứng thì chuyển vị vectơ dòng nhận được từ ký pháp hai chấm, tính toán cột giá trị, rồi định dạng ma trận từ hai cột. Ví dụ

```
x = (0.0 : 0.2 : 3.0)';
```

```
y = exp(-x) .* sin(x);
```

```
[x y]
```

cho ra

```
ans =
```

```
0.0000 0.0000
```

```
0.2000 0.1627
```

```
0.4000 0.2610
```

```
0.6000 0.3099
```

```
0.8000 0.3223
```

```
1.0000 0.3096
```

```
1.2000 0.2807
```

```
1.4000 0.2430
```

```
1.6000 0.2018
```

```
1.8000 0.1610
```

```
2.0000 0.1231
```

```
2.2000 0.0896
```

```
2.4000 0.0613
```

```
2.6000 0.0383
```

```
2.8000 0.0204
```

```
3.0000 0.0070
```

Các hàm phát sinh vectơ khác gồm **linspace**, cho phép số thực tốt hơn là cách tăng như đã chỉ định,

```
k = linspace(-pi,pi,4)
```

```
k =
```

```
-3.1416 -1.0472 1.0472 3.1416
```

và hàm **logspace** phát sinh **vector logarit** đồng đều.

4.2. Mô tả chỉ số

Các phần tử riêng biệt của ma trận có thể được tham chiếu bằng cách đưa vào chỉ số trong cặp ngoặc đơn. Một biểu thức dùng làm chỉ số được làm tròn thành số nguyên gần nhất. Ví dụ, cho ma trận **A** :

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

thì lệnh

$$\mathbf{A}(3,3) = \mathbf{A}(1,3) + \mathbf{A}(3,1)$$

kết quả là

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{bmatrix}$$

Một chỉ số có thể là một vectơ.

Nếu **X** và **V** là các vectơ thì **X(V)** là **[X(V(1), X(V(2)),..., X(V(n))]**. Đối với các ma trận chỉ số vectơ cho phép truy cập đến các ma trận con liên tục và không liên tục. Ví dụ, giả sử **A** là ma trận cấp **10**. Thì

$$\mathbf{A}(1:5,3)$$

là ma trận con cỡ **5x1**, hay là vectơ cột gồm **5** phần tử đầu trên cột thứ **3** của ma trận **A**. Tương tự,

$$\mathbf{A}(1:5, 7:10)$$

là ma trận con cỡ **5x4** gồm các phần tử từ **5** dòng đầu và **4** cột cuối.

Dùng chính dấu hai chấm đặt tại vị trí mô tả chỉ số biểu hiện tất cả các dòng hoặc các cột tương ứng. Ví dụ,

$$\mathbf{A}(:,3)$$

là cột thứ 3 và

$$\mathbf{A}(1:5,:)$$

là 5 cột đầu.

Hiệu lực khá tinh vi là nhận được từ việc tham chiếu ma trận con ở cả hai phía của lệnh gán. Ví dụ,

$$\mathbf{A}(:,[3 \ 5 \ 10]) = \mathbf{B}(:,1:3)$$

thay các cột thứ 3, 5 và 10 của **A** với 3 cột đầu của **B**.

Nói chung, nếu **v** và **w** là các vectơ với các thành phần nguyên thì

$$\mathbf{A}(\mathbf{v},\mathbf{w})$$

là ma trận nhận được bằng cách lấy các phần tử của **A** với chỉ số dòng trong **v** và chỉ số cột trong **w**. Vì vậy

$$\mathbf{A}(:,\mathbf{n}:-1:1)$$

đảo lại các cột của **A** và

$$\mathbf{v} = 2:2:\mathbf{n};$$

$$\mathbf{w} = [3 \ 1 \ 4 \ 1 \ 6];$$

$$\mathbf{A}(\mathbf{v},\mathbf{w})$$

là hợp pháp, nhưng có lẽ đáng ngờ.

Một đặc điểm nữa cũng khá hữu ích là **A(:)**. ở vế phải câu lệnh gán, **A(:)** biểu hiện tất cả các phần tử của **A** được giống thành một vectơ cột. Ví dụ

$$\mathbf{A} = [1 \ 2; 3 \ 4; 5 \ 6]$$

$$\mathbf{b} = \mathbf{A}(:)$$

kết quả là

$$\mathbf{A} =$$

$$1 \ 2$$

```

3  4
5  6

b =

1
2
3
4
5
6

```

Ở vế trái câu lệnh gán, **A(:)** có thể được dùng để đổi lại cỡ ma trận. Để làm điều này, trước hết **A** phải có. Sau đó **A(:)** biểu hiện ma trận cùng cỡ với **A**, nhưng với nội dung mới lấy bên vế phải. Ví dụ, **A** ở trên cỡ **3X2**, vì vậy

```
A(:) = 11:16
```

đổi 6 phần tử của vector dòng thành ma trận cỡ **3X2**,

```

A =

11  14
12  15
13  16

```

4.3. Mô tả chỉ số bằng vector 0-1

Có thể dùng các vector **0-1**, thường được tạo ra bằng các phép toán quan hệ, để tham chiếu các ma trận con. Giả sử **A** là ma trận cỡ **m x n** và **L** là vector **m** chiều gồm các phần tử **0** và **1**. Thì **A(L,:)** chỉ định các dòng của **A** ứng với các phần tử **1** của **L**.

Sau đây là các cách trích ra, các phần tử lớn hơn **3** lần độ lệch chuẩn bị xóa trong vector:

```
x = x(x<=3*std(x));
```

Tương tự,

```
L = X(:,3)>100;
```

```
X = X(L,:);
```


thay **X** với các dòng có phần tử ở cột thứ **3** lớn hơn **100**.

4.4. Ma trận rỗng

Lệnh

$$\mathbf{x} = []$$

gán ma trận cỡ **0x0** cho **x**. Sau đó dùng ma trận này sẽ không dẫn đến điều kiện lỗi; mà truyền cho ma trận rỗng. Lệnh này khác với lệnh

$$\text{clear } \mathbf{x}$$

Xóa **x** từ danh sách các biến hiện thời. Các ma trận rỗng có trong vùng làm việc; chúng có đúng cỡ **0x0**. Hàm **exist** có thể dùng để kiểm tra sự tồn tại của một ma trận (hoặc một tập cho ma trận), trong khi đó hàm **isempty** kiểm tra ma trận rỗng.

Có thể phát sinh các vectơ rỗng. Nếu **n<1** thì **1:n** không chứa phần tử nào cả và do đó

$$\mathbf{x} = 1:\mathbf{n}$$

là một cách việc tạo một vectơ **x** rỗng.

Quan trọng hơn nữa là một cách có hiệu lực để xóa các dòng, các cột của một ma trận là gán chúng cho một ma trận rỗng. Ví dụ

$$\mathbf{A}(:, [2, 4]) = []$$

xóa cột **2** và **4** của ma trận **X**.

Chắc chắn các hàm ma trận sẽ trả về các giá trị hợp lý về mặt toán học nếu ma trận rỗng. Chúng là các hàm **det**, **cond**, **prod**, **sum**, và một số hàm khác. Ví dụ, các hàm **prod**, **det**, và **sum** trả về **1**, **1**, và **0** tương ứng khi ma trận đối là rỗng.

Hơn nữa chúng ta đã biết không có đại số về các ma trận rỗng. Chúng tôi không chắc rằng chúng tôi sẽ thực hiện điều đó một cách đúng đắn, nhưng chúng tôi đã tìm thấy nhiều điều hữu ích.

4.5. Ma trận đặc biệt

Tập hợp các hàm phát sinh các ma trận đặc biệt để đưa vào trong đại số tuyến tính và xử lý tín hiệu.

Các ma trận đặc biệt	
companion	Ma trận Liên hợp
diag	Ma trận Chéo
gallery	Ma trận riêng
hadamard	Ma trận Hadamard
hankel	Ma trận Hankel
hilb	Ma trận Hilbert
invhilb	Ma trận Hilbert đảo
magic	Ma phương
pascal	Tam giác Pascal
toeplitz	Ma trận Toeplitz
vander	Ma trận Vandermonde

Ví dụ, phát sinh một ma trận liên hợp với đa thức $x^3 - 7x + 6$.

p = [1 0 -7 6]

a = companion(p)

a =

0 7 -6

1 0 0

0 1 0

Các giá trị riêng của **a** là nghiệm của đa thức.

-3.0000

2.0000

1.0000

Một ma trận **Toeplitz** không đồng bộ về đường chéo là

c = [1 2 3 4 5];

r = [1.5 2.5 3.5 4.5 5.5];

t = toeplitz(c,r)

t =

1.000	2.500	3.500	4.500	5.500
2.000	1.000	2.500	3.500	4.500
3.000	2.000	1.000	2.500	3.500
4.000	3.000	2.000	1.000	2.500
5.000	4.000	3.000	2.000	1.000

Các hàm khác phát sinh các ma trận tiện ích ít quan tâm nhưng hữu ích hơn.

Các ma trận tiện ích	
zeros	Ma trận zero
ones	Ma trận một
rand	Ma trận phần tử ngẫu nhiên
eye	Ma trận đơn vị
linspace	Vectơ không gian tuyến tính
logspace	Vectơ không gian loga
meshdom	Phạm vi để vẽ lưới

Trong các hàm này có hàm **eye(A)** trả về ma trận đơn vị cùng cỡ với **A**. Nên dùng tên để nhớ vì **I** và **i** thường dùng như các chỉ số hay đơn vị ảo **sqrt(-1)**.

Các hàm **zeros** và **ones** phát sinh các ma trận hằng có kích cỡ khác nhau, và hàm **rand** để phát sinh các ma trận phân bố đồng bộ hoặc bình thường các phần tử ngẫu nhiên. Ví dụ, để phát sinh ma trận cỡ **4x3**

A = rand(4,3)

A =

0.2113 0.8096 0.4832

0.0824 0.8474 0.6135

0.7599 0.4524 0.2749

0.0087 0.8075 0.8807

4.6. Cách tạo ra ma trận lớn

Các ma trận lớn có thể được tạo ra từ các ma trận nhỏ bằng cách dùng cặp ngoặc vuông bao quanh các ma trận nhỏ. Ví dụ,

$$C = [A \text{ eye}(4); \text{ones}(A) \ A^2]$$

tạo ra ma trận lớn với giả thiết **A** có 4 dòng. Các ma trận nhỏ hơn trong kiểu này của cách xây dựng phải cùng cỡ hoặc kết quả là một thông báo lỗi.

4.7. Thực hiện trên ma trận

Các hàm sẽ quay, đổi hàng-cột, thay đổi kích thước, hoặc trích ra các phần của ma trận.

Thao tác trên ma trận	
rot90	Quay ma trận
fliplr	Đổi cột ma trận
flipud	Đổi cột ma trận
diag	Trích hoặc tạo ra đường chéo
tril	Phần tam giác dưới
triu	Phần tam giác trên
reshape	Đặt lại kích thước
.'	Chuyển vị
:	Sắp xếp tổng quát

Ví dụ, để đặt lại kích thước ma trận cỡ **3x4** thành ma trận **2x6**:

a =

1 4 7 10

2 5 8 11

3 6 9 12

b = reshape(a,2,3)

b =

1 3 5 7 9 11

2 4 6 8 10 12

Ba hàm **diag**, **triu**, và **tril** cung cấp truy cập đến đường chéo, tam giác trên, và tam giác dưới của ma trận. Ví dụ,

tril(rand(4,3))

```
cho ra      ans =  
  
      0.2113      0      0  
      0.0824      0.8474      0  
      0.7599      0.4524      0.2749  
      0.0087      0.8075      0.8807
```

Cũng rất hữu ích là các hàm **size** và **length**. Hàm **size** trả về vector **2** chiều chứa số dòng và số cột của một ma trận. Nếu biến là một vectơ thì **length** trả về số chiều của vectơ, hoặc **max(size(V))**.

Chương 5. THAO TÁC TRÊN VECTO VÀ MA TRẬN

Phần này giới thiệu về cách phân tích dữ liệu bằng cách dùng **MATLAB** và mô tả vài công cụ thống kê cơ bản. Kỹ thuật mạnh hơn là dùng đại số tuyến tính và các hàm xử lý tín hiệu bàn đến trong phần sau.

5.1. Phân tích theo hướng cột

Tất nhiên các ma trận dùng để giữ tất cả dữ liệu, nhưng điều này dẫn đến một lựa chọn của hướng dữ liệu khác nhau. Theo quy ước thì các biến khác nhau trong tập hợp dữ liệu được đặt theo các cột, cho phép quan sát qua các dòng. một tập hợp dữ liệu gồm **50** mẫu, **13** biến được lưu trong một ma trận cỡ **50x13**.

Bắt đầu bằng ví dụ, dữ liệu kinh tế **Longley** đã có gồm các biến

- 1) Lạm phát quốc dân
- 2) Thu nhập quốc dân
- 3) Thất nghiệp
- 4) Lực lượng quân đội
- 5) Dân số
- 6) Năm
- 7) Lực lượng lao động

Nói chung có nhiều cách đưa dữ liệu vào **MATLAB**; điều này được khám phá trong phần sau. Giả sử dữ liệu chưa có trong máy đọc được thì cách nhập dữ liệu dễ nhất là dùng một trình soạn thảo văn bản hoặc trình xử lý từ. Nếu tạo ra một tệp tên là **longley.m** chứa các lệnh gán

```
ldata = [
83.0  234.289  235.6  159.0  107.608  1947  60.232
88.5  259.426  232.5  145.6  108.623  1948  61.122
88.2  258.054  368.2  161.6  109.773  1949  60.171
89.5  284.599  335.1  165.0  110.929  1950  61.187
96.2  328.975  209.9  309.9  112.075  1951  63.221
```

```

98.1 346.999 193.2 359.4 113.270 1952 63.639
99.0 365.385 187.0 354.7 115.094 1953 64.989
100.0 363.112 357.8 335.0 116.219 1954 63.761
101.2 397.469 290.4 304.8 117.388 1955 66.019
104.6 419.180 282.2 285.7 118.734 1956 67.857
108.4 442.769 293.6 279.8 120.445 1957 68.169
110.8 444.546 468.1 263.7 121.950 1958 66.513
112.6 482.704 381.3 255.2 123.366 1959 68.655
114.2 502.601 393.1 251.4 125.368 1960 69.564
115.7 518.173 480.6 257.2 127.852 1961 69.331
116.9 554.894 400.7 282.7 130.081 1962 70.551 ]

```

thì có thể thực hiện lệnh **longley**. Lệnh này truy cập tệp **longley.m** và tạo ra ma trận **ldata** (hoặc tên bất kỳ khác nếu muốn) trong vùng làm việc. Thử nhập ma trận này trong chế độ tương tác, nhưng chỉ được sửa đổi ở lần đầu. Nếu nhập sai thì không có cách sửa đổi.

Nếu quan sát nhiều hơn là có trên màn hình, các dòng có thể tiếp tục trên dòng tiếp theo bằng cách dùng dấu tính lược gồm 3 dấu chấm. Ma trận cũng có thể nhập trong các khối cột và nối toàn bộ lại ở cuối dòng.

Với dữ liệu **longley** có 16 mẫu xét gồm 7 biến. Điều này biểu lộ bởi

```
[n,p] = size(ldata)
```

```
n =
```

```
16
```

```
p =
```

```
7
```

Đối với dữ liệu nhập theo từng cột này thì một nhóm các hàm cung cấp các công cụ phân tích dữ liệu cơ bản:

Phân tích dữ liệu theo từng cột	
max	giá trị cực đại
min	giá trị cực tiểu
mean	giá trị trung bình
median	giá trị trung gian
std	độ lệch chuẩn
sort	sắp xếp
sum	tổng các phần tử
prod	tích các phần tử
cumsum	tổng tích lũy các phần tử
cumprod	tích tích lũy các phần tử
diff	đạo hàm xấp xỉ
hist	biểu đồ tần số
corrcoef	hệ số tương quan
cov	ma trận hiệp phương sai
cplxpair	Sắp lại thành cặp số phức

Đối với các đối là vector thì các hàm này không xét đến các vector có được định hướng theo dòng hay theo cột. Đối với các đối là mảng thì các hàm thực hiện theo cách định hướng cột trên dữ liệu trên mảng. Điều này có nghĩa là, chẳng hạn nếu hàm **max** áp dụng cho mảng thì kết quả là một vector dòng chứa các giá trị lớn nhất trên mỗi cột.

Do đó, nếu

```
A =
    9    8    4
    1    6    5
    3    2    7
```

thì

```
m = max(A)
mv = mean(A)
s = sort(A)
```

kết quả là

```
m =
```



```

          9  8  7
mv =
      4.3333  5.3333  5.333
s =
      1  2  4
      3  6  5
      9  8  7

```

Hoặc với dữ liệu **longley**

```

m = median(ldata)
m =
      1.0E+003*
      0.1012  0.3975  0.3351  0.2798  0.1174  1.9550  0.0660

```

Có thể trừ giá trị trung bình mỗi cột của **ldata** bằng cách dùng phép nhân bên ngoài

```
lmean = ldata - ones(n,1)*m;
```

Có thể thêm vào danh sách này bằng cách dùng các siêu tệp M-file, nhưng khi dùng hãy cẩn thận để xử lý trường hợp vector dòng. Nếu viết các tệp M-file theo hướng cột riêng thì hãy xem cách hoàn thành điều này trong các tệp M-file khác, ví dụ **mean.m** và **diff.m**.

5.2. Các giá trị bỏ qua

Giá trị đặc biệt, **NaN**, viết tắt chữ **Not-a-Number** trong **MATLAB**. Thông thường cho ra bởi các biểu thức không xác định như **0/0**, nguyên nhân của một thông báo lỗi, tùy theo quy ước thiết lập bởi chuẩn **IEEE**. Đối với công dụng thống kê thì các giá trị **NaN** có thể dùng để biểu hiện các giá trị bỏ qua hoặc dữ liệu không dùng được, **NA**.

Cách "sửa đổi" các giá trị **NA** là một điều khó khăn và thường khác nhau tùy theo từng trường hợp cụ thể. Tuy nhiên, **MATLAB** đồng dạng và nghiêm ngặt trong cách xem xét của nó về các giá trị **NaN**; chúng truyền một cách tự nhiên cho kết quả cuối cùng của mọi tính toán. Do đó nếu một giá trị **NaN** được dùng trong

mọi lần tính toán trung gian thì kết quả cuối cùng sẽ là một **NaN**, trừ khi kết quả cuối cùng không phụ thuộc vào giá trị **NaN**.

Về mặt thực hành, điều này có nghĩa là nên xóa các **NaN** trong dữ liệu trước khi thực hiện việc tính toán thống kê. Các **NaN** trong vectơ **x** được tìm ở:

```
i = find(isnan(x));
```

vì vậy

```
x = x(find(~isnan(x)))
```

trả về dữ liệu trong **x** với các **NaN** đã xóa. Có 2 cách khác nhau để thực hiện việc này là

```
x = x(~isnan(x));
```

```
x(isnan(x)) = [ ];
```

có lẽ cách thứ hai là rõ nhất. Phải dùng hàm đặc biệt **isnan** để tìm các **NaN** vì không thể dùng lệnh

```
x(x==NaN) = [ ];
```

Các **NaN** trả về **NaN** cho mọi phép toán, kể cả các phép toán quan hệ.

Nếu thay vì vectơ, dữ liệu ở trên các cột của ma trận, và xóa mọi dòng của ma trận có **NaN** thì dùng

```
X(any(isnan(X)'),:) = [ ];
```

đây là một lệnh khá thô, nhưng có hiệu lực. Nếu cho rằng khó nhớ thì hoàn toàn biện hộ. Nếu thường cần xóa các **NaN** thì cách giải quyết là viết một tệp M-file, ví dụ

```
function X = excise(X)
```

```
X(any(isnan(X)'),:) = [ ];
```

Bây giờ đánh vào

```
X = excise(X);
```

là hoàn thành cùng công việc. Biết thêm về các tệp M-file sau này.

5.3. Cách xóa các giá trị quá hạn

Cách xóa các giá trị quá hạn trong dữ liệu giống như cách xóa các NaN. Với dữ liệu **Longley**, giá trị trung bình và các độ lệch chuẩn của mỗi cột dữ liệu là:

```
mv = mean(ldata)

sigma = std(ldata)

mv =

    1.0E+003*

    0.101    0.387    0.319    0.260    0.117    1.954    0.065

sigma =

    10.448    96.238    90.479    67.382    6.735    4.609    3.400
```

Số dòng có giá trị chênh lệch lớn hơn 3 lần độ lệch chuẩn là:

```
[n,p] = size(ldata);
e = ones(n,1);
dist = asb(ldata-e*mv);
outliers = dist > 3*e*sigma;
nout = sum(any(outliers'))
nout =
    0
```

Không có. Nếu có thì chúng bị xóa với lệnh

```
X(any(outliers'),:) = [ ];
```

5.4. Hồi quy và đường cong thực nghiệm

Trước khi đưa đường cong thực nghiệm vào dữ liệu thì phải chuẩn hóa dữ liệu. Việc chuẩn hóa có thể cải thiện độ chính xác của kết quả cuối cùng. Vẫn làm việc với dữ liệu Longley, một cách chuẩn hóa là xóa giá trị trung bình

```
X = X - e* mean(X);
```

và để chuẩn hóa thành đơn vị độ lệch chuẩn

```
X = X ./ (e*std(X));
```

Có thể tính hồi quy thất nghiệp (cột cuối cùng) theo các cột trước đó, dùng trong dữ liệu thô trong trường hợp này,

```
y = ladat(:,7);
```

```
A = [ldata(:,1:6) ones(y)];
```

```
coef = A\y
```

kết quả là

```
coef =
```

```
1.0E+0003
```

```
0.00001506187227
```

```
-0.00003581917929
```

```
-0.00002020229804
```

```
-0.00001033226867
```

```
-0.00005110410565
```

```
0.00182915146461
```

```
-3.48225863459802
```

Dữ liệu **Longley** có tương quan cao, xem qua các hệ số tương quan.

```
corr(X)
```

```
ans =
```

```
1.0000 0.9916 0.6206 0.4647 0.9792 0.9911 0.9709
0.9916 1.0000 0.6043 0.4464 0.9911 0.9953 0.9836
0.6206 0.6043 1.0000 -0.1774 0.6866 0.6683 0.5025
0.4647 0.4464 -0.1774 1.0000 0.3644 0.4172 0.4573
0.9792 0.9911 0.6866 0.3644 1.0000 0.9940 0.9604
0.9911 0.9953 0.6683 0.4172 0.9940 1.0000 0.9713
0.9709 0.9836 0.5025 0.4573 0.9604 0.9713 1.0000
```

Thường đưa đa thức vào dữ liệu là có ích. Nói chung, một đa thức đưa vào dữ liệu theo các vector \mathbf{x} và \mathbf{y} là một hàm, \mathbf{p} , có dạng:

$$p(x) = c_1 x^d + c_2 x^{d-1} + \dots + c_n$$

Cấp là d và số hệ số là $n = d + 1$. Các hệ số c_1, c_2, \dots, c_n được xác định bằng cách giải hệ phương trình tuyến tính:

$$\mathbf{A}\mathbf{c} = \mathbf{y}$$

Các cột của \mathbf{A} là lũy thừa thoái của vector \mathbf{x} . Sau đây là một cách tạo ra \mathbf{A}

```
for j=1:n
    A(:,j) = x.^(n-j);
end
```

Lời giải của hệ phương trình tuyến tính $\mathbf{A}\mathbf{c} = \mathbf{y}$ nhận được với phép chia ma trận của **MATLAB**:

$$\mathbf{c} = \mathbf{A} \backslash \mathbf{y}$$

Hàm **polyfit.m** trong **MATLAB TOOLBOX** tự động làm thủ tục này.

Trong bài toán hồi quy, các hàm khác, thường là hàm nhiều biến các cột của ma trận dữ liệu, được đưa vào dữ liệu bằng cách tìm dạng của ma trận \mathbf{A} tương ứng. Ví dụ, dùng dữ liệu **longley**,

```
A = [ldata(:,1) ldata(:,2).^2 sin(ldata(:,3)) ones(n,1)];
coef = A \ y;
```

tìm các hệ số hồi quy cho một hàm phức tạp hơn.

Chương 6. HÀM MA TRẬN

Nhiều khả năng toán học của **MATLAB** nhận được từ các hàm ma trận của nó. Một số hàm gắn liền với trình xử lý **MATLAB**. Các hàm khác, trong thư viện các tệp M-file, phân phối cùng với **MATLAB**. Và một số được thêm vào bởi từng người dùng, hoặc nhóm người cho các trình ứng dụng đặc biệt. ở đây chúng tôi không đi sâu vào chi tiết từng hàm; điều đó được thực hiện trong công cụ trợ giúp và phần tham khảo. Thông tin thêm nữa cũng có thể tìm trong hướng dẫn sử dụng phần mềm **LINPACK** và **EISPACH**, cung cấp cơ bản về thuật toán cho **MATLAB**. Trong phần này, chúng tôi cho xem qua các hàm được nhóm theo các hàm thừa số ma trận và hàm phân tích ma trận.

Gồm 4 nhóm:

- Thừa số tam giác
- Thừa số trực giao
- Tách giá trị riêng
- Tách giá trị kỳ dị

6.1. Thừa số tam giác

Cách tách thừa số cơ bản nhất là tách ma trận vuông bất kỳ thành tích **2** ma trận tam giác, một ma trận là hoán vị của một ma trận tam giác dưới và ma trận kia là ma trận tam giác trên. Việc tách thừa số thường gọi là "thừa số **LU**" hoặc đôi khi gọi là "thừa số **LR**". Hầu hết các thuật toán để tính là các phép biến đổi theo phương pháp khử **Gauss**.

Chính các thừa số lấy được từ hàm **lu**. Các thừa số được dùng để nhận nghịch đảo ma trận với hàm **inv** và lấy định thức với hàm **det**. Đó cũng là cơ sở cho việc giải hệ phương trình tuyến tính hay "chia ma trận vuông" với các toán tử \ và /.

Ví dụ, bắt đầu với

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}$$

Để xem phân tích LU, dùng câu lệnh gán kép của **MATLAB**.

$$[L,U] = \text{lu}(A)$$

cho ra

$$L =$$

$$\begin{bmatrix} 0.1429 & 1.0000 & 0 \\ 0.5714 & 0.5000 & 1.0000 \\ 1.0000 & 0 & 0 \end{bmatrix}$$

$$U =$$

$$\begin{bmatrix} 7.0000 & 8.0000 & 0.0000 \\ 0 & 0.8571 & 3.0000 \\ 0 & 0 & 4.5000 \end{bmatrix}$$

Lưu ý rằng **L** là hoán vị của ma trận tam giác dưới có các số 1 trên đường chéo, và **U** là tam giác trên. Để kiểm tra thừa số có thể tính tích

$$L*U$$

cho ra giá trị **A** gốc. Đó là,

$$\text{ans} =$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}$$

Ma trận đảo của ma trận ví dụ nhận được với

$$X = \text{inv}(A)$$

Ma trận đảo được tính toán thực sự qua các nghịch đảo của các thừa số tam giác

$$X = \text{inv}(U)*\text{inv}(L)$$

Định thức của ma trận ví dụ nhận được với

$$\mathbf{d} = \det(\mathbf{A})$$

cho ra

$$\mathbf{d} =$$

$$27$$

Được tính từ định thức của các thừa số tam giác

$$\mathbf{d} = \det(\mathbf{L}) * \det(\mathbf{U})$$

cho ra

$$\mathbf{d} =$$

$$27.0000$$

Tại sao 2 lần in ra \mathbf{d} ở dạng khác nhau? Khi **MATLAB** được yêu cầu tính $\det(\mathbf{A})$, nó nhận thấy tất cả các phần tử của \mathbf{A} là nguyên, vì vậy nó cho định thức là số nguyên. Nhưng khi tính lần hai, các phần tử của \mathbf{U} không nguyên, vì vậy **MATLAB** không cho kết quả là số nguyên.

Như một ví dụ về hệ phương trình tuyến tính, lấy

$$\mathbf{b} =$$

$$1$$

$$3$$

$$5$$

Lời giải phương trình $\mathbf{Ax} = \mathbf{b}$ nhận được với phép chia ma trận của **MATLAB**

$$\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$$

cho ra

$$\mathbf{x} =$$

$$0.3333$$

$$0.3333$$

$$0.0000$$

Lời giải được tính toán thực sự bằng cách giải 2 hệ tam giác,

$$\mathbf{y} = \mathbf{L} \backslash \mathbf{b}, \mathbf{x} = \mathbf{U} \backslash \mathbf{y}$$

Lời giải trung gian là

$$y = \begin{bmatrix} 5.0000 \\ 0.2857 \\ 0.0000 \end{bmatrix}$$

Thừa số tam giác cũng được dùng bởi một hàm đặc biệt là **rcond**. Đây là sản phẩm của một số chương trình con của **LINPACK** để ước lượng số điều kiện tính nghịch đảo của một ma trận vuông.

Hai hàm khác, **chol** và **rref**, có thể được đưa vào nhóm này vì thuật toán cơ bản quan hệ gần gũi với thừa số **LU**. Hàm **chol** cho thừa số **Cholesky** của một ma trận đối xứng xác định dương. Dạng xếp bậc thu gọn dòng của ma trận chữ nhật, **rref**, có một ít đáng quan tâm trong lý thuyết đại số tuyến tính, mặc dù có giá trị tính toán không lớn. Nó được đưa vào **MATLAB** vì tính sự phạm.

6.2. Thừa số trực giao

Thừa số "**QR**" là hữu ích cho cả ma trận vuông lẫn ma trận chữ nhật. Nó tách một ma trận thành tích của một ma trận trực chuẩn và một ma trận tam giác trên. Ví dụ, lấy

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}$$

là ma trận khuyết hạng; cột giữa là trung bình cộng của hai cột kia. Tính khuyết hạng có thể được biểu lộ bằng thừa số.

$$[Q,R] = \text{qr}(A)$$

cho ra

$$Q = \begin{bmatrix} -0.0776 & -0.8331 & 0.5444 & 0.0605 \\ -0.3105 & -0.4512 & -0.7709 & 0.3251 \end{bmatrix}$$

$$\begin{array}{cccc}
 -0.5433 & -0.0694 & -0.0913 & -0.8317 \\
 -0.7762 & 0.3124 & 0.3178 & 0.4461 \\
 \mathbf{R} = & & & \\
 -12.8841 & -14.5916 & -16.2992 & \\
 0 & -1.0413 & -2.0826 & \\
 0 & 0 & 0.0000 & \\
 0 & 0 & 0 &
 \end{array}$$

Có thể kiểm tra rằng tích $\mathbf{Q}^*\mathbf{R}$ cho ra \mathbf{A} gốc, nhưng đừng băn khoăn điều đó. Cấu trúc tam giác của \mathbf{R} cho nó các số **0** dưới đường chéo; số **0** trên đường chéo ở $\mathbf{R}(3,3)$ cho thấy rằng \mathbf{R} , và do đó \mathbf{A} , không phải là ma trận đủ hạng.

Phân tích thừa số \mathbf{QR} được dùng để giải các hệ phương trình tuyến tính với số phương trình nhiều hơn số ẩn. Ví dụ

$$\mathbf{b} = \begin{array}{c} 1 \\ 3 \\ 5 \\ 7 \end{array}$$

Hệ phương trình $\mathbf{Ax}=\mathbf{b}$ là hệ 4 phương trình chỉ có 3 ẩn. Lời giải tốt nhất theo phương pháp bình phương bé nhất được tính bởi

$$\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$$

cho ra

Warning: Rank deficient, rank = 2 tol = 1.4594E-014

$$\mathbf{x} = \begin{array}{c} 0.5000 \\ 0.0000 \\ 0.1667 \end{array}$$

Báo trước về sự khuyết hạng. Giá trị **tol** dùng để xác định rằng một phần tử trên đường chéo của \mathbf{R} là không đáng kể. Lời giải \mathbf{x} được tính bằng cách tách thừa số và qua 2 bước

$$\begin{aligned} \mathbf{y} &= \mathbf{Q}' * \mathbf{b}; \\ \mathbf{x} &= \mathbf{R} \backslash \mathbf{y} \end{aligned}$$

Nếu kiểm tra lời giải đã tính theo công thức $\mathbf{A} * \mathbf{x}$ thì thấy rằng nó bằng \mathbf{b} với sai số làm tròn. Điều này nói rằng mặc dù hệ phương trình $\mathbf{A}\mathbf{x}=\mathbf{b}$ là vô định và khuyết hạng nhưng chúng vẫn thích hợp. Có nhiều lời giải vô định vector \mathbf{x} ; thừa số \mathbf{QR} tìm ra một lời giải trong chúng.

Việc phân tích thừa số này cũng là cơ sở cho các hàm **null** và **orth**, chúng phát sinh các cơ sở trực giao cho không gian $\mathbf{0}$ và phạm vi của một ma trận đã cho.

6.3. Tách giá trị kỳ dị

Chúng tôi không có ý giải thích cách tách giá trị kỳ dị ở đây; chúng ta phải chấp nhận với ý kiến cho rằng nó là công cụ mạnh mẽ cho việc giải các bài toán về ma trận. Xem sách hướng dẫn sử dụng **LINPACK** hoặc sách viết bởi **Golub** và **VanLoan** đối với vấn đề này. Trong **MATLAB**, lệnh gán ba

$$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \text{svd}(\mathbf{A})$$

cho ra 3 thừa số trong việc tách giá trị kỳ dị,

$$\mathbf{A} = \mathbf{U} * \mathbf{S} * \mathbf{V}'$$

\mathbf{U} và \mathbf{V} là ma trận trực giao và \mathbf{S} là ma trận chéo. Bằng chính nó, hàm **svd(A)** trả về đúng các phần tử trên đường chéo của \mathbf{S} , đó là các giá trị kỳ dị của \mathbf{A} .

Việc tách giá trị kỳ dị được sử dụng cho một số hàm khác, kể cả hàm giả đảo, **pinv(A)**; tính hạng, **rank(A)**; chuẩn ma trận **O-clit**, **norm(A,2)**; và số điều kiện, **cond(A)**.

6.4. Giá trị riêng

Nếu \mathbf{A} là ma trận vuông cấp n thì n số λ thỏa mãn $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ gọi là các giá trị riêng của \mathbf{A} . Chúng được tính bằng

$$\text{eig}(\mathbf{A})$$

trả về các giá trị riêng trong một vector cột. Nếu \mathbf{A} là ma trận thực và đối xứng thì các giá trị riêng là thực. Nhưng nếu \mathbf{A} không đối xứng thì các giá trị riêng luôn là số phức. Ví dụ, với

$$\mathbf{A} = \begin{bmatrix} & 1 \\ 0 & \end{bmatrix}$$

-1 0

Lệnh **eig(A)** cho ra

ans =

0.0000 + 1.0000 i

0.0000 - 1.0000 i

Các vector riêng và các giá trị riêng nhận được với lệnh gán kép,

[X,D] = eig(A)

Trong trường hợp này các phần tử trên đường chéo của **D** là các giá trị riêng và các cột của **X** là các vector riêng tương ứng mà **A*X = X*D**.

Hai kết quả trung gian dùng trong việc tính các giá trị riêng là chuẩn **Hessenberg**, **hess(A)**, và chuẩn **Schur**, **Schur(A)**. Chuẩn **Schur** dùng để tính các hàm ma trận siêu việt, như **sqrtnm(A)** và **logm(A)**.

Nếu **A** và **B** là các ma trận vuông thì hàm **eig(A,B)** trả về một vector chứa các giá trị riêng suy rộng từ lời giải phương trình

$$Ax = \lambda Bx$$

Lệnh gán kép dùng để nhận các vector riêng

[X,D] = eig(A,B)

cho ra ma trận chéo **D** gồm các giá trị riêng suy rộng và ma trận **X** đầy đủ có các cột là các vector riêng tương ứng mà **A*X = B*X*D**. Các kết quả trung gian trong lời giải của bài toán giá trị riêng suy rộng này có thể dùng từ hàm **qz(A,B)**.

6.5. Hạng và điều kiện

Các hàm của **MATLAB** liên quan đến hạng và điều kiện gồm:

Điều kiện về ma trận	
Cond	Số điều kiện trong chuẩn 2
norm	chuẩn 1, chuẩn 2, chuẩn F, chuẩn ∞
rank	hạng ma trận
rcond	ước lượng điều kiện

Có nhiều nơi trong **MATLAB** có tính hạng ma trận: trong **rref(A)**, trong **A\B** với **A** không vuông, trong **orth(A)** và **null(A)**, và trong giả đảo **pinv(A)**. Ba thuật toán khác nhau với ba tiêu chuẩn khác nhau không đáng kể và vì vậy ba giá trị khác nhau đó có thể cho ra ma trận giống nhau.

Với **rref(A)** thì hạng của **A** là số dòng khác không. Thuật toán khử của **rref** là nhanh nhất trong 3 thuật toán xác định hạng ma trận, nhưng ít tinh vi và ít tin cậy nhất.

Với **A\B**, **orth(A)**, và **null(A)**, cách phân tích thừa số **QR** được dùng như mô tả trong chương 9 sách hướng dẫn **LINPACK**.

Với **pinv(A)**, thuật toán dựa vào cách phân tích giá trị kỳ dị và được mô tả trong chương 11 sách hướng dẫn **LINPACK**. thuật toán **pinv** tốn thời gian nhất, nhưng đáng tin cậy nhất và do đó cũng được dùng để tính hạng ma trận, **rank(A)**.

Chương 7. ĐA THỨC VÀ XỬ LÝ TÍN HIỆU

7.1. Đa thức

Các đa thức biểu hiện trong **MATLAB** như các vector dòng chứa các hệ số theo lũy thừa thoái. Ví dụ, Phương trình đặc trưng của ma trận

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

được tính với

$$\begin{aligned} p &= \text{poly}(A) \\ p &= \\ &1 \quad -6 \quad -72 \quad -27 \end{aligned}$$

Đây là biểu hiện **MATLAB** của ma trận $s^3 - 6s^2 - 72s - 27$.

Các nghiệm của phương trình này là:

$$\begin{aligned} r &= \text{roots}(p) \\ r &= \\ &12.1229 \\ &-5.7345 \\ &-0.3884 \end{aligned}$$

Tất nhiên giống các giá trị riêng của ma trận **A**. Có thể ráp ngược về đa thức gốc với hàm **poly**,

$$\begin{aligned} p2 &= \text{poly}(r) \\ p2 &= \\ &1 \quad -6 \quad -72 \quad -27 \end{aligned}$$

Xét các đa thức $a(s) = s^2 + 2s + 3$ và $b(s) = 4s^2 + 5s + 6$. Tích các đa thức là tích chập các hệ số:

a = [1 2 3]; b = [4 5 6];

c = conv(a,b)

c =

4 13 28 27 18

Dùng hàm tách tích chập để chia ngược lại,

[q,r] = deconv(c,a)

q =

4 5 6

r =

0 0 0 0 0

Danh sách đầy đủ các hàm về đa thức gồm:

Đa thức	
poly	Đa thức đặc trưng
roots	Nghiệm đa thức-phương pháp ma trận liên hợp
roots1	Nghiệm đa thức-phương pháp Laguerre
polyval	Ước lượng đa thức
polyvalm	Ước lượng đa thức ma trận
conv	Nhân đa thức
deconv	Chia đa thức
residue	Khai triển thừa số từng phần
polyfit	Vẽ đường cong đa thức

7.2. Xử lý tín hiệu

Các vectơ dùng để giữ các tín hiệu dữ liệu mẫu, hoặc chuỗi, cho việc xử lý tín hiệu. Đối với hệ thống nhiều dữ kiện nhập, mỗi dòng ứng với một điểm mẫu, với việc quan sát băng qua các cột của ma trận. Một vài hàm xử lý tín hiệu được đưa vào hệ thống chính của **MATLAB**:

Xử lý tín hiệu	
abs	Chuẩn của số phức
angle	Góc pha
conv	Tích chập

cov	Hiệp phương sai
deconv	Tách tích chập
fit	Biến đổi Fourier nhanh
ifft	Nghịch đảo biến đổi Fourier nhanh
fftshift	Hoán đổi dạng toàn phương ma trận

Vài hàm có bản sao 2 chiều, trong trường hợp đó "tín hiệu" đúng là một ma trận:

Xử lý tín hiệu 2 chiều	
fft2	FFT 2 chiều
ifft2	FFT 2 chiều ngược
fftshift	Sắp xếp lại các kết quả FFT
conv2	Tích chập 2 chiều

Có nhiều hàm xử lý tín hiệu nữa có thể sử dụng trong **SIGNAL PROCESSING TOOLBOX**. Phần này có ý giới thiệu sơ bộ về khả năng xử lý tín hiệu của **MATLAB**; để biết thêm thông tin xem riêng sách hướng dẫn sử dụng **SIGNAL PROCESSING TOOLBOX**.

7.3. Lọc dữ liệu

Trong **SIGNAL PROCESSING TOOLBOX**, hàm

$$\mathbf{y} = \text{filter}(\mathbf{b}, \mathbf{a}, \mathbf{x})$$

lọc dữ liệu trong vector \mathbf{x} với bộ lọc mô tả bởi các vector \mathbf{a} và \mathbf{b} , tạo ra dữ liệu \mathbf{y} đã lọc. Cấu trúc lọc là bộ lọc dãy tổng quát mô tả bởi phương trình vi phân:

$$\begin{aligned} \mathbf{y}(n) = & \mathbf{b}(1)\mathbf{x}(n) + \mathbf{b}(2)\mathbf{x}(n-1) + \dots + \mathbf{b}(nb)\mathbf{x}(n-nb+1) \\ & - \mathbf{a}(2)\mathbf{y}(n-1) - \dots - \mathbf{a}(na)\mathbf{y}(n-na+1) \end{aligned}$$

hoặc tương đương phép biến đổi \mathbf{Z}

$$\mathbf{H}(\mathbf{z}) = \frac{\mathbf{Y}(\mathbf{z})}{\mathbf{X}(\mathbf{z})} = \frac{\mathbf{b}(1) + \mathbf{b}(2)\mathbf{z}^{-1} + \dots + \mathbf{b}(nb)\mathbf{z}^{-(nb-1)}}{1 + \mathbf{a}(2)\mathbf{z}^{-1} + \dots + \mathbf{a}(na)\mathbf{z}^{-(na-1)}}$$

Ví dụ, để tìm và vẽ đơn vị n điểm đáp ứng bộ lọc:

$$\mathbf{x} = [1 \text{ zeros}(1, n-1)];$$

$$\mathbf{y} = \text{filter}(\mathbf{b}, \mathbf{a}, \mathbf{x});$$

plot(y,'o')

Hàm **freqz** trả về kết quả phức của bộ lọc số. Kết quả là hàm **H(z)** ước lượng quanh đơn vị tròn trong mặt phẳng phức, $\mathbf{x} = e^{j\omega}$. Để dùng **freqz** để tìm và vẽ đường cong thực nghiệm **n** điểm:

[h,w] = frqz(b,a,n);

mag = abs(h);

phase = angle(h);

semilogy(w,mag), plot(w,phase)

Các hàm có thể sử dụng trong **SIGNAL PROCESSING TOOLBOX** dùng để thiết kế bộ lọc số. Chúng tôi đưa nội dung vào đây để yêu cầu một số kiến thức về kỹ thuật thiết kế bộ lọc, có thể dùng nhiều phương pháp. Ví dụ, các hàm số học về số phức cho phép vẽ kỹ thuật giống như biến đổi song tuyến tính và vẽ cực **0** để đổi sang các nguyên mẫu **s-phạm trù** sang **z-phạm trù**. Cũng như thế, các bộ lọc **FIR** được thiết kế một cách dễ dàng bằng kỹ thuật về cửa sổ.

7.4. *FFT(Fast Fourier Transform-Biến đổi Fourier nhanh)*

Phải nói rằng thuật toán **FFT** chủ yếu dùng cho việc tính toán phép biến đổi **Fourier** của chuỗi là thích hợp của việc xử lý tín hiệu số. Miền giá trị sử dụng của nó từ việc lọc dữ liệu, tích chập, tính toán các yêu cầu thường xuyên đến các trình ứng dụng trong việc ước lượng quang năng.

Hàm **fft(x)** là phép biến đổi **Fourier** của vector **x**, tính toán biến đổi **Fourier** cơ sở **2** nhanh nếu độ dài của **x** là bội lũy thừa của **2**, và với thuật toán chuyển cơ sở nếu độ dài của **x** không phải là bội lũy thừa của **2**. Nếu **X** là ma trận thì **fft(X)** là biến đổi **Fourier** nhanh của mỗi cột của **X**.

Hàm **fft(x,n)** là **FFT n-điểm**. Nếu độ dài của **x** nhỏ hơn **n** thì **x** được thêm với đuôi các số **0** thành độ dài **n**. Nếu độ dài của **x** lớn hơn **n** thì **x** bị cắt phần đuôi. Khi **X** là ma trận thì độ dài các cột của **X** được chỉnh lý theo cùng cách này.

Hàm **ifft(x)** là phép biến đổi **Fourier** ngược của vector **x**, hàm **ifft(x,n)** là **FFT ngược n-điểm**.

Cặp hai hàm cài đặt phép biến đổi và biến đổi ngược cho bởi:

$$\mathbf{X}(k+1) = \sum_{n=0}^{N-1} x(n+1) W_N^{kn}$$

$$\mathbf{x}(n+1) = 1/N \sum_{k=0}^{N-1} X(k+1) W_N^{-kn}$$

ở đây $W_N = e^{-j(2\pi/N)}$ và $N = \text{length}(\mathbf{x})$. Lưu ý rằng chỉ số được viết theo cách không chính tắc chạy đến $n+1$ và $k+1$ thay vì đến n và k bình thường, vì các vector của **MATLAB** chạy từ 1 đến N thay vì từ 0 đến $N-1$.

Giả sử một dãy độ dài chẵn gồm N điểm có cùng tần số mẫu của f_s . Sau đó chuyển sang tần số **Nyquist**, hoặc điểm $n = N/2+1$, thì quan hệ giữa số nhị phân và tần số thực là:

$$f = (\text{bin_number} - 1) * f_s / N$$

FFT của vector cột \mathbf{x}

$$\mathbf{x} = [4 \ 3 \ 7 \ -9 \ 1 \ 0 \ 0 \ 0]';$$

tìm thấy với

$$\mathbf{y} = \text{fft}(\mathbf{x})$$

kết quả là

$$\begin{aligned} \mathbf{y} = & \\ & 6.0000 \\ & 11.4853 - 2.7574 i \\ & -2.0000 - 12.0000 i \\ & -5.4853 + 11.2426 i \\ & 18.0000 \\ & -5.4853 - 11.2426 i \\ & -2.0000 + 12.0000 i \\ & 11.4853 + 2.7574 i \end{aligned}$$

Lưu ý rằng mặc dù dãy \mathbf{x} là thực, nhưng \mathbf{y} lại là phức. Thành phần thứ nhất của dữ liệu biến đổi là đóng góp **DC** và phần tử thứ năm tương ứng với tần số **Nyquist**. Ba giá trị cuối cùng của \mathbf{y} tương ứng các tần số âm và, đối với dãy số thực \mathbf{x} , chúng là các liên hợp phức của $\mathbf{y}(4)$, $\mathbf{y}(3)$ và $\mathbf{y}(2)$.

Để biết thêm thông tin thì xem phần tham khảo. Nếu làm nhiều với việc xử lý tín hiệu thì xem sách hướng dẫn sử dụng **SIGNAL PROCESSING TOOLBOX**.

Chương 8. HÀM CÓ ĐỐI SỐ LÀ HÀM

Một lớp các hàm trong **MATLAB** không làm việc với các ma trận số mà với các hàm toán học. Các *hàm có đối số là hàm* này gồm:

Tích phân số

Phương trình và tối ưu phi tuyến

Giải phương trình vi phân

Các hàm toán học được biểu hiện trong **MATLAB** bởi các tệp M-file về hàm. Ví dụ, hàm

$$\text{humps}(x) = \frac{1}{(x-3)^2 + .01} + \frac{1}{(x-9)^2 + .04} - 6$$

được tạo ra có thể dùng trong **MATLAB** bằng cách tạo ra tệp M-file có tên là **humps.m**:

```
function y = humps(x)
```

```
y = 1 ./ ((x-3).^2+.01) + 1 ./ ((x-9).^2 +.04) - 6;
```

Đồ thị của hàm là:

```
x = -1:.01:2;
```

```
plot(x,humps(x))
```

Kết quả là hình 8.1.

8.1. Tích phân số

Diện tích vùng dưới hàm **humps(x)** có thể được xác định bằng tích phân hàm **humps(x)**, một cách xử lý xem như phép cầu phương. Để lấy tích phân hàm **humps** từ 0 đến 1:

```
q = quad('humps',0,1)
```

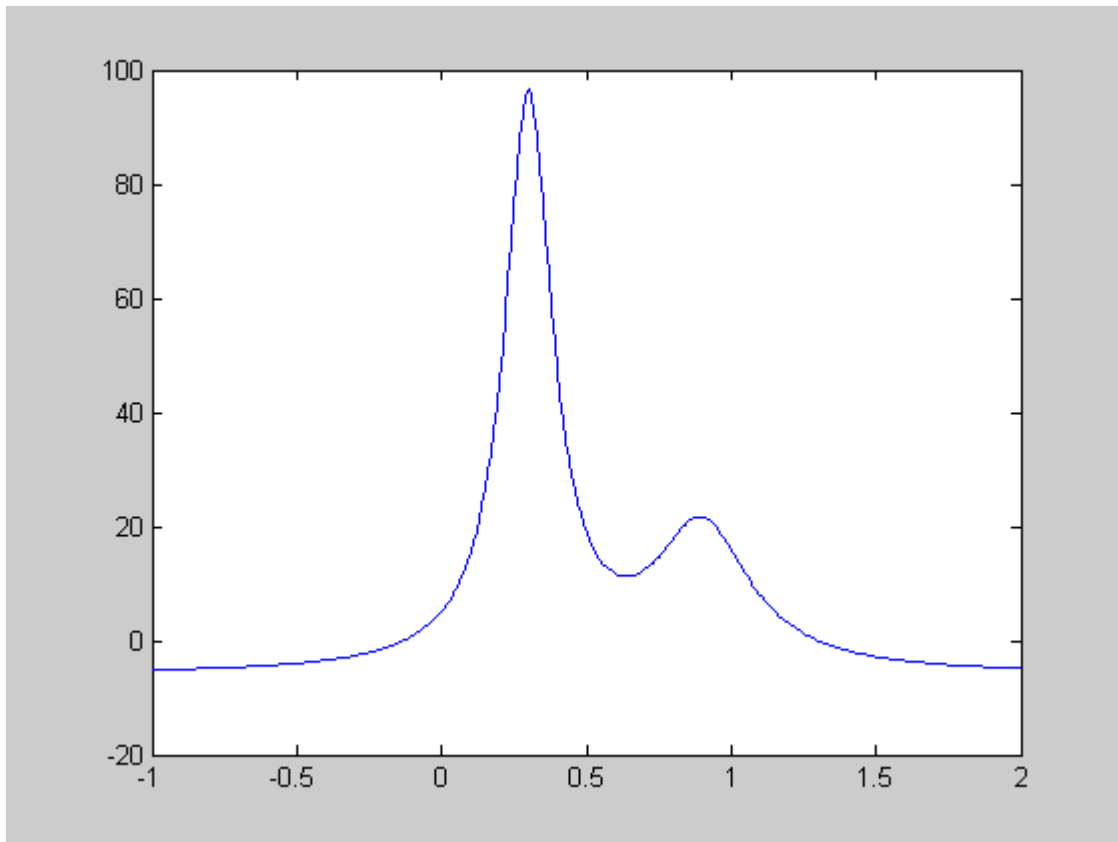
```
q =
```

```
29.8583
```

Hai hàm của **MATLAB** để tính cầu phương là:

Tích phân số	
quad	Phương pháp Simson
quad8	Phương pháp Newton

Lưu ý rằng đối số thứ nhất của hàm **quad** là xâu chữ đặt trong cặp dấu nháy chứa tên của một hàm. Điều này cho thấy tại sao gọi **quad** là một *hàm có đối số là hàm* - là một hàm tính toán trên các hàm khác.



Hình 8.1

8.2. Phương trình và tối ưu phi tuyến

Các hàm về hàm dùng cho phương trình và tối ưu phi tuyến gồm:

Phương trình và tối ưu phi tuyến	
fmin	Cực tiểu của hàm một biến
fmins	Cực tiểu của hàm nhiều biến (Tối ưu phi tuyến không ràng buộc)
fsolve	Lời giải của hệ phương trình phi tuyến (giá trị không của hàm nhiều biến)
fzero	Giá trị không của hàm một biến

Tiếp tục ví dụ, vị trí của giá trị cực tiểu của hàm **humps(x)** trong miền từ **0.5** đến **1** được tính với **fmin**:

```
xm = fmin('humps', .5, 1)
```

```
xm =
```

```
0.6370
```

Giá trị của nó ở điểm cực tiểu là:

```
ym = humps(xm)
```

```
ym =
```

```
11.2528
```

Theo đồ thị, rõ ràng **humps** có **2** giá trị **0**. Vị trí của giá trị **0** gần **x = 0** là:

```
xz1 = fzero('humps', 0)
```

```
xz1 =
```

```
-0.1316
```

Vị trí của giá trị **0** gần **x = 1** là:

```
xz2 = fzero('humps', 1)
```

```
xz2 =
```

```
1.2995
```

8.3. Phương trình vi phân

Các hàm của **MATLAB** để giải *phương trình vi phân thường* là:

Giải phương trình vi phân	
ode23	Phương pháp Runge-Kutta cấp 2/3
ode45	Phương pháp Runge-Kutta-Fehlberg cấp 4/5

Xét phương trình vi phân cấp 2 Van der Pol

$$\ddot{x} + (x^2 - 1)\dot{x} + x = 0$$

Có thể viết lại phương trình này như một hệ gồm cặp phương trình vi phân cấp một:

$$\dot{x}_1 = x_1(1 - x_2^2) - x_2$$

$$\dot{x}_2 = x_1$$

Bước thứ nhất đối với việc mô phỏng hệ này là tạo ra một tệp M-file hàm chứa các phương trình vi phân này. Có thể gọi nó là **vdpol.m**:

```
function xdot = vdpol(t,x)

xdot(1) = x(1) .* (1 - x(2) .^ 2) - x(2);

xdot(2) = x(1);

xdot = xdot'; % phiên bản mới yêu cầu vector cột
```

Để mô phỏng phương trình vi phân **vdpol** xác định trên đoạn $0 \leq t \leq 20$, gọi hàm **ode23**

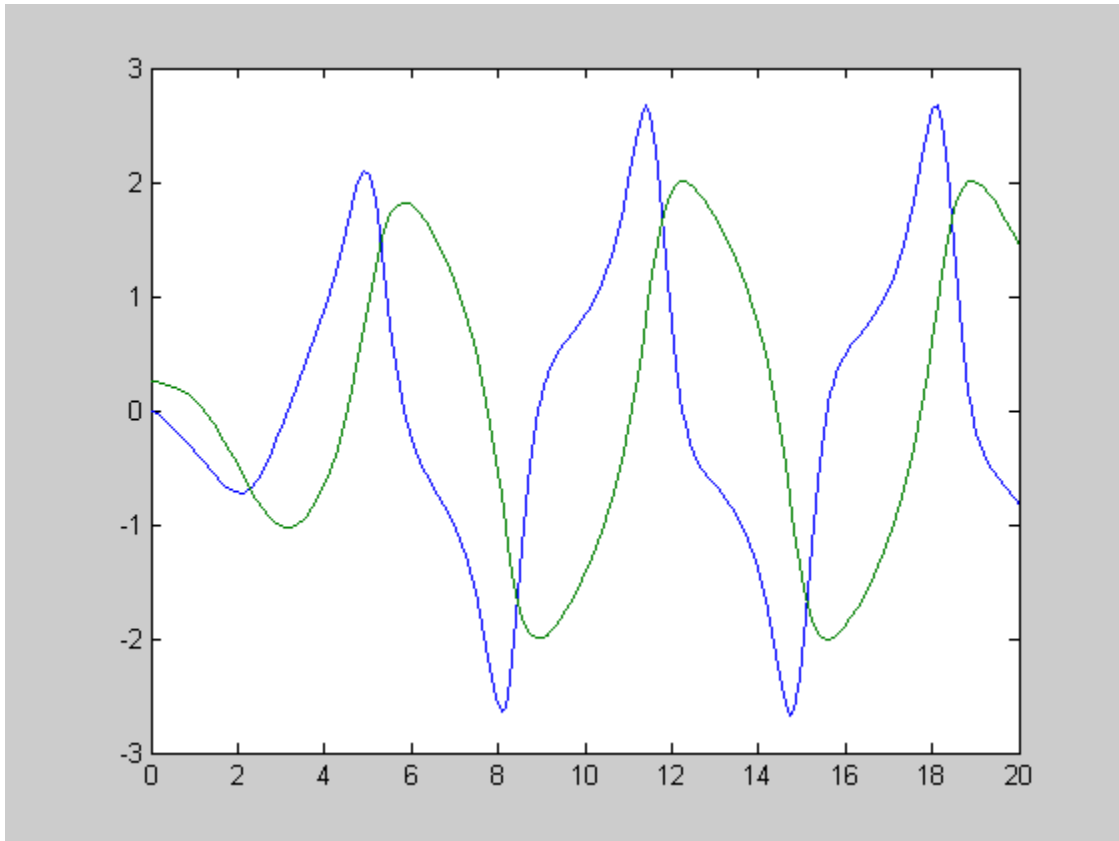
```
t0 = 0; tf = 20;

x0 = [ 0  0.25]'; % các điều kiện khởi đầu

[t,x] = ode23('vdpol', t0, tf, x0);

plot(t,x)
```

Kết quả là hình 8.2.



Hình 8.2

Chương 9. ĐỒ THỊ

Dữ liệu về khoa học và kỹ thuật được xét đến ở dạng đồ thị trong MATLAB bằng cách dùng các lệnh vẽ đồ họa để tạo ra hình vẽ trên màn hình. Có nhiều kiểu đồ họa khác nhau có thể chọn:

Đồ thị	
plot	Vẽ đường tuyến tính x-y
loglog	Vẽ loga x-y
semilogx	Vẽ bán loga x-y (loga trục x)
semilogy	Vẽ bán loga x-y (loga trục y)
polar	Vẽ tọa độ cực
mesh	Vẽ mặt lưới 3 chiều
contour	Vẽ đường mức
bar	Vẽ biểu đồ
stairs	Vẽ đồ thị bậc thang

Khi một đồ thị có trên màn hình thì có thể có nhãn, tiêu đề, hoặc các dòng lưới theo:

title	Tiêu đề đồ thị
xlabel	Nhãn trục x
ylabel	Nhãn trục y
text	Văn bản ở vị trí bất kỳ
gtext	Văn bản ở vị trí chuột
grid	Các dòng lưới

Có các lệnh để điều khiển việc chia trục và đồ thị:

axis	Chia trục
hold	Giữ hình vẽ trên màn hình
shg	Hiện màn hình đồ thị
clg	Xóa màn hình đồ thị
subplot	Chia màn hình đồ thị thành các cửa sổ con
ginput	Dấu chữ thập cho chuột

Và có các lệnh để đưa bản sao ra máy in:

print	Đưa đồ thị ra máy in
prtsc	In đồ thị màn hình
meta	Tạo siêu tệp đồ thị

9.1. Hình vẽ trong mặt phẳng x-y

Lệnh **plot** tạo ra các hình vẽ mặt phẳng x-y. Khi lệnh **plot** là chủ thì các hình vẽ loga và cực được tạo ra bằng cách thay các từ **loglog**, **semilogx**, **semilogy**, hoặc **polar** cho từ **plot**. Cả 5 lệnh được dùng cùng một cách; chúng chỉ ảnh hưởng đến cách chia trục và cách hiển thị dữ liệu.

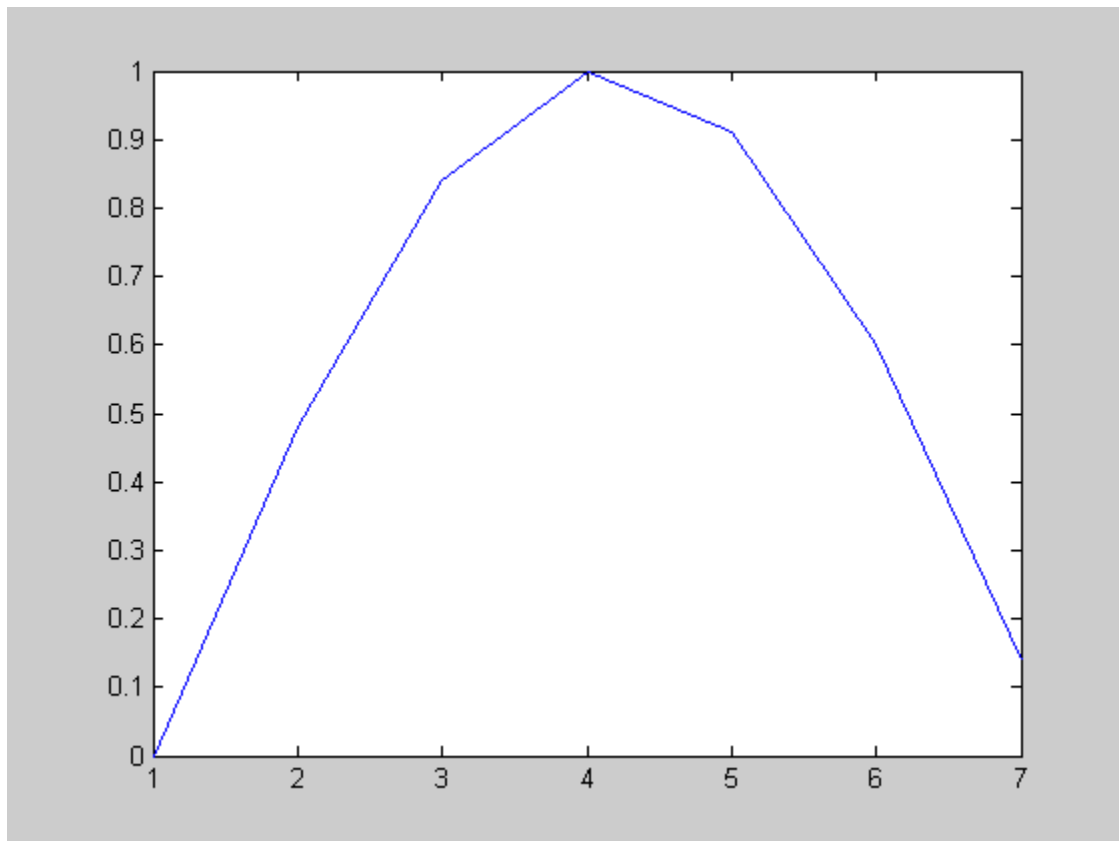
9.2. Dạng thức cơ bản

Nếu **Y** là một vector thì lệnh **plot(Y)** cho ra một hình vẽ gồm các phần tử của **Y** đối số là chỉ số của các phần tử của **Y**. Ví dụ, để vẽ các số {0., .48, .84, 1, .91, .6, .14}, nhập chúng vào một vector và thực hiện lệnh **plot**:

```
Y = [0. .48 .84 1. .91 .6 .14];
```

```
plot(Y)
```

Kết quả là hình 9.1.



Hình 9.1

Lưu ý rằng dữ liệu được chia trực tự động và các trục **X** và **Y** được vẽ. Ở điểm này, tùy theo phần cứng của máy sử dụng mà màn hình có đáp ứng các lệnh đánh vào hay không. **MATLAB** có hai màn hình, một màn hình đồ họa và một màn hình lệnh. Một số cấu hình phần cứng cho phép cả hai màn hình hiện đồng thời, trong khi một số khác chỉ hiện mỗi lúc một màn hình. Nếu màn hình lệnh không còn ở đó nữa thì có thể quay lại bằng cách ấn một phím bất kỳ.

Khi màn hình lệnh đã quay lại thì một tiêu đề đồ thị, nhãn trục **X** và nhãn trục **Y**, và các dòng lưới có thể đặt vào hình vẽ bằng cách nhập liên tục vào các lệnh

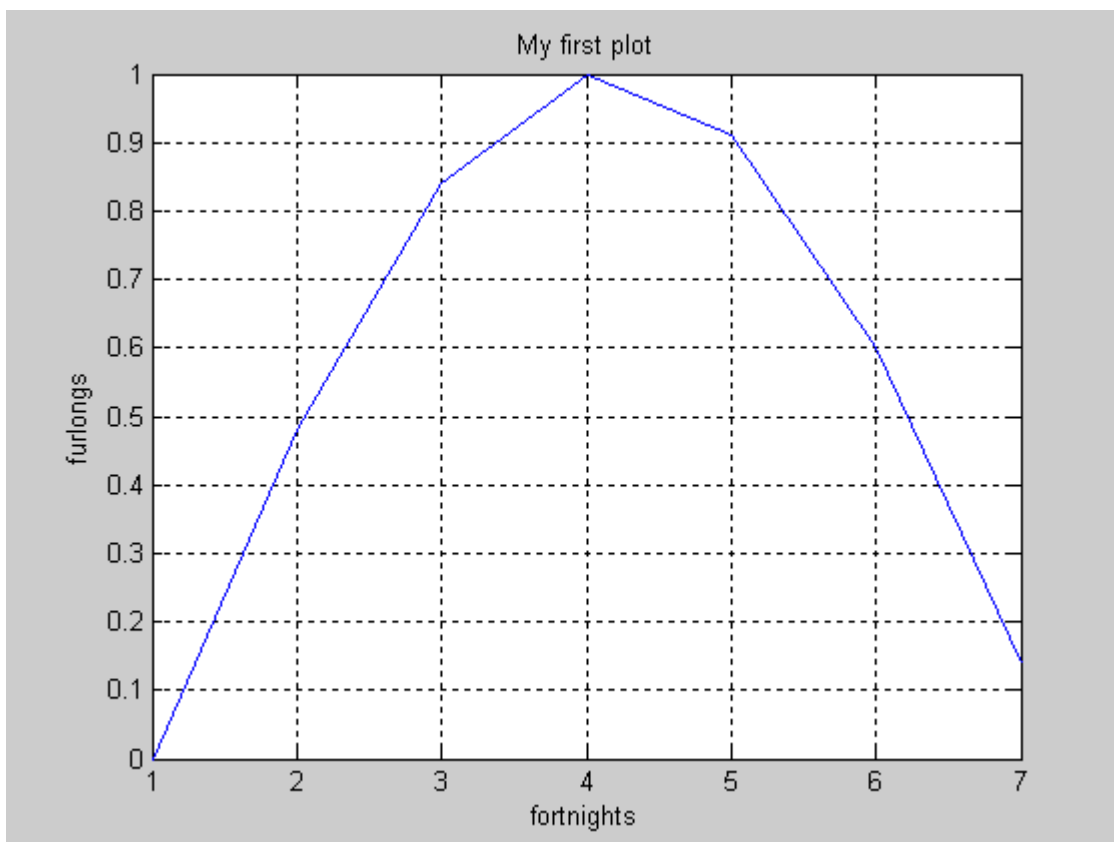
```
title('My first plot')
```

```
xlabel('fortnights')
```

```
ylabel('furlongs')
```

```
grid
```

Kết quả là hình 9.2.



Hình 9.2

Hàm **gtext('text')** cho phép chuột hoặc các phím mũi tên định vị bằng một dấu chữ thập trên đồ thị, ở điểm mà văn bản sẽ đặt khi có phím hoặc nút chuột được nhấn.

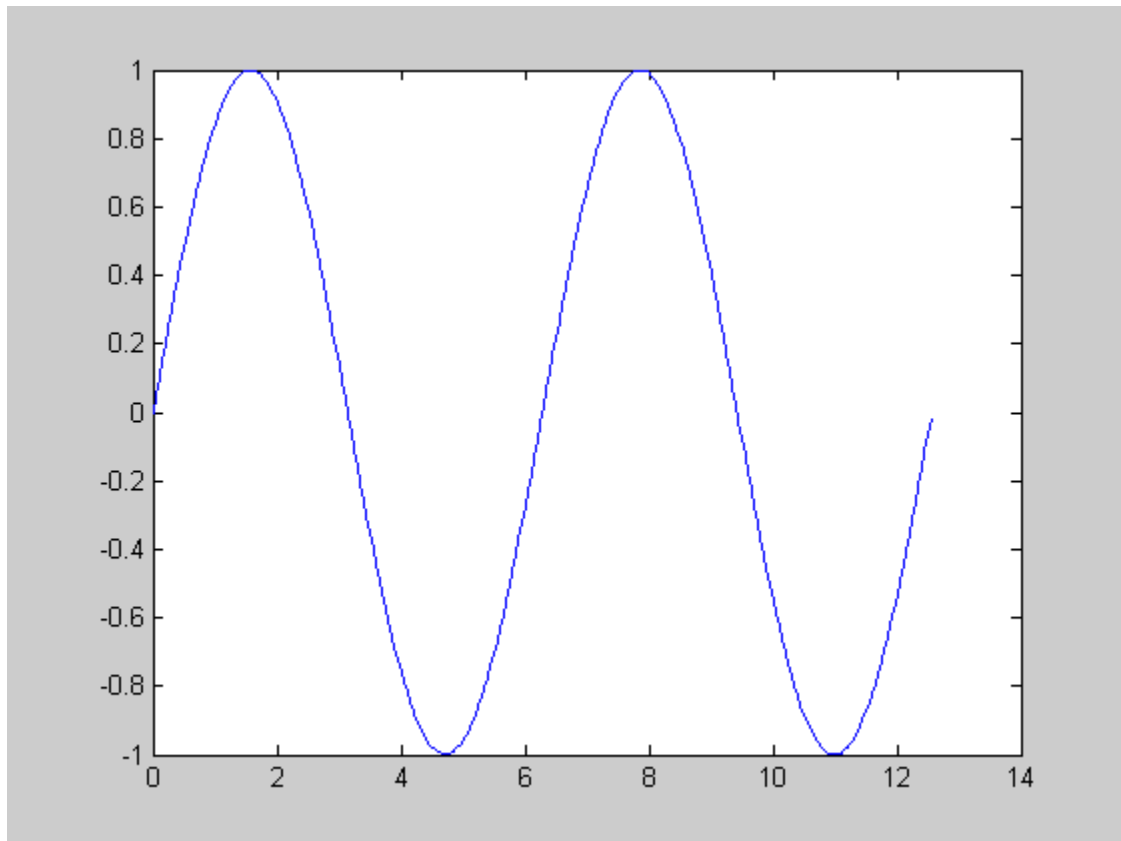
Nếu **X** và **Y** là các vector cùng độ dài, thì lệnh **plot(X,Y)** vẽ hình vẽ **x-y** gồm các phần tử của **X** đối số là các phần tử của **Y**. Ví dụ,

```
t = 0:.05:4*pi;
```

```
y = sin(t);
```

```
plot(t,y)
```

Kết quả là hình 9.3.



Hình 9.3

9.3. Nhiều đường

Có hai cách để vẽ nhiều đường trên một đồ thị đơn. Thứ nhất là cho lệnh **plot** với 2 đối số, như **plot(X,Y)**, ở đây hoặc là **X**, hoặc là **Y**, hoặc là cả hai là ma trận. Sau đó:

[1] Nếu Y là ma trận và X là vector, thì **plot(X,Y)** vẽ liên tục các dòng hoặc các cột của Y đối số là vector X , dùng kiểu đường khác nhau cho mỗi dòng hoặc cột. Việc "định hướng" dòng hay cột của Y được chọn để có cùng số phần tử như vector X . Nếu Y là ma trận vuông thì tự chọn hướng cột.

[2] Nếu X là ma trận và Y là vector, thì các quy tắc trên được áp dụng, ngoại trừ các đường từ X được vẽ đối số là vector Y .

[3] Nếu cả X và Y là ma trận cùng cỡ, thì **plot(X,Y)** vẽ các cột của X đối số là các cột của Y .

[4] Nếu không chỉ định X , như **plot(Y)**, ở đây Y là ma trận, thì các đường được vẽ cho mỗi cột của Y đối số là chỉ số dòng.

Cách thứ hai và dễ dàng hơn để vẽ nhiều đường trên một đồ thị đơn là dùng lệnh **plot** với nhiều đối số:

plot(X1, Y1, X2, Y2, ..., Xn, Yn)

Các biến $X1, Y1, X2, Y2, \dots$ là các cặp vector. Mỗi cặp x - y được vẽ, phát sinh ra nhiều đường trên đồ thị. Phương pháp nhiều đối số có điều thuận lợi là cho phép các vector có độ dài khác nhau hiển thị trên cùng một đồ thị. Như trước đây, mỗi cặp dùng một kiểu đường khác nhau.

9.4. Kiểu đường và kiểu điểm

9.4.1. Kiểu

Kiểu đường dùng trong đồ thị có thể điều khiển nếu không thỏa mãn kiểu ngầm định. Cũng có thể vẽ điểm bằng các ký hiệu khác nhau. Ví dụ,

plot(X,Y,'x')

vẽ một hình vẽ điểm bằng cách dùng các dấu x trong khi

plot(X1,Y1,':',X2,Y2,'+')

dùng đường chấm chấm cho đường cong thứ nhất và dấu $+$ cho đường cong thứ hai. Các kiểu đường và kiểu điểm khác là:

Kiểu đường		Kiểu điểm	
đặc	-	dấu chấm	.
gạch	--	dấu cộng	+
chấm	:	dấu sao	*
chấm gạch	-.	dấu tròn	o
		dấu x	x

9.4.2. Màu

Trong hệ thống có cung cấp màu, thì màu đường và màu điểm có thể chỉ định theo cách tương tự kiểu đường và kiểu điểm. Ví dụ, các lệnh

plot(X,Y,'r')

plot(X,Y,'+g')

dùng màu đỏ cho đồ thị thứ nhất và dấu + màu xanh cho đồ thị thứ hai. Các màu khác là:

Màu	
đỏ	r
xanh lá cây	g
xanh nước biển	b
trắng	w
đen	k

Nếu thiết bị phần cứng không cung cấp màu, thì các màu khác nhau trên màn hình làm cho các kiểu đường vẽ sẽ khác nhau.

9.5. Dữ liệu ảo và phức

Khi đối số của lệnh **plot** là phức (có phần ảo khác không), thì phần ảo được bỏ qua ngoại trừ khi **plot** được cho một đối số phức đơn. Đối với trường hợp đặc biệt này, thì kết quả là hình vẽ tất của hàm phần thực đối số là phần ảo. Do đó lệnh **plot(Z)**, khi **Z** là một vectơ phức hoặc ma trận phức thì tương đương lệnh **plot(real(Z),imag(Z))**.

Để vẽ nhiều đường trong mặt phẳng phức thì không có cách vẽ tất, và các phần thực và ảo phải chỉ định rõ ràng.

9.6. Hình vẽ loga, cực, và biểu đồ

Cách dùng các lệnh **loglog**, **semilogx**, **semilogy**, và **polar** là giống như lệnh **plot**. Các lệnh này cho phép dữ liệu được vẽ theo các kiểu khác nhau, nghĩa là trong các hệ tọa độ khác nhau:

- **polar(theta, rho)** là hình vẽ trong hệ tọa cực của góc **theta**, theo đơn vị **radian**, đối số là bán kính **rho**. Sau đó dùng lệnh **grid** để vẽ các lưới cực.
- **loglog** là hình vẽ dùng đơn vị chia trục **log₁₀ - log₁₀**.

- **semilogx** là hình vẽ dùng đơn vị chia trục bán loga. Trục **x** là **log10** trong khi trục **y** là tuyến tính.
- **semilogy** là hình vẽ dùng đơn vị chia trục bán loga. Trục **y** là **log10** trong khi trục **x** là tuyến tính.

Lệnh **bar(x)** hiển thị biểu đồ thanh của các phần tử của vectơ **x**, lệnh **bar** không chấp nhận nhiều đối số. Tương tự, nhưng bỏ qua các đường đứng là lệnh **stairs**, cho ra hình vẽ bậc thang là hữu ích cho việc vẽ biểu đồ hệ dữ liệu mẫu.

9.7. *Vẽ mặt lưới 3 chiều và đường mức*

Lệnh **mesh(Z)** tạo ra hình vẽ phối cảnh 3 chiều của các phần tử trong ma trận **Z**. Mặt lưới được xác định bởi các tọa độ **Z** của các điểm bên trên lưới chữ nhật trong mặt phẳng **x-y**. Hình vẽ được định dạng bằng cách nối các điểm kề nhau bằng các đoạn thẳng.

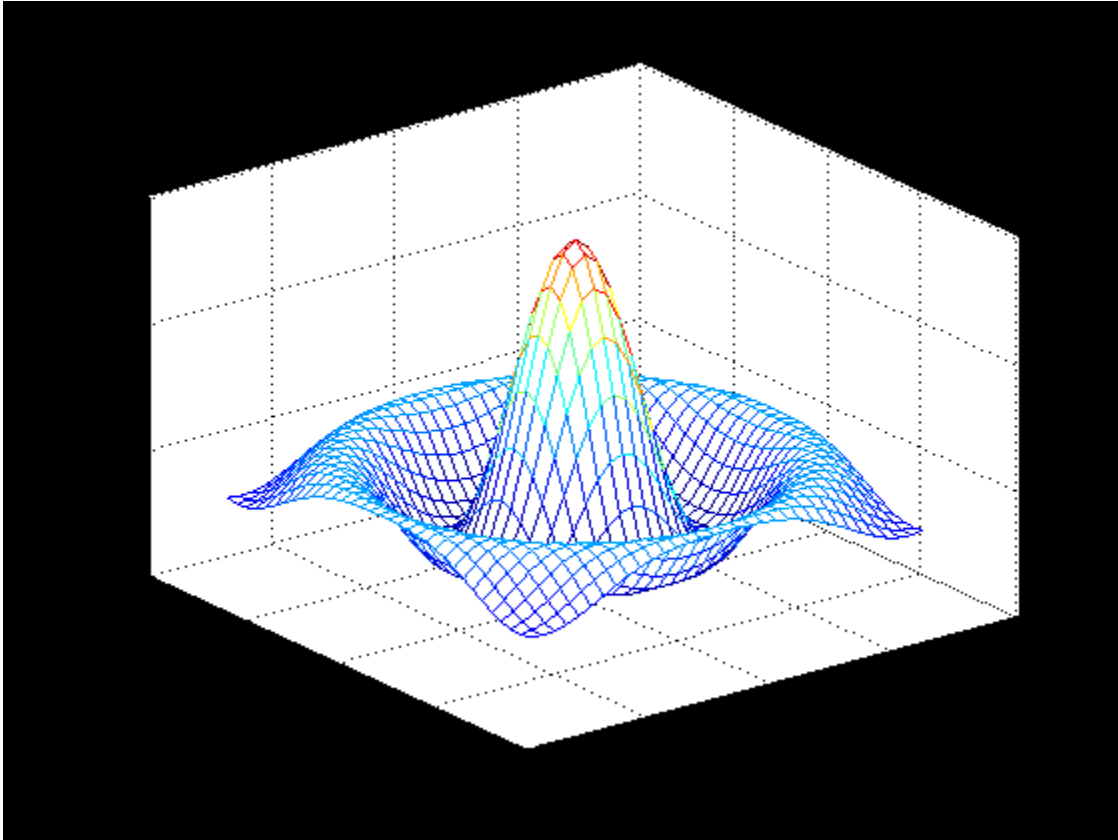
Lệnh **mesh** có thể dùng để xem các ma trận lớn, mà nếu in ra ở dạng thức số thì quá lớn. Nó cũng có thể dùng để vẽ các hàm hai biến.

Bước thứ nhất trong việc hiển thị hàm hai biến **z = f(x,y)** là phát sinh các ma trận **X** và **Y** gồm các dòng và các cột lặp tương ứng trên miền giá trị của hàm. Sau đó hàm có thể được tính toán trực tiếp và vẽ.

Xét hàm **sin(r)/r** hay **sinc** mà kết quả là mặt mũ phốt rộng vành mà mọi người ưa nhìn Một cách tạo ra là:

```
x = -8:.5:8;
y = x';
X = ones(size(y))*x;
Y = y*ones(size(x));
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;
mesh(Z)
```

Lệnh thứ nhất xác định miền giá trị **x** mà trên đó hàm được ước lượng. Lệnh thứ ba tạo ra ma trận **X** gồm các dòng lặp. Sau khi phát sinh ma trận **Y** tương ứng, ma trận **R** được tạo ra chứa khoảng cách từ tâm của ma trận, đó là gốc. Việc định dạng hàm **sinc** và áp dụng lệnh **mesh** kết quả là hình 9.4.



Hình 9.4

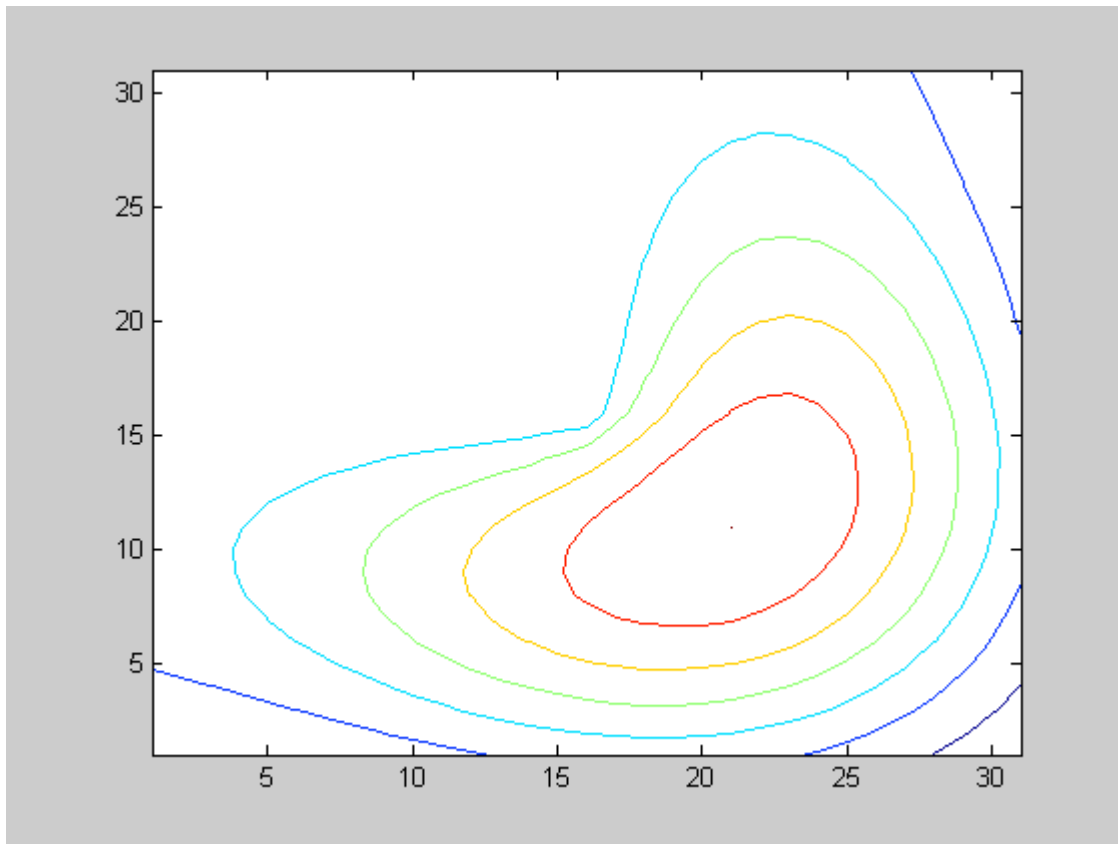
Một ma trận đơn vị trông như mặt lưới gì ? Hãy thử lệnh **mesh(eye(14))**. Với một phương pháp dễ dàng phát sinh các ma trận đặc biệt **X** và **Y** đòi hỏi để ước lượng hàm hai biến, xem lệnh **meshdom** trong phần tham khảo.

Xen vào vẽ lưới là vẽ đường mức để xem nội dung của ma trận. Một đường mức của mảng dạng **L** trong sách hướng dẫn này là

```
z = membrane(1, 15, 9, 2);
```

```
contour(z)
```

Kết quả là hình 9.5.



Hình 9.5

9.8. Điều khiển màn hình

MATLAB có 2 màn hình, một cửa sổ đồ họa và một cửa sổ lệnh. Cấu hình phần cứng có thể cho phép cả hai màn hình thấy được đồng thời trên 2 cửa sổ khác nhau, hoặc có chỉ cho phép thấy mỗi lúc một cửa sổ. Một số lệnh dùng để chuyển qua lại giữa 2 cửa sổ, và/hoặc xóa các cửa sổ theo yêu cầu:

shg	Hiện cửa sổ đồ họa
any key	Quay ngược lại cửa sổ lệnh
clc	Xóa cửa sổ lệnh
clg	Xóa cửa sổ đồ họa
home	Đưa con trỏ lệnh về đầu dòng

Ví dụ, nếu trong lúc **MATLAB** đang làm việc mà chỉ có màn hình lệnh trên màn hình, thì vào lệnh **shg** sẽ gọi lại hình vẽ cuối cùng đã vẽ trên màn hình đồ họa.

Ngầm định, với cấu hình phần cứng không hiển thị cả hai màn hình lệnh và màn hình đồ họa đồng thời sẽ, tạm dừng trong chế độ vẽ sau khi vẽ xong và chờ ấn phím.

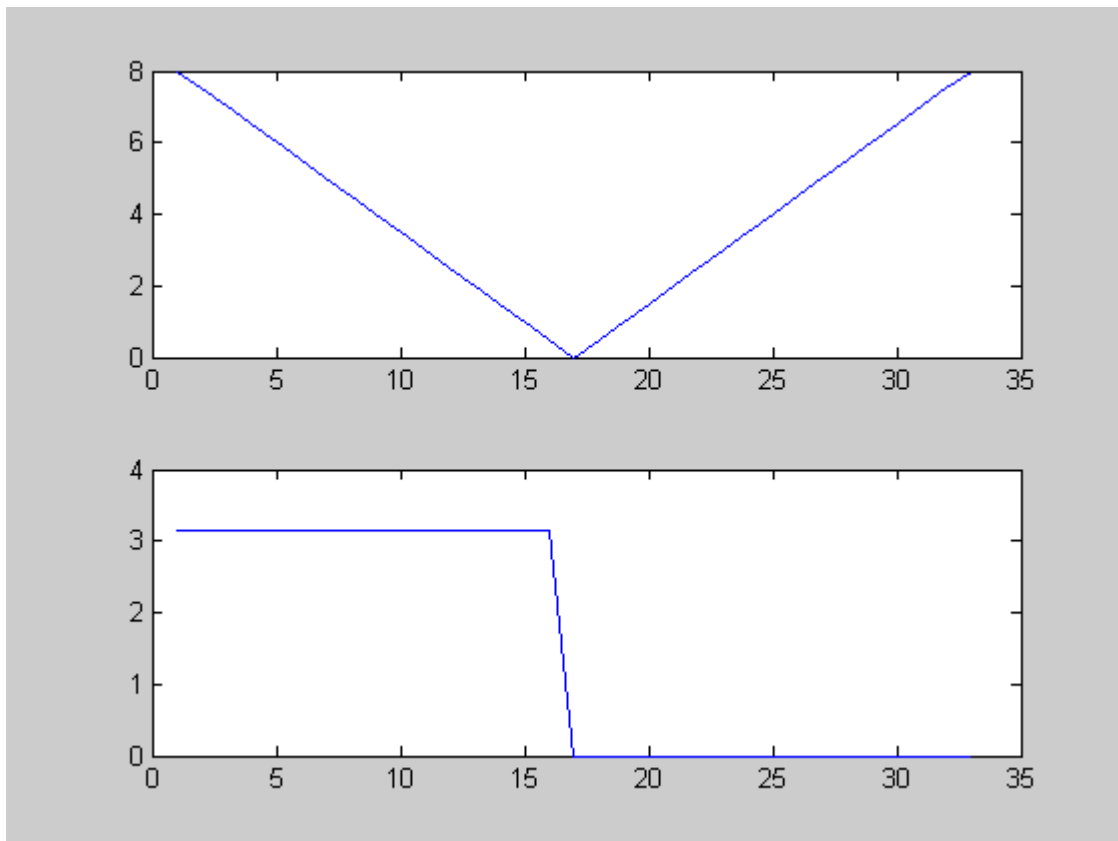
Có thể tách cửa sổ đồ họa thành nhiều phần, nhằm để xem một số hình vẽ cùng một lúc. Lệnh **subplot(m,n,p)** cắt cửa sổ đồ họa thành $m \times n$ lưới và dùng hộp thứ p cho hình vẽ tiếp sau. Ví dụ,

subplot(2, 1, 1), plot(abs(y))

subplot(2, 1, 2), plot(angle(y))

cắt màn hình thành hai, vẽ độ dài của vector phức trong nửa trên, và vẽ góc pha trong nửa dưới. Xem hình 9.6.

Lệnh **subplot(1, 1, 1)**, hoặc đúng **subplot**, trở về cửa sổ đơn ngầm định là toàn màn hình.



Hình 9.6

9.9. Cách chia đơn vị trục tọa độ

Trong trường hợp nào đó, có thể muốn đề lên đặc tính chia trục ngầm định của lệnh vẽ và chọn giới hạn vẽ. Việc thực hiện lệnh **axis**, chính nó giữ lại cách chia đơn vị trục hiện thời cho các hình vẽ sau. Vào lệnh **axis** lần nữa tiếp tục chia tự động. Hàm **axis** trả về vector dòng gồm 4 phần tử chứa **[x_min, x_max, y_min, y_max]** từ hình vẽ cuối cùng. Lệnh **axis(V)**, ở đây **V** là vector 4 phần tử, đặt cách chia trục vào các giới hạn chỉ định.

Dùng lần thứ hai lệnh **axis** là để điều khiển tỉ lệ phân giải của hình vẽ trên màn hình. Lệnh **axis('square')** đặt vùng vẽ trên màn hình là hình vuông. Với tỉ số phân giải vuông thì một đường thẳng có hệ số góc 1 đúng là 45 độ, không bị lệch bởi hình dáng không đều của màn hình. Cũng vậy, các đường tròn, như **plot(sin(t),cos(t))**, giống đường tròn thay vì đường ô-van. Lệnh **axis('normal')** đặt tỉ lệ phân giải về giá trị chuẩn.

Lệnh **hold** giữ hình vẽ hiện thời trên màn hình. Lệnh **plot** tiếp theo sẽ thêm vào hình vẽ, dùng các giới hạn vẽ trục đã thiết lập và duy trì các đường cong đã vẽ trước đó. Lệnh **hold** vẫn còn hiệu lực cho đến khi được gọi lại.

9.10. Bản sao phân cứng

Ba lệnh **prtsc**, **print** và **meta**, cung cấp các khả năng về phân cứng nói chung: **prtsc prtsc** khởi động liệt kê cửa sổ đồ họa, như **Shift-Prtsc**, và cho phép thực hiện trong tệp M-file hoặc trong vòng lặp **for**. Nói chung, kết quả này theo hình vẽ phân giải thấp, vì điểm ảnh trên màn hình chuyển thành điểm ảnh trên máy in.

meta meta mở một siêu tệp đồ họa, dùng tên tệp chỉ định, và ghi hình vẽ hiện thời vào đó để xử lý về sau. Lệnh **meta** tiếp sau nối vào tên tệp đã chỉ định trước đó. Siêu tệp có thể được xử lý sau đó, dùng các chương trình xử lý hình vẽ.

print print đưa bản sao phân giải cao của hình vẽ hiện thời ra máy in. Một số máy có giới hạn về bộ nhớ thì lệnh này không thực hiện được.

Cách dễ nhất để nhận hình vẽ trên màn hình trên mọi máy tính cá nhân là giữ phím **Shift** và ấn phím **Prtsc**. Hoạt động này đưa hình ảnh trong màn hình đồ họa ra máy in.

Xem phần đầu về đặc tả máy của sách hướng dẫn này để biết thêm thông tin về cách nhận bản sao phân cứng.

Chương 10. ĐIỀU KHIỂN LUỒNG

MATLAB có các lệnh điều khiển luồng như đã tìm thấy trong hầu hết các ngôn ngữ máy tính. Các lệnh điều khiển luồng đưa **MATLAB** sang cấp độ khác máy tính tay, cho phép nó được dùng như một ngôn ngữ bậc cao về ma trận.

10.1. Vòng lặp *FOR*

MATLAB có phiên bản riêng của nó về vòng lặp "**DO**" hoặc "**FOR**" tìm thấy trong các ngôn ngữ máy tính. Nó cho phép một câu lệnh, một nhóm lệnh, được lặp lại một số lần cố định xác định trước. Ví dụ

```
for i = 1:n, x(i) = 0, end
```

gán **0** vào n phần tử đầu của **x**. Nếu n nhỏ hơn **1** thì lệnh vẫn hợp pháp, nhưng câu lệnh bên trong không được thực hiện. Nếu **x** chưa có, hoặc có ít hơn n phần tử thì không gian thêm vào được tự động phân phối.

Có thể tổ hợp các vòng lặp và thường thụt vào để dễ đọc.

```
for i = 1:m
    for j = 1:n
        A(i,j) = 1/(i+j-1);
    end
end
A
```

Dấu chấm phẩy cuối câu lệnh bên trong vòng lặp để hủy việc in lặp ra màn hình, trong khi lệnh **A** sau vòng lặp hiển thị kết quả cuối cùng.

Một điểm quan trọng là: mỗi vòng lặp **for** phải gắn với từ khóa **end**. Nếu đơn giản vào lệnh

```
for i = 1:n, x(i) = 0
```

thì hệ thống sẽ kiên nhẫn chờ nhập các lệnh còn lại trong thân vòng lặp. Không có gì xảy ra đánh vào **end**.

Một ví dụ khác, giả sử

t =

-1

0

1

3

5

và muốn phát sinh một ma trận **Vandermonde**, ma trận có các cột là lũy thừa các phần tử của **t**.

A =

1 -1 1 -1 1

0 0 0 0 1

1 1 1 1 1

81 27 9 3 1

625 125 25 5 1

ở đây vòng lặp kép rõ ràng nhất.

n = max(size(t));

for j = 1:n

for i = 1:n

A(i,j) = t(i)^(n-j);

end

end

Nhưng vòng lặp đơn với các phép toán trên vector có ý nghĩa hơn và cũng minh họa cho vấn đề vòng lặp **for** có thể chạy ngược.

A(:,n) = ones(n,1);

for j = n-1:-1:1

A(:,j) = t .* A(:,j+1);

end

Dạng tổng quát của vòng lặp **for** là

for v = expression

statements

end

expression đúng là một ma trận, vì đúng là trong **MATLAB**. Các cột của ma trận được gán từng cột vào biến **v** và rồi các lệnh **statements** được thực hiện. Một cách rõ ràng hơn của việc hoàn thành cùng công việc này là

E = expression;

[m,n] = size(E);

for j = 1:n

v = E(:,j);

statements

end

Thông thường **expression** là loại như **m:n**, hoặc **m:i:n**, đó là ma trận chỉ có một dòng, và bởi vậy các cột của nó đơn giản là các đại lượng vô hướng. Trong trường hợp đặc biệt này, vòng lặp **for** giống như các vòng lặp "**FOR**" hay "**DO**" của các ngôn ngữ lập trình khác.

10.2. Vòng lặp *WHILE*

MATLAB cũng có phiên bản về vòng lặp "**WHILE**", cho phép một lệnh hoặc nhóm lệnh lặp lại với số lần không xác định, dưới điều khiển của một điều kiện logic. Sau đây là bài toán đơn giản minh họa cho vòng lặp **while**. Số nguyên **n** đầu tiên là số nào để **n!** (**n** giai thừa) là một số gồm **100** chữ số? Vòng lặp **while** sau đây sẽ tìm ra nó. Nếu chưa biết câu trả lời thì có thể chạy các lệnh này

n = 1;

while(prd(1:n)<1.e100, n = n+1; end

n

Một minh họa tính toán có tính ứng dụng cao hơn về vòng lặp **while** là tính hàm mũ của một ma trận, trong **MATLAB** gọi là **expm(A)**. Một định nghĩa có thể có của hàm mũ là chuỗi lũy thừa,

$$\text{expm}(\mathbf{A}) = \mathbf{I} + \mathbf{A} + \mathbf{A}^2/2! + \mathbf{A}^3/3! + \dots$$

Lý do để dùng cách tính toán thực sự này là với số phần tử của **A** không lớn lắm. ý tưởng này là tính tổng nhiều hạng tử của chuỗi này là cần thiết để cho ra kết quả không đổi nếu một số hạng tử nữa thêm vào độ chính xác số học của máy. Trong vòng lặp sau, **A** là ma trận đã cho, **E** sẽ thành lũy thừa mong muốn, **F** là hạng tử riêng trong chuỗi, và **k** là chỉ số của hạng tử đó. Các lệnh trong vòng lặp được lặp lại cho đến khi **F** là quá nhỏ để việc thêm nó vào **E** không làm thay đổi giá trị của **E**.

```
E = zeros(A);

F = eye(A);

k=1;

while norm(E+F-E,1)>0

    E = E + F;

    F = A*F/k;

    k = k+1;

end
```

Thay vào đó nếu muốn tính mảng hoặc mũ từng phần tử **exp(A)** thì đúng là phải thay đổi giá trị khởi tạo của **F** từ **eye(A)** thành **ones(A)** và thay đổi tích ma trận **A*F** thành tích mảng **A.*F**.

Dạng tổng quát của vòng lặp **while** là

```
while expression

    statements

end
```

Các lệnh **statements** được thực hiện lặp khi tất cả các phần tử trong ma trận biểu thức **expression** khác không. Ma trận biểu thức thường dùng nhất là một biểu thức quan hệ **1-1**, bởi vậy khác không tương ứng với **TRUE**. Khi ma trận

biểu thức không phải là một đại lượng vô hướng thì có thể thu gọn nó lại bằng các hàm **any** và **all**.

10.3. Các lệnh **IF** và **BREAK**

Sau đây là cặp ví dụ minh họa cho lệnh **if** của **MATLAB**. Đầu tiên trình bày cách tính có thể rơi vào **3** trường hợp, tùy thuộc vào dấu và tính chẵn lẻ của **n**.

```
if n<0
    A = negative(n)
elseif rem(n,2) == 0
    A = even(n)
else
    A = odd(n)
end
```

Ví dụ thứ hai là một bài toán hấp dẫn trong lý thuyết số. Lấy một số nguyên dương bất kỳ. Nếu chẵn thì chia cho **2**; nếu lẻ thì nhân **3** cộng **1**. Lặp lại tiến trình trên cho đến khi số nguyên bằng **1**. Vấn đề lý thú chưa được chứng minh là: Có số nguyên nào để tiến trình không được kết thúc không ? Chương trình **MATLAB** minh họa cho các câu lệnh **while** và **if**. Cũng trình bày hàm **input** - nhắc người dùng nhập dữ liệu từ bàn phím, và lệnh **break** - cung cấp một việc nhảy ra ngoài vòng lặp.

% Bài toán cổ "3n+1" trong lý thuyết số.

```
while 1
    n = input(' Nhập n, âm để thoát. ');
    if n<=0, break, end
    while n >1
        if rem(n,2)==0
            n = n/2
        else
```



```
        n = 3*n+1
    end;
end
end
```

Có thể chương trình này chạy mãi mãi.

Chương 11. SIÊU TỆP M-FILE

NGUYÊN BẢN VÀ HÀM

MATLAB thường dùng chế độ dòng lệnh; khi nhập một dòng lệnh đơn thì **MATLAB** thực hiện ngay lập tức và hiển thị kết quả. **MATLAB** cũng có khả năng thực hiện một dãy các lệnh lưu trong một tệp. Hai chế độ này tạo thành một môi trường thông dịch.

Các tệp chứa các lệnh của **MATLAB** gọi là siêu tệp M-file vì chúng có tên mở rộng là ".m" (".m" là lựa chọn của **Macintosh**). Ví dụ, một tệp tên là **bessel.m** có thể chứa các lệnh của **MATLAB** để tính các hàm **Bessel**.

Một M-file gồm một dãy các lệnh chuẩn của **MATLAB**, có thể chứa các tham chiếu đến các M-file khác. Một M-file có thể gọi đệ quy đến chính nó.

Một cách dùng M-file là một dãy dài tùy ý các lệnh. Các tệp như thế gọi là các tệp nguyên bản. Một kiểu thứ hai của M-file cung cấp khả năng mở rộng **MATLAB**. Gọi là tệp hàm, chúng cho phép các hàm mới thêm vào các hàm đã có. Nhiều tính năng của **MATLAB** nhận được từ khả năng này để tạo ra các hàm mới để giải các bài toán do người dùng chỉ định.

Cả hai kiểu M-file, nguyên bản và hàm, là các tệp văn bản ASCII bình thường, và được tạo ra bằng cách dùng một trình soạn thảo văn bản hay trình xử lý từ, tùy chọn.

11.1. Tệp nguyên bản

Khi nguyên bản được gọi, **MATLAB** đơn giản thực hiện các lệnh trong tệp, thay cho việc đợi nhập từ bàn phím. Các lệnh trong tệp nguyên bản thực hiện toàn cục trên dữ liệu trong vùng làm việc. Các nguyên bản thường hữu ích cho việc vận hành các phân tích, giải toán, hoặc thực hiện các thiết trí đòi hỏi quá nhiều lệnh mà trở nên cồng kềnh trong chế độ tương tác.

Ví dụ, giả sử các lệnh của **MATLAB**

```
% Một M-file để tính các số Fibonacci
```

```
f = [1 1]; i = 1;
```

```
while f(i) + f(i+1) < 1000
```

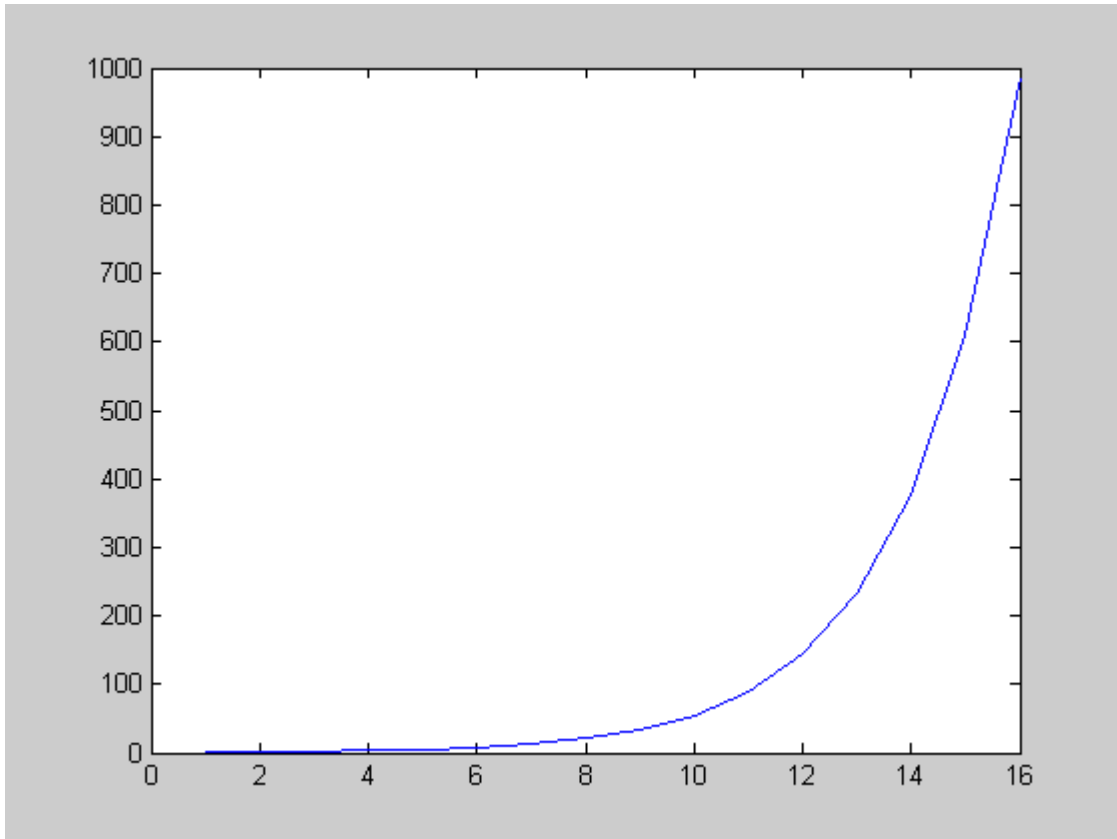
```
    f(i+2) = f(i) + f(i+1);
```

```
    i = i+1;
```

end

plot(f)

được chứa trong một tệp tên là **fibno.m**. Vào lệnh **fibno** làm cho **MATLAB** thực hiện các lệnh, tính **16** số **Fibonnaci** đầu tiên và tạo ra hình vẽ như hình 11.1.



Hình 11.1

Sau khi thực hiện tệp xong, các biến **f** và **i** còn lại trong vùng làm việc.

Các chương trình mẫu của **MATLAB** là các ví dụ tốt cho cách sử dụng các M-file để thực hiện nhiều nhiệm vụ phức tạp hơn. Tên nguyên bản **startup.m** được tự động thi hành khi **MATLAB** được gọi. Các hằng vật lý, các thừa số chuyển đổi kỹ thuật, hoặc các thứ khác muốn định nghĩa trước trong vùng làm việc có thể đặt trong các tệp này. Trên hệ thống mạng hoặc nhiều người dùng, thì có một nguyên bản tên **matlab.m** được dành riêng để dùng cho quản lý hệ thống. Nó có thể dùng để cài đặt các định nghĩa và các thông điệp rộng rãi.

11.2. Tệp hàm

Nếu dòng thứ nhất của một M-file chứa từ "**function**", thì tệp là một tệp hàm. Một hàm khác với một nguyên bản là có thể truyền các đối số, và các biến định nghĩa và thực hiện bên trong tệp là cục bộ của hàm và không thao tác toàn cục trong vùng làm việc. Các tệp hàm là hữu ích cho việc mở rộng **MATLAB**, đó là tạo ra các hàm **MATLAB** mới bằng cách dùng chính ngôn ngữ **MATLAB**.

Sau đây là một ví dụ đơn giản. Tệp **mean.m** chứa các lệnh:

```
function y = mean(x)

% MEAN  Giá trị trung bình. Đối với vector , MEAN(x)

% trả về giá trị trung bình. Đối với ma trận, MEAN(x) là một
%vector dòng chứa các giá trị trung bình của mỗi cột.
[m,n] = size(x);

if m== 1

        m = n;      % xử lý vector dòng.

end

y = sum(x)/m;
```

Tệp này định nghĩa một hàm mới tên là **mean**. Hàm mới **mean** được dùng như mọi hàm **MATLAB** khác. Ví dụ, nếu **Z** là một vector gồm các số từ 1 đến 99,

```
Z = 1:99;
```

giá trị trung bình tìm thấy bằng cách đánh vào

```
mean(Z)
```

kết quả là **ans =**

```
50
```

Hãy xét vài chi tiết của **mean.m**:

- Dòng thứ nhất khai báo tên hàm, các đối số nhập, và các đối số xuất. Không có dòng này thì tệp sẽ là tệp nguyên bản thay vì tệp hàm.
- Dấu % biểu hiện phần còn lại của dòng là lời chú thích và được bỏ qua.

- Vài dòng đầu cung cấp tư liệu M-file và được hiển thị nếu đánh vào **help mean**.
- Các biến **m**, **n**, và **y** là cục bộ của **mean** và sẽ không còn trong vùng làm việc khi **mean** thực hiện xong. (Hoặc nếu trước đó đã có thì không bị thay đổi.)
- Không cần phải đặt các số nguyên từ **1** đến **99** vào biến **x**. Thực ra, dùng **mean** với biến tên là **Z**. Vector **Z** chứa các số nguyên từ **1** đến **99** được truyền hoặc sao chép vào **mean** ở đây nó trở thành một biến cục bộ tên là **x**.

Một phiên bản có một ít phức tạp hơn của **mean** gọi là **stat** tính độ lệch chuẩn:

```
function [mean, stdev] = stat(x)

[m,n] = size(x);

if m==1

    m = n;    % xử lý vector dòng

end

mean = sum(x)/m;

stdev = sqrt(sum(x.^2)/ m - mean.^2);
```

stat minh họa cho khả năng trả về nhiều đối số xuất.

Một hàm tính hạng ma trận dùng nhiều đối số nhập:

```
function r = rank(y,tol)

% hạng của một ma trận

s = svd(x);

if (nargin == 1)

    tol = max(size(x)) * s(1) * eps;

end

r = sum(s>tol);
```

Ví dụ này minh họa cách dùng biến thường xuyên **nargin** để tìm số đối số nhập. Biến **nargout**, mặc dù không được dùng ở đây nhưng chứa số đối số xuất.

Vài gợi ý trợ giúp :

Khi một tệp M-hàm được gọi lần đầu thì được biên dịch và đưa vào bộ nhớ. Sau đó có thể sử dụng cho các lần gọi sau mà không biên dịch lại. Nó còn trong bộ nhớ trừ khi không đủ bộ nhớ, trong trường hợp này có thể bị xóa tự động.

Lệnh **what** trình bày danh sách thư mục các tệp M-file có thể sử dụng trong thư mục hiện hành, lệnh **type** liệt các tệp M-file, và **!** dùng để gọi trình soạn thảo, cho phép tạo ra hoặc sửa đổi tệp M-file.

Nói chung, nếu nhập tên nào đó cho **MATLAB**, ví dụ đánh vào **whoopie**, thì **MATLAB** thông dịch qua các bước sau:

- [1] Tìm xem **whoopie** có phải là một biến.
- [2] Kiểm tra **whoopie** có phải là hàm cài sẵn.
- [3] Tìm trong thư mục hiện hành có không một tệp có tên **whoopie.m**.
- [4] Tìm trong các thư mục chỉ định bởi biến môi trường

MATLABPATH có không một tệp có tên **whoopie.m**. (Xem phần giới thiệu cách cài đặt để học cách đặt biến môi trường **MATLABPATH**)

Do đó đầu tiên **MATLAB** thử dùng **whoopie** như một biến, nếu có, trước khi dùng **whoopie** như một hàm.

11.3. Các lệnh *Echo, input, pause, keyboard*

Thông thường, khi thực hiện M-file, các lệnh trong tệp không được hiển thị trên màn hình. Lệnh **echo** làm cho tệp M-file được thấy khi thực hiện, điều này hữu ích cho việc gỡ rối hoặc làm mẫu. Xem phần tham khảo để biết thêm chi tiết.

Hàm **input** nhận dữ liệu nhập từ người dùng. Lệnh

n = input('Có bao nhiêu quả táo')

cho người dùng câu văn bản nhắc, đợi người dùng nhập số hoặc biểu thức từ bàn phím. Một cách dùng **input** là xây dựng M-file điều khiển **menu**. Công cụ **demo** là một ví dụ cho trường hợp này.

Tương tự **input**, nhưng mạnh hơn, là hàm **keyboard**. Hàm này gọi bàn phím như một nguyên bản. Đặt trong các tệp M-file, thì đặc tính này giúp ích cho việc gỡ rối, hoặc cho việc thay đổi các biến trong thời gian thi hành.

Lệnh **pause** tạo ra thủ tục dừng và chờ người dùng ấn phím bất kỳ trước khi tiếp tục. Lệnh **pause(n)** tạm dừng *n* giây trước khi tiếp tục.

Cũng có thể định nghĩa các biến toàn cục, mặc dù chúng tôi không khuyến khích. Xem phần tham khảo nếu có ý muốn.

11.4. *Xâu chữ và macro chuỗi*

Các chuỗi văn bản được nhập vào **MATLAB** trong cặp nháy đơn. Ví dụ,
s = 'Hello'

kết quả là

```
s =  
  
Hello
```

Xâu chữ được lưu trong một vector, mỗi phần tử một ký tự. Trong trường hợp này, lệnh

```
size(s)  
  
ans =  
  
1    5
```

biểu hiện rằng **s** có **5** phần tử. Các ký tự được lưu giá trị ASCII của chúng và hàm **abs** trình bày giá trị này,

```
abs(s)  
  
ans =  
  
72   101   108   108   111
```

Hàm **setstr** đặt các vector để hiển thị như văn bản thay vì trình bày các giá trị ASCII. Cũng hữu ích là lệnh **disp** đơn giản hiển thị văn bản có trong biến, và các hàm **isstr** và **stremp** dò tìm và so sánh các chuỗi tương ứng.

Các biến văn bản có thể nối lại thành chuỗi lớn bằng cách dùng cặp ngoặc vuông:

```
s = [s, 'World']
```

```
s =
```

```
Hello World
```

Các giá trị số được chuyển sang các chuỗi bằng các hàm **sprintf**, **num2str**, và **int2str**. Các giá trị số sau khi chuyển sang chuỗi thường được nối vào chuỗi lớn để đặt tiêu đề cho hình vẽ có giá trị số:

```
f = 70; c = (f-32)/1.8;
```

```
title(['Nhiệt độ trong phòng là ', num2str(c), ' độ C'])
```

eval là hàm làm việc với các biến chuỗi để cài đặt một công cụ **macro** văn bản khá mạnh mẽ. **eval(t)** làm cho văn bản chứa trong **t** được ước lượng. Nếu **STRING** là văn bản nguồn cho nhiều biểu thức hoặc câu lệnh của **MATLAB** thì

```
t = 'STRING';
```

mã hóa văn bản trong **t**. Đánh vào **t** in văn bản và **eval(t)** làm cho văn bản được thông dịch, hoặc là một lệnh hoặc là một nhân tử trong biểu thức. Ví dụ

```
t = '1/(i+j-1)';
```

```
for i = 1:n
```

```
    for j = 1:n
```

```
        a(i,j) = eval(t);
```

```
    end
```

```
end
```

phát sinh ma trận **Hilbert** cấp **n**. Một ví dụ khác trình bày văn bản đánh chỉ số,

```
S = ['x = 3          ';
```

```
     'y = 4          ';
```

```
     'z = sqrt(x*x+y*y) '];
```

```
for k = 1:3
```

```
    eval(S(k,:));
```


end

Các chuỗi tạo thành các dòng của ma trận **A** cần phải có cùng độ dài. Sau đây là ví dụ cuối cùng trình bày cách **eval** có thể dùng lệnh **load** để nạp 10 tệp dữ liệu được đánh số liên tục:

```
fname = 'mydata';
for i = 1:10
    eval(['load ',fname,int2str(i)]
end
```

Công cụ macro văn bản được ứng dụng hữu hiệu trong việc truyền tên hàm cho các tệp M-hàm. Để lấy ví dụ, xem tệp **funm.m** trong **MATLAB TOOLBOX**.

11.5. Chương trình bên ngoài

Có thể, và thường hữu ích, để tạo ra các chương trình độc lập bên ngoài riêng của mình hoạt động như các hàm **MATLAB** mới. Điều này có thể thực hiện bằng cách viết các tệp M-file để

- [1] Lưu các biến trên đĩa,
- [2] Chạy các chương trình bên ngoài (đọc các tệp dữ liệu, xử lý chúng, và ghi kết quả trở lại đĩa), và
- [3] Nạp các tệp đã xử lý ngược về vùng làm việc.

Ví dụ, sau đây là một M-hàm giả định để tìm lời giải phương trình **Garfield** dùng chương trình **GAREQN** bên ngoài

```
function y = garfield(a,b,q,r)
save gardata a b q r
!gareqn
load gardata
```

Nó yêu cầu đã viết một chương trình tên là **GAREQN** (bằng **Fortran** hoặc ngôn ngữ nào đó) để đọc tệp tên là **gardata.mat**, xử lý nó, và đặt kết quả trở ra tệp đó. Các chương trình con tiện ích mô tả trong phần sau có thể dùng để đọc và ghi các tệp **MAT**.

Công cụ này là một trong các lựa chọn để "liên kết chương trình riêng" vào **MATLAB**. Một lựa chọn khác là dùng công cụ tệp **MEX** - một kỹ thuật nhờ đó có thể liên kết vật lý chương trình có đối tượng mới vào **MATLAB**. Xem phần đặc tả máy để thấy công cụ này có thể dùng cho máy mình không.

11.6. Vấn đề về tốc độ và bộ nhớ

Các thao tác về vector và ma trận gắn liền của **MATLAB** thực hiện nhanh hơn các thao tác được dịch của nó. Điều này có nghĩa là để nhận tốc độ nhanh nhất ngoài **MATLAB** phải cố gắng vector hóa thuật toán trong tệp M-file. Bất kỳ đâu có thể được, các vòng lặp **for** và **while** nên chuyển sang các phép toán về vector hoặc ma trận. Ví dụ, một cách lấy **sin** của **1000** số từ **1** đến **10**:

```
i = 0;

for t = 0:.01:10

    i = i+1;

    y(i) = sin(t);

end
```

Một phiên bản vector hóa của cùng chương trình này là:

```
t = 0:.01:10;

y = sin(t);
```

Trên một máy, ví dụ thứ nhất chạy hết **15** giây, trong khi ví dụ thứ hai chỉ tốn **0.6** giây, nhanh gấp **25** lần. Không phải luôn luôn tối ưu được các chương trình phức tạp, nhưng khi tốc độ là quan trọng thì nên tìm cách vector hóa thuật toán.

Nếu không thể vector hóa mảnh chương trình, thì đây là một cách để làm cho vòng lặp **for** chạy nhanh hơn: *định vị trước mọi vector có kết quả xuất được lưu*. Ví dụ, việc đưa vào câu lệnh thứ nhất ở đây, dùng hàm **zeros**, làm cho vòng lặp **for** thực hiện nhanh đáng kể:

```
y = zeros(1,100);

for i = 1:100

    y(i) = det(x^i);

end
```

Lý do là, nếu không định vị trước, thì **MATLAB** phải tăng kích thước của vector **y** lên **1** qua mỗi lần lặp. Nếu vector được định vị trước thì bước này được khử đi và việc thực hiện được nhanh hơn.

Đối với công việc thực hiện với các ma trận lớn trên các máy có bộ nhớ hạn chế, thì ý đồ định vị trước có một tiện lợi thứ hai: là sử dụng bộ nhớ hiệu lực hơn và giúp kiểm tra có chạy tràn bộ nhớ không. Nó trợ giúp vì bộ nhớ có khuynh hướng bị phân mảnh, bởi vậy có thể có nhiều vùng nhớ tự do, nhưng không đủ không gian liên tục để giữ một biến lớn. Việc định vị trước giúp thu gọn sự phân mảnh.

Trong vấn đề về bộ nhớ, nếu lệnh **who** hiển thị tổng số bộ nhớ tự do còn lại, thì có vài điều về số này có lẽ nên cẩn thận. Nếu xóa một biến trong vùng làm việc, thì con số hiển thị bởi lệnh **who** thường không tăng lên, *trừ khi nó là biến "cao nhất" trong vùng làm việc*.

Con số này biểu hiện thực sự tổng bộ nhớ tự do liên tục và chưa dùng. Việc xóa biến cao nhất làm cho bộ nhớ lớn hơn, nhưng xóa biến dưới biến cao nhất không có hiệu lực. Về mặt ứng dụng, toàn bộ ý nghĩa này là có thể có nhiều vùng nhớ tự do hơn lệnh **who** biểu hiện.

Các máy tính với bộ nhớ ảo không hiển thị tổng số vùng nhớ tự do còn lại vì không có các giới hạn phải chấp nhận của **MATLAB** hay của phần cứng.

Có một cách tối ưu mà **MATLAB** thực hiện giúp để biết khi viết M-file. Các đối số gọi hàm M-file không sao chép sang vùng làm việc cục bộ của hàm trừ khi xen vào nội dung của đối số vào bên trong M-hàm. Điều này có nghĩa là không có không tồn bộ nhớ cho việc truyền các biến lớn vào các hàm M-file.

Chương 12. VỀ TỆP TRÊN ĐĨA

load và **save** là các lệnh của **MATLAB** để lưu vào hoặc lấy ra từ đĩa nội dung của vùng làm việc. Các lệnh khác quan hệ đến tệp giúp cho việc quản lý đĩa, cho phép các chương trình bên ngoài chạy, và cung cấp khả năng nhập/xuất dữ liệu.

12.1. Thao tác về tệp

Các lệnh **dir**, **type**, **delete**, và **chdir** cài đặt tập hợp các lệnh về hệ điều hành chung để thao tác về tệp. Sau đây là bảng biểu hiện các lệnh này sắp xếp với các hệ điều hành khác, có lẽ người dùng làm quen với một trong chúng:

MATLAB	MS-DOS	UNIX	VAX/VMS
dir	dir	is	dir
type	type	cat	type
delete	del	rm	delete
chdir	chdir	cd	set default

Với hầu hết các lệnh này, đường dẫn, ký tự đại diện, và tên ổ đĩa dùng theo cách thông thường.

Lệnh **type** khác với lệnh **type** thông thường ở một điểm đặc biệt; nếu không cho kiểu tệp thì ngầm định là **.m**. Điều này thuận tiện cho việc hay dùng nhất của lệnh **type** là để liệt kê các tệp M-file trên màn hình.

Lệnh **diary** tạo ra nhật ký cho công việc của **MATLAB** trên đĩa (tuy nhiên không lưu các hình ảnh). Kết quả là tệp văn bản ASCII phù hợp với việc đưa vào các bản báo cáo và các tài liệu khác dùng trình xử lý từ bất kỳ.

Để biết thêm chi tiết về các lệnh này, xem phần tham khảo hoặc dùng công cụ trợ giúp nóng **help**.

12.2. Chạy chương trình bên ngoài

Ký tự chấm than, **!**, là ký tự thoát và biểu hiện phần còn lại của dòng nhập là lệnh của hệ điều hành. Điều này hoàn toàn có ích cho việc gọi các trình tiện ích hoặc chạy các chương trình khác mà không ra khỏi **MATLAB**. Ví dụ

!f77 simpleprog

gọi trình biên dịch Fortran và

!edt darwin.m

gọi trình soạn thảo edt cho một tệp có tên là darwin.m. Sau khi chương trình này chạy xong, quyền điều khiển trả về cho MATLAB.

Cách xử lý đúng đắn về ! tùy thuộc vào từng loại máy cụ thể. Xem phần đặc tả máy để biết thêm thông tin.

12.3. Nhập và xuất dữ liệu

Dữ liệu từ các chương trình khác và bên ngoài có thể đưa vào **MATLAB** bằng nhiều cách. Tương tự, dữ liệu **MATLAB** có thể xuất ra bên ngoài. Cũng có thể có các chương trình thao tác dữ liệu trực tiếp trong các tệp **MAT**, dạng tệp **MATLAB** sử dụng.

Cách tốt nhất phụ thuộc vào số lượng dữ liệu đang có, dữ liệu có ở dạng máy có thể đọc được không, dạng gì, v.v... Sau đây là một số lựa chọn; hãy chọn một để tương thích.

[1] Nhập như một danh sách rõ ràng các phần tử. Nếu ít dữ liệu, nhỏ hơn **10x15** phần tử, thì dễ dàng nhập trực tiếp dữ liệu vào bằng cặp ngoặc vuông, [và]. Phương pháp này bất tiện đối với dữ liệu lớn vì không thể sửa dữ liệu nhập nếu bị lỗi.

[2] Tạo trong một M-file. Dùng trình soạn thảo văn bản để tạo ra một nguyên bản M-file chứa danh sách rõ ràng các phần tử. Cách này là tốt khi dữ liệu chưa ở dạng máy có thể đọc được và phải đánh chúng vào. Cơ bản là giống cách 1, có thuận tiện là cho phép dùng trình soạn thảo để thay đổi dữ liệu hoặc sửa lỗi. Sau đó có thể chạy lại M-file để nhập lại dữ liệu.

[3] Nạp từ một tệp ASCII phẳng. Nếu dữ liệu được lưu ở dạng ASCII, với các dòng có độ dài cố định kết thúc bằng ký tự sang dòng, và các khoảng trống ngăn cách các số, thì tệp như thế được gọi là tệp phẳng. (Tệp ASCII phẳng có thể được soạn bằng trình soạn thảo văn bản thông thường.) Các tệp phẳng có thể được đọc trực tiếp vào **MATLAB** bằng lệnh **load**. Kết quả được đặt vào một biến có tên là tên tệp.

[4] Viết một chương trình bằng **Fortran** hoặc **C** để dịch dữ liệu sang dạng **MAT**-file.

Vài cách đưa dữ liệu ra bên ngoài là:

[1] Với các ma trận nhỏ, dùng lệnh **diary** để tạo ra một tệp nhật ký, rồi sau đó liệt kê các biến trong tệp này. Có thể dùng soạn thảo văn bản để thao tác trên tệp nhật ký sau này. Việc xuất dữ liệu của lệnh **diary** dùng trong suốt thời

gian làm việc của **MATLAB**, nó có ích cho việc đưa dữ liệu vào tài liệu và các bản báo cáo.

[2] Lưu vào một biến bằng lệnh **save**, với lựa chọn **/ascii**. Ví dụ,

```
A rand(4,3);  
save temp.dat A/ascii
```

tạo ra một tệp ASCII tên là **temp.dat** chứa nội dung:

```
0.2113    0.8096    0.4832  
0.0824    0.8474    0.6135  
0.7599    0.4524    0.2749  
0.0087    0.8075    0.8807
```

[3] Viết một chương trình bằng **Fortran** hoặc **C** để dịch tệp **MAT** sang dạng đặc biệt riêng của mình.

Có thể muốn có các chương trình bên ngoài đọc hoặc ghi dữ liệu trực tiếp vào tệp **MAT** đã dùng các lệnh **load** và **save**. Dạng thức của tệp **MAT** được trình bày dưới lệnh **load** trong phần tham khảo.

Nếu chương trình viết bằng **Fortran** hoặc **C**, thì có vài phục vụ cung cấp trong **MATLAB TOOLBOX** giúp giao diện chương trình với các tệp **MAT**:

savemat.for Ghi tệp **MAT**.

loadmat.for Đọc tệp **MAT**.

testls1.for Ví dụ sử dụng **savemat** và **loadmat**.

testls2.for Ví dụ khác sử dụng **savemat** và **loadmat**.

loadmat.c Nạp ma trận từ tệp **MAT**.

savemat.c Lưu ma trận vào tệp **MAT**.

testls.c Ví dụ dùng **loadmat.c** và **savemat.c**.

Việc cài đặt các phiên bản **Fortran** của các phục vụ này có thể khác nhau theo từng loại máy.

PHỤ LỤC

BỘ LỆNH VÀ HÀM

Quaán lý Lãn vai haim

HELP

Àæa ra tạiç liãûu trãu giuïp træuc tuyãún
 Lãnũh HELP liãût kã danh saïch táút cái caïc váún ããö
 chênũ. Mãúi váún ããö tæång æĩng vãi mãüt thæ muüc
 trong ãæång dãn MATLABPATH
 Lãnũh "HELP topic" cho trãu giuïp vãi <topic>. <topic>
 coĩ thãø lai mãüt tãn lãnũh hoàuc tãn thæ muüc. Nãúu
 lai tãn lãnũh thç HELP hiãøn thẽ thãng tin vãi lãnũh
 àoĩ. Nãúu lai tãn thæ muüc thç HELP hiãøn thẽ nãúi dũng
 thæ muüc. MORE ON laim cho HELP taũm dæĩng mãúi khi
 áãöy mãĩn hçnh

DOC

DOC Naũp tài liãûu hypertext (siãu vàn baĩn)
 DOC <topic> naũp tài liãûu hypertext coĩ váún ããö chè
 æẽnh lai topic
 Chênũ lãnũh DOC naũp danh saïch caïc tài liãûu
 hypertext

WHAT

Liãût kã danh saïch caïc tãũp M-file, MAT- file vai
 MEX-file
 Lãnũh WHAT liãût kã danh saïch caïc tãũp M-file, MAT-
 file vai MEX-file trong thæ muüc hiãũn hainũ.
 Lãnũh WHAT <dirname> liãût kã caïc tãũp trong thæ muüc
 <dirname> trãn ãæång dãn MATLABPATH. Khãng cáon cho
 ãæång dãn áãöy áuí

TYPE

Hiãøn thẽ nãúi dũng mãüt tãũp vàn baĩn
 TYPE foo.bar xem nãúi dũng tãũp foo.bar
 TYPE foo xem nãúi dũng tãũp foo.m

LOOKFOR

LOOKFOR Tçm caïc ããö muüc HELP
 LOOKFOR XYZ tçm chuãúi XYZ trong doĩng chuĩ thẽch áãöu
 tiãn cuĩa vàn baĩn HELP trong táút cái caïc tãũp M-file
 tçm trãn ãæång dãn MATLABPATH. Khãng cáon cho ãæång
 dãn áãöy áuí.
 WHAT liãût kã caïc haim trong mãüt thæ muüc, WHICH tçm
 thæ muüc chæĩa haim chè æẽnh, vai LOOKFOR tçm táút cái
 caïc haim trong táút cái caïc thæ muüc mãĩ coĩ tại chè
 æẽnh

WHICH

Tçm caïc haim vai tãũp
 WHICH FCN hiãũn ãæång dãn áãöy áuí cuĩa haim ãæũc
 chè æẽnh baĩi FCN. Haim coĩ thãø lai haim M-file, MEX-
 file, áãö hoĩa SIMULINK, hoàuc haim chuáon. Caïc haim
 chuáon vai haim SIMULINK thç ãæũc hiãũn mãüt thãng
 baĩo cho biãút lai haim chuáon hay lai haim thainũ
 pháon cuĩa SIMULINK

Làm màu vài khái niệm của MATLAB, châu chảnh
trên Expo của MATLAB

Kiểu tra bảng dẫn tọc kiểu của MATLAB
Lệnh PATH hiểu bảng dẫn tọc kiểu hiểu hẻnh của
MATLAB. Bảng dẫn tọc kiểu là biểu mải trểng
MATLABPATH trong hủi iểu hẻnh hoặc iểu iểu bải
MATLABRC, vài cỏi thỏ chẻ iểnh bải STARTUP
P = PATH trái vể mủt chuểi chẻi bảng dẫn trong P
PATH(P) iểi bảng dẫn sang P
PATH(P1,P2) iểi bảng dẫn sang P1 vài P2. Do iểi
PATH(PATH,P) nủi mủt thẻ mủc mải vào bảng dẫn
hiểu hẻnh vài PATH(P,PATH) cho mủt bảng dẫn mải
bẻng cẻch nủi P vào trểi bảng dẫn củ. Nủu
P1 hoặ P2 iểi cỏi trẻn bảng dẫn thẻ khẻ thẻ
vẻo

Liăût kă caïc biăún hiăún hăinh
WHO liăût kă caïc biăún trong vùng làm viăúc hiăún
hăinh
WHOS hiăøn thê nhiăõu thăng tin hăn văõ măùi biăún
WHO GLOBAL vài WHOS GLOBAL liăût kă caïc biăún trong
vùng làm viăúc toàin cuê

Hiãøn thẽ thăng tin ááy âuí hãn vảo cáic biảún hiãũn hảnh
WHOS laì ðaũng ááy âuí cuía WHO, liãũt kã cáic biảún
trong vuõng laìm viãũc hiãũn hảnh cũng vảo thăng tin
vảo kễch thẩic, máũt áũũ, áũũ phẩic taũp cuía chuõng,
...

```
Láúy caïc biăún tại âéa
LOAD <fname> láúy caïc biăún tại tấű MAT-file
'fname.mat'
Lăűnh LOAD nấű tại tấű 'matlab.mat'
LOAD <xxx.yyy> âôűc tấű ASCII xxx.yyy, nôi phái
chăiă mấű máing chăi nhấű caïc sắ liăű sắű xắű
thănh m đoiğ, mắi đoiğ gắm n giăi trề. Kắű quă
lăi mắű ma trắű câi mxn tắ lăi xxx Ăăű nấű mắű
tấű ASCII mắ khăng cắ tắ mắ rắűg thặ đuiğ lăűnh
LOAD <fname> -ascii. Ngoăi ra, MATLAB thắ văo phắű
mắ rắűg '.mat' văi nấű nôi nhă mắű tấű MAT-file.
Ăăű nấű tấű MAT-file mắ khăng cắ phắű mắ rắűg
'.mat' thặ đuiğ lăűnh LOAD <fname.ext> -mat. Nắű tắ
tấű lăi "stdio" thặ LOAD âôűc tại thiắű bắ nhắű
chắűn
```

SAVE

Lưu các biến vào áe

SAVE <fname> lưu tất cả các biến vào tệp nh
phân "MAT-file" coi tên <fname.mat>.

Đe liêu coi thâo lúy lúi vâi lúnh LOAD. Náu boi qua tên tệp thc tên tệp ngóm ênh lai "matlab.mat"

SAVE fname X chề lưu biến X

SAVE fname X Y Z lưu X, Y, và Z

SAVE fname X Y Z -ascii dùng đùng ASCII 8 chấ sắ thay cho đùng nhề phân

SAVE fname X Y Z -ascii -double dùng đùng ASCII 8 chấ sắ

SAVE fname X Y Z -ascii -double -tabs cách nhau các khoảng tab

Náu tên tệp lai "stdio" thc SAVE gắi đê liêu ra thiút bề xuút chuáon. Các đùng nhề phân dùng trong tệp MAT-file phu thuâc vào kêch thắc vài kiău của ma trậ. Các ma trậ nôi vài ma trậ coi sắ khắ nguyă âêúc lưu trong đùng chắu âêung 8 byte/sắ. Các ma trậ lăn hoâc nguyă coi thâo lưu trong đùng cả âoùngchề cáon1, 2 hoâc 4 byte/ sắ. Xem các phu vủ C vài Fortran vâo âoüc/ghi các tệp

MAT-file tặ các chắng trệh bắ ngoài

CLEAR

Xoă các loaúi khắc nhau trong vùng làm viăc

CLEAR xoă tất cả các biến trong vùng làm viăc

CLEAR VARIABLES giắng CLEAR

CLEAR X xoă biến hoâc hàm X

CLEAR FUNCTIONS xoă tất cả các hàm biă dềch M-file

CLEAR MEX xoă tất cả các liă kắ tệp MEX-file

CLEAR ALL xoă tất cả các biến , các hàm vài các liă kắ MEX

Náu X toăi cuê thc CLEAR X xoă X trong vùng làm viăc hiău hănh, nhềg coi thâo truy cáúp ái các hàm khai bắo nôi toăi cuê

CLEAR GLOBAL X xoă biến toăi cuê X

CLEAR GLOBAL xoă tất cả các biến toăi cuê

PACK

Hắp nhắu vùng nhắ làm viăc

PACK thêc hiău viăc táúp hắp bắ nhắ vung vài.

Các phắn MATLAB mắ rắng coi thâo làm cho bắ nhắ bề phân nhiău mắnh, ngă cáin viăc lưu các biến lăn. PACK lai lúnh lưu tất cả các biến vào áe, Xoă bắ nhắ, rắi nắp lúi các biến

SIZE

Lúy kêch thắc ma trậ

D = SIZE(X), vâi ma trậ X câ MxN thc trái vâ vectắ đòng D = [M, N] chắa sắ đòng vài sắ cắ của ma trậ X

[M,N] = SIZE(X) sắ đòng vài sắ cắ của ma trậ X táich thănh 2 biến

M = SIZE(X,1) trái vảo ầuĩng sấu ðoĩng
 N = SIZE(X,2) trái vảo ầuĩng sấu cẩt

LENGTH

Láúy sấu thàĩnh pháõn củá vectả
 LENGTH(X) trái vảo ầũ ðàĩ củá vectả X. Tẻng ầẻng
 MAX(SIZE(X))

DISP

Hiẩn mẩt mả trẩn nhẻ vản bảĩn
 DISP(X) hiẩn mả trẩn mảĩ khẻng ĩn rả tẻn mả trẩn.
 Nẩũ X chẻĩ vản bảĩn thẻ vản bảĩn ầẻũ hiển thẻ

Làm viẩc vẩĩ tẩp vại hẩ ầĩẩũ hẻĩn**CD**

Thẻ ầẻĩ thẻ mủc hiẩn hẻĩn
 CD <dir> ầũt thẻ mủc hiẩn hẻĩn lại <dir>
 CD .. chuyển vảo thẻ mủc chẻ
 Lẩĩn CD ầũ xem thẻ mủc hiẩn hẻĩn

DIR

Xẻ danh sảĩh cảĩ tẩp trỏĩ mẩt thẻ mủc
 DIR <dir> liểt kả cảĩ tẩp trỏĩ thẻ mủc <dir>. Cỏĩ
 thẩ ðủĩng ầẻĩng ðẩn vại kỷ tẻũ ầũĩ ðẩĩn

DELETE

Xỏĩ mẩt tẩp hẻ mẩt ầũĩ tẻũĩng ầũ hỏĩ
 DELETE <file_name> xỏĩ tẩp <file_name> trẻn ầẻ
 DELETE(H) xỏĩ ầũĩ tẻũĩng ầũ hỏĩ cỏĩ thẻĩ lại H.
 Nẩũ ầũĩ tẻũĩng lại mẩt cẻĩ sẩ thẻ cẻĩ sẩ ầẻũ
 ầẻĩng

GETENV

Láúy gẻĩ trẻ mảĩ trẻĩĩng
 GETENV('NAME'), NAME lại chũũĩ vản bảĩn, tẻm danh
 sảĩh mảĩ trẻĩĩng vẩĩ chũũĩ cỏĩ ðẩĩng NAME=VALUE vại
 trái vảo chũũĩ VALUE nẩũ cỏĩ chũũĩ nhẻ thẩũ. Nẩũ
 khẻng tẻm thẩũĩ tẻn chẻ ầẻĩ thẻ trái vảo mẩt mả
 trẩn rẩĩĩng

Đẩũ ! ầũ thẻ hẻĩn mẩt lẩĩĩn củá hẩ ầĩẩũ hẻĩn

DIARY

Lẻ vản bảĩn củá pháõn MATLAB
 DIARY <file_name> tầũ mẩt bảĩn sỏ cảĩ ðẩĩ liểũ
 nhẩũp/xuẩt gẻĩ vại tẩp <file_name>
 DIARY OFF huỷ nỏĩ. DIARY ON ngẻũĩ lẩĩ
 Lẩĩĩn DIARY lẩũ trẩĩĩng thẻĩ hiẩn thẻĩĩ

Ầĩẩũ khiểĩn cẻĩ sẩ lẩĩĩn**CLC**

Xỏĩ cẻĩ sẩ lẩĩĩn vại ầẻ cỏn trỏĩ vảo ầũũ cẻĩ sẩ

HOME

Âa con troí vảo ááou cáia sảo

FORMAT

Àaút đaùng thăic xuát

Táút cáic cáic tênh toaĩn trong MATLAB áaăuc thăuc hiăũn áĩ áăũ chênħ xăic kếp

Lăũnh FORMAT coi thăo đùng thay áăoi đaùng thăic hiăũn thê nhê sau:

FORMAT giăúng nhê SHORT vai
lại ngăom âênħ

FORMAT SHORT đaùng đăúu cháúm ténħ văĩi
5 chăĩ săũ

FORMAT LONG đaùng đăúu cháúm ténħ văĩi 15
chăĩ săũ

FORMAT SHORT E đaùng đăúu cháúm áăũng văĩi
5 chăĩ săũ

FORMAT LONG E đaùng đăúu cháúm áăũng văĩi 15
chăĩ săũ

FORMAT HEX đaùng thăúp luúc phán

FORMAT + cáic kyĩ hiăũu +, - vai đăúu
căich ĩn ra áăúi văĩi cáic phăon
tăĩ đăang, áĩ, vai 0. Phăon áio
áaăuc boĩ qua

FORMAT BANK đaùng áă_la vai xu

FORMAT COMPACT boĩ cáic kyĩ tăũ sang đoiğ

FORMAT LOOSE áaút vaio cáic kyĩ tăũ sang
doiğ

FORMAT RAT xăúp xê bàong đaùng
phán săũ cho cáic săũ nguyăĩn nhói

ECHO

Âa ra cáic lăũnh trong tăũp nguyăĩn baĩn

ECHO ON báũt viăũc áa ra cáic lăũnh trong tăũp nguyăĩn
baĩn

ECHO OFF tăũt viăũc áa ra cáic lăũnh trong tăũp nguyăĩn
baĩn

ECHO file ON , áĩ ááy 'file' lại tăĩn haĩm laĩm cho tăũp
haĩm áa ra khi đùng noi

ECHO file OFF tăũt noi . ECHO file láũt traũng thaĩi
hiăũn thăĩi

ECHO ON ALL báũt viăũc áa ra cáic lăũnh trong táút cáic
cáic tăũp nguyăĩn baĩn. ECHO OFF ALL tăũt noi

MORE

Áiăũ khiăũn xuát tăĩng trang trong cáia sảo lăũnh

MORE OFF huĩy xuát tăĩng trang trong cáia sảo lăũnh

MORE ON cho pheĩp xuát tăĩng trang trong cáia sảo
lăũnh

MORE(N) chê âênħ kêch thăăĩc trang lại N đoiğ. Ngăom
âênħ lại OFF vai N = 23

Khi MORE cho pheĩp vai xuát tăĩng trang thê áũn ↵ áăo
áăũn đoiğ tiăũp theo, áũn <spacebar> áăo sang trang
tiăũp. Áũn phêm "q" áăo thoaĩt áăo muúc hiăũn thê

Thăng tin chung

INFO

Thăng tin vảo MATLAB và MathWorks

INFO <toolboxpath> hiều các tập Readme của toolbox
vảỉ àềỉng dầỉn chè àềỉh

WHATSNEW

WHATSNEW Hiều các tập của MATLAB và các
toolbox

WHATSNEW <toolboxpath> hiều các tập Readme của
toolbox vảỉ àềỉng dầỉn chè
àềỉh

VER

Sầỉ phiểỉn baỉn của MATLAB

VER hiểỉn thề sầỉ phiểỉn baỉn của MATLAB và toolbox
hiểỉn thầỉỉ

VER(TBX) hiểỉn thề thắỉg tin phiểỉn baỉn hiểỉn thầỉỉ
của toolbox chè àềỉh bàỉng chuầỉ TBX

Caïc hàm Logic

EXIST

Kiểm tra biểu thức logic hàm có

EXIST('A') trả về:

- 0 nếu A không có
- 1 nếu A là một biểu thức trong vùng làm việc
- 2 nếu A là một M-file trong đường dẫn tìm kiếm của MATLAB
- 3 nếu A là một MEX-file trong đường dẫn tìm kiếm của MATLAB
- 4 nếu A là hàm SIMULINK có sẵn
- 5 nếu A là hàm của MATLAB

Lưu ý: EXIST('A') cũng trả về 2 nếu một tập dữ liệu 'A' có trong đường dẫn của MATLAB

ANY

Trả về một phần tử bất kỳ của vectơ (khác 0)

Với vectơ thực ANY(V) trả về 1 nếu một phần tử bất kỳ của vectơ khác 0. Ngược lại trả về 0

Với ma trận thực ANY(X) thực hiện trên từng cột của X, trả về vectơ dòng gồm các số 1 và 0

ALL

Trả về tất cả các phần tử của vectơ đều

Với vectơ thực ALL(V) trả về 1 nếu tất cả các phần tử của vectơ khác 0. Ngược lại trả về 0

Với ma trận thực ANY(X) thực hiện trên từng cột của X, trả về vectơ dòng gồm các số 1 và 0

FIND

Tìm số của các phần tử khác 0

I = FIND(X) trả về các số của các phần tử khác 0. Ví dụ, I = FIND(A>100), trả về số của các phần tử lớn hơn 100 của A

[I,J] = FIND(X) trả về số dòng và cột của các phần tử khác 0 trong ma trận X. Thường dùng với ma trận thưa

[I,J,V] = FIND(X) cũng trả về số dòng và cột của các phần tử khác 0 trong ma trận X. Lưu ý rằng find(X) và find(X~=0) sẽ cho ra cùng I và J, nhưng sau đó cho ra V với tất cả các số 1

ISNAN

Trả về một phần tử số (NaN, Not-A-Number như 0/0)

ISNAN(X) trả về 1 với các phần tử của X là NaN và 0 nếu khác

Ví dụ ISNAN([pi NaN Inf -Inf]) là [0 1 0 0]

ISINF

Trả về các phần tử không xác định (1/0 chẵn) (hầu)

ISINF(X) trái vảo 1 vảoi caic phảo n tẻi cuía X laì +Inf
 hoảuc -Inf vaì 0 nảu khaic
 Vẻ duu, isinf([pi NaN Inf -Inf]) laì [0 0 1 1]

FINITE

Ấuỉg vảoi caic phảo n tẻi xaic ảẻnh
 FINITE(X) trái vảo 1 vảoi caic phảo n tẻi cuía X laì
 xaic ảẻnh vaì 0 nảu khaic ảẻnh
 Vẻ duu, finite ([pi NaN Inf -Inf]) laì [1 0 0 0]

ISEMPTY

ISEMPTY Ấuỉg vảoi caic ma trỏu rỏuỉg
 ISEMPTY(X) trái vảo 1 nảu X laì mỏt ma trỏu rỏuỉg
 vaì 0 nảu khaic rỏuỉg. Mỏt ma trỏu rỏuỉg coỉ kẻch
 thỏic 0 ẻt nhỏt 1 chỏỏu, nhỏ 0x0, 0x5, ...

ISSPARSE

ISSPARSE Ấuỉg nảu laì ma trỏu thỏa
 ISSPARSE(S) laì 1 nảu lỏp lỏu trỏi cuía S laì thỏa
 vaì 0 nảu ngỏỏc lỏuỉ

ISSTR

Ấuỉg vảoi chuỏuỉ kyỉ tỏu
 ISSTR(S) trái vảo 1 nảu S laì mỏt chuỏuỉ kyỉ tỏu vaì
 0 nảu ngỏỏc lỏuỉ

ISGLOBAL

ISGLOBAL Ấuỉg vảoi caic biỏỏn toỏn cuỏc
 ISGLOBAL(A) laì 1 nảu A laì biỏỏn toỏn cuỏc, 0 nảu
 ngỏỏc lỏuỉ

Caic haỉm đẻi liỏu cả baỉn

MAX

Ấỏuỉ vảoi vectỏ thỏc MAX(X) laì phảo n tẻi lỏn nhỏt
 trong X. Vảoi ma trỏu thỏc MAX(X) laì vectỏ đỏuỉg chỏa
 phảo n tẻi lỏn nhỏt trỏn mỏuỉ cỏt
 [Y,I] = MAX(X) lỏu chẻ sỏu cuía caic giaỉ trẻ lỏn
 nhỏt trong vectỏ I
 MAX(X,Y) trái vảo mỏt ma trỏu củuỉg kẻch thỏic vảoi
 X vaì Y vảoi caic phảo n tẻi lỏn nhỏt lỏuỉ ra tẻi X
 hoảuc Y. Vảoi sỏu phỏic thỏc đủuỉg MAX(ABS(X))

MIN

Ấỏuỉ vảoi vectỏ thỏc MIN(X) laì phảo n tẻi nhỏỉ nhỏt
 trong X. Vảoi ma trỏu thỏc MIN(X) laì vectỏ đỏuỉg chỏa
 phảo n tẻi nhỏỉ nhỏt trỏn mỏuỉ cỏt
 [Y,I] = MIN(X) lỏu chẻ sỏu cuía caic giaỉ trẻ nhỏỉ
 nhỏt trong vectỏ I
 MIN(X,Y) trái vảo mỏt ma trỏu củuỉg kẻch thỏic vảoi
 X vaì Y vảoi caic phảo n tẻi nhỏỉ nhỏt lỏuỉ ra tẻi X
 hoảuc Y. Vảoi sỏu phỏic thỏc đủuỉg MIN(ABS(X))

MEAN

Tên gọi giải trẻ trung bệnh

Âm vị vectơ thực MEAN(X) là giải trẻ trung bệnh của các phần tử trong X. Âm vị ma trận thực MEAN(X) là vectơ dòng chứa giải trẻ trung bệnh của mỗi cột

MEDIAN

Tên gọi giải trẻ trung gian

Âm vị vectơ thực MEDIAN(X) là giải trẻ trung gian của các phần tử lần trong X. Âm vị ma trận thực MEDIAN(X) là vectơ dòng chứa giải trẻ trung gian của mỗi cột

STD

Âm lệch chuẩn

Âm vị vectơ thực STD(X) là âm lệch chuẩn. Âm vị ma trận thực STD(X) là vectơ dòng chứa âm lệch chuẩn của mỗi cột

STD tên âm lệch chuẩn "tiểu biểu", âm lệch chuẩn hoán N-1, âm vị N là âm dài của dãy

SORT

Sắp xếp tăng dần

SORT(X) sắp xếp mỗi cột theo thứ tự tăng dần. Nếu X phức thực các phần tử âm sắp xếp theo ABS(X)

[Y,I] = SORT(X) cũng trả về một ma trận chèn số I. Nếu X là vectơ thực Y = X(I). Nếu X là ma trận các m x n thực for j = 1:n, Y(:,j) = X(I(:,j),j)

SUM

Tên tổng các phần tử

Âm vị vectơ thực SUM(X) là tổng các phần tử của X. Âm vị ma trận thực SUM(X) là vectơ dòng chứa tổng của mỗi cột. SUM(DIAG(X)) là vế của X

PROD

Tên tích các phần tử

Âm vị vectơ thực PROD (X) là tích các phần tử của X. Âm vị ma trận thực PROD (X) là vectơ dòng chứa tích của mỗi cột

CUMSUM

Tên tổng tích lũy các phần tử

Âm vị vectơ thực CUMSUM(X) là tổng tích lũy các phần tử của X. Âm vị ma trận thực CUMSUM(X) là vectơ dòng chứa tổng tích lũy của mỗi cột

CUMPROD

Tên tích tích lũy các phần tử

Âm vị vectơ thực CUMPROD (X) là tích tích lũy các phần tử của X. Âm vị ma trận thực CUMPROD (X) là vectơ dòng chứa tích tích lũy của mỗi cột

TRAPZ

Tên tích phân số bằng phương pháp thang

$Z = \text{TRAPZ}(X,Y)$ tính tích phân của Y theo X bằng cách dùng tích phân hình thang. X và Y phải là các vectơ cùng chiều dài, hoặc X phải là vectơ cột và Y là ma trận cùng số dòng với X . TRAPZ tính tích phân của mỗi cột tăng dần. Kết quả Z là một array ngang và hằng hoặc là một vectơ dòng

$Z = \text{TRAPZ}(Y)$ tính tích phân hình thang Y với giá trị thiếu khoảng cách giữa các ô là 1. Tính tích phân cho các khoảng khác nhau thì nhân Z với khoảng tầng

Vi phân sai phân

DIFF

Vi phân hoặc sai phân

DIFF có thể dùng để tính các array số là số hoặc các chuỗi. Với array số là vectơ số thì DIFF tính các sai phân. Với vectơ X thì $\text{DIFF}(X)$ là $[X(2)-X(1) \quad X(3)-X(2) \quad \dots \quad X(n)-X(n-1)]$

Với ma trận X thì $\text{DIFF}(X)$ là ma trận các cột sai phân $[X(2:n,:) - X(1:n-1,:)]$

$\text{DIFF}(X,n)$ là hàm sai phân bậc n

Với array số các chuỗi (chuỗi) thì DIFF dùng Symbolic Math Toolbox để tính array hàm

$\text{DIFF}(S)$ tính vi phân của S bằng với biến array là của nó

$\text{DIFF}(S,'v')$ tính vi phân của S bằng với biến $'v'$

$\text{DIFF}(S,n)$ và $\text{DIFF}(S,'v',n)$ vi phân S n lần

DIFF không tham số tính vi phân bằng chuỗi hoặc trapezoid

GRADIENT

Xấp xỉ gradient

$[PX,PY] = \text{GRADIENT}(Z,DX,DY)$ tính các array hàm số theo phương của ma trận Z

$PX = dZ/dx$ và $PY = dZ/dy$. DX và DY có thể là các hằng hoặc khoảng mẫu trong hằng của X và Y , hoặc chuỗi có thể là các vectơ hằng tất cả các vẽ trên rồi

$[PX,PY] = \text{GRADIENT}(Z)$ giá trị thiếu $DX = DY = 1$

Nếu Y là vectơ thì $\text{GRADIENT}(Y)$ và $\text{GRADIENT}(Y,DX)$ tính các array hàm số một chiều dY/dX

DEL2

Hàm array rời rạc Laplace

$V = \text{del2}(U)$ là một ma trận cùng cỡ với U với mỗi phần tử bằng sai phân giữa một phần tử của ma trận U và trung bình của bốn phần tử lân cận. Với các phần tử ở "góc" và "cạnh" thì một hoặc ba phần tử lân cận

Các thao tác với vectơ

CROSS

Tính tích hằng vectơ

$C = \text{CROSS}(A,B)$ trả về tích có hướng của các vectơ A và B . Ở đây, $C = A \times B$. A và B phải là các vectơ 3 chiều.

Nếu A và B là các ma trận cỡ $3 \times N$ thì C là ma trận cỡ $3 \times N$ chứa tích có hướng giữa các cột tương ứng của A và B

DOT

Tích vô hướng vectơ

$C = \text{DOT}(A,B)$ trả về tích vô hướng của các vectơ A và B . A và B phải là các vectơ cùng chiều dài.

Nếu A và B là các ma trận cỡ $M \times N$ (với các vectơ cột) thì C là vectơ dòng chiều dài N chứa các tích vô hướng giữa các cột tương ứng của A và B

Với các vectơ cột, thay vào đó dùng $A' * B$

Caic hâu s u t ang quan

CORRCOEFF

CORRCOEFF là h u s u t ang quan

CORRCOEFF(X) là ma trận các h u s u t ang quan ta u n n t i m ng X m i m i dòng của n i là m t quan tr c, v i m i c t là m t bi n

CORRCOEFF(X,Y) gi ng nh  CORRCOEFF([X Y])

Nếu C là ma trận ph ng sai, $C = \text{COV}(X)$, thì CORRCOEFF(X) là ma trận m i ph n t i th i (i,j) là $C(i,j) / \sqrt{C(i,i) * C(j,j)}$

COV

Ma trận ph ng sai

Nếu X là vectơ thì $\text{COV}(X)$ là m t bi n. Với ma trận thì m i dòng là m t quan tr c, v i m i c t là m t bi n, $\text{COV}(X)$ là ma trận ph ng sai.

DIAG(COV(X)) là m t vectơ các bi n cho m i c t, v i $\sqrt{\text{DIAG}(\text{COV}(X))}$ là vectơ các   u l ch ch n. $\text{COV}(X,Y)$ bằng $\text{COV}([X Y])$, với X và Y là các vectơ c t cùng   u dài

SUBSPACE

SUBSPACE là g c gi a 2 kh ng gian con

SUBSPACE(A,B) t m g c gi a 2 kh ng gian con ch    nh b i các c t của A và B . Nếu A và B là các vectơ c i   u dài   n về thì bằng $\text{ACOS}(A' * B)$

Nếu g c n i thì 2 kh ng gian g n phu  thu c tuy n t nh. Trong th  nghi m v t ly  m  t i b i v i quan tr c A và m t th c h nh th i hai của th  nghi m m  t i b i B , SUBSPACE(A,B) cho các   u    của t ng s u th ng tin m i   nh h ng b i th  nghi m th i hai kh ng quan h u với các sai s u th ng k  của các thay   i

Lo c v i t ch ch p

FILTER

Lo c s u

$Y = \text{FILTER}(B, A, X)$ lọc dữ liệu trong vectơ X với bộ lọc mà tại bất kỳ vectơ A và B sẽ tạo ra vectơ lọc dữ liệu. Bộ lọc thực hiện "Biến đổi tần số ngược của phương trình vi phân" như sau:

$$y(n) = b(1)*x(n) + b(2)*x(n-1) + \dots + b(nb+1)*x(n-nb) - a(2)*y(n-1) - \dots - a(na+1)*y(n-na)$$

$[Y, Zf] = \text{FILTER}(B, A, X, Zi)$ cho truy xuất các tần số của đầu vào và đầu ra, Zi và Zf , của thời gian giữa mẫu (trục hoành)

FILTER2

FILTER2 lọc hai chiều

$Y = \text{FILTER2}(B, X)$ lọc dữ liệu trong vectơ X với bộ lọc hai chiều trong ma trận B . Kết quả, Y , sẽ có tên bằng tên của cấp 2 chiều và cùng cái với X

$Y = \text{FILTER2}(B, X, 'shape')$ trả về Y tên bằng qua tên của cấp 2 chiều với kích thước của ảnh bất kỳ 'shape':

'same' - (nguyên ảnh) trả về phần tám của tên của cấp cùng kích thước với X

'valid' - trả về các phần của tên của cấp mà ảnh tên không coi các cạnh thăm giá trị 0, $\text{size}(Y) < \text{size}(X)$

'full' - trả về tên của cấp hai chiều đầy đủ, $\text{size}(Y) > \text{size}(X)$

CONV

Têch chấup vai têch âa thăic

$C = \text{CONV}(A, B)$ têch chấup caic vectă A vai B. Vectă kăút quai coi âău dài $\text{LENGTH}(A) + \text{LENGTH}(B) - 1$. Năúu A vai B lai caic vectă hău săú âa thăic thç têch chấup cuia chũng lai têch 2 âa thăic

CONV2

Têch chấup 2 chiăöu

$C = \text{CONV2}(A, B)$ thăuc hiăün têch chấup 2 chiăöu cuia caic ma trăün A vai B.

Năúu $[ma, na] = \text{size}(A)$ vai $[mb, nb] = \text{size}(B)$ thç $\text{size}(C) = [ma+mb-1, na+nb-1]$.

$C = \text{CONV2}(A, B, 'shape')$ trái vă phăön con cuia têch chấup 2 chiăöu văii kêch thăăic chề âênh bắi 'shape':

'full' - (ngăöm âênh) trái văo têch chấup 2 chiăöu âăöy âúi

'same' - trái văo phăön tám cuia têch chấup cũng câi văii A

'valid' - chề trái văo caic phăön cuia têch chấup

mai âăăuc tênh kăng coi caic caũnh thăm giaỉ trề 0,

$\text{size}(C) = [ma-mb+1, na-nb+1]$ năúu $\text{size}(A) > \text{size}(B)$

CONV2 nhanh nhăút khi $\text{size}(A) > \text{size}(B)$

DECONV

Tăich chấup vai chia âa thăic

$[Q, R] = \text{DECONV}(B, A)$ tăich vectă A ra khoíi vectă B.

Kăút quai trái văo trong vectă Q vai phăön còin laũi trong vectă R mai $B = \text{conv}(Q, A) + R$

Năúu A vai B lai caic vectă hău săú âa thăic thç tăich chấup lai chia âa thăic. Kăút quai cuia pheĩp chia B cho A lai thăng Q vai còin laũi R

BESSELF

Thiăút kău bău loũc tăng tău Bessel

$[b, a] = \text{besself}(n, Wn)$

$[b, a] = \text{besself}(n, Wn, 'ftype')$

$[z, p, k] = \text{besself}(\dots)$

$[A, B, C, D] = \text{besself}(\dots)$

BILINEAR

Chuyăön bău loũc tăng tău sang săú theo phăng phaĩp biăún âăöi song tuyăún tênh

$[zd, pd, kd] = \text{bilinear}(z, p, k, Fs)$

$[zd, pd, kd] = \text{bilinear}(z, p, k, Fs, Fp)$

$[numd, dend] = \text{bilinear}(num, den, Fs)$

$[numd, dend] = \text{bilinear}(num, den, Fs, Fp)$

$[Ad, Bd, Cd, Dd] = \text{bilinear}(A, B, C, D, Fs)$

$[Ad, Bd, Cd, Dd] = \text{bilinear}(A, B, C, D, Fs, Fp)$

BLACKMAN

Căia săø Blackman

$w = \text{blackman}(n)$

$w = \text{blackman}(n, sflag)$

BOXCAR

Căia săú chăi nhăút

```
w = boxcar(n)
```

BUTTAP

Nguyễn màu lọc tầng tần Butterworth

```
[z,p,k] = buttap(n)
```

BUTTER

Thiết kế bộ lọc tầng tần vai sáu Butterworth

```
[b,a] = butter(n,Wn)
[b,a] = butter(n,Wn,'ftype')
[b,a] = butter(n,Wn,'s')
[b,a] = butter(n,Wn,'ftype','s')
[z,p,k] = butter(...)
[A,B,C,D] = butter(...)
```

BUTTORD

Chọn tần bộ lọc Butterworth

```
[n,Wn] = buttord(Wp,Ws,Rp,Rs)
[n,Wn] = buttord(Wp,Ws,Rp,Rs,'s')
```

CHEB1AP

Nguyễn màu bộ lọc tầng tần Chebyshev loại I

```
[z,p,k] = cheb1ap(n,Rp)
```

CHEB1ORD

Chọn tần tần lọc Chebyshev loại 1

```
[n,Wn] = cheb1ord(Wp,Ws,Rp,Rs)
[n,Wn] = cheb1ord(Wp,Ws,Rp,Rs,'s')
```

CHEB2AP

Nguyễn màu lọc Chebyshev loại II

```
[z,p,k] = cheb2ap(n,Rs)
```

CHEB2ORD

Chọn tần tần lọc Chebyshev loại II

```
[n,Wn] = cheb2ord(Wp,Ws,Rp,Rs)
[n,Wn] = cheb2ord(Wp,Ws,Rp,Rs,'s')
```

CHEBWIN

Cửa sổ Chebyshev

```
w = chebwin(n,r)
```

CHEBY1

Thiết kế bộ lọc Chebyshev loại I

```
[b,a] = cheby1(n,Rp,Wn)
[b,a] = cheby1(n,Rp,Wn,'ftype')
[b,a] = cheby1(n,Rp,Wn,'s')
[b,a] = cheby1(n,Rp,Wn,'ftype','s')
[z,p,k] = cheby1(...)
[A,B,C,D] = cheby1(...)
```

CHEBY2

Thiết kế bộ lọc Chebyshev loại II

```
[b,a] = cheby2(n,Rs,Wn)
[b,a] = cheby2(n,Rs,Wn,'ftype')
```

```
[b,a] = cheby2(n,Rs,Wn,'s')
[b,a] = cheby2(n,Rs,Wn,'ftype','s')
[z,p,k] = cheby2(...)
[A,B,C,D] = cheby2(...)
```

CHIRP

Phaït sinh âæåìng cosine táön säú rung

```
y = chirp(t,f0,t1,f1)
y = chirp(t,f0,t1,f1,'method')
y = chirp(t,f0,t1,f1,'method',phi)
```

CONV2

Têch cháúp 2 chiãöu

C = CONV2(A, B) thæüc hiãûn têch cháúp 2 chiãöu cuía caïc ma tráûn A vaì B.

Nãúu [ma,na] = size(A) vaì [mb,nb] = size(B) thç size(C) = [ma+mb-1,na+nb-1].

C = CONV2(A,B,'shape') traí vã pháön con cuía têch cháúp 2 chiãöu vãi kêch thæäic chè æënh báíi 'shape':

'full' - (ngáöm æënh) traí vã têch cháúp 2 chiãöu ääöy äuí

'same' - traí vã pháön tám cuía têch cháúp cùìg câ vãi A

'valid' - chè traí vã caïc pháön cuía têch cháúp mà æäüc tênh khãng coì caïc caûnh thãm giaì trë 0,

size(C) = [ma-mb+1,na-nb+1] nãúu size(A) > size(B)

CONV2 nhanh nháút khi size(A) > size(B)

CONVMTX

Têch cháúp ma tráûn

```
A = convmtx(c,n)
```

```
A = convmtx(r,n)
```

Vê duû:

```
h = [1 2 3 2 1];
```

```
convmtx(h,7)
```

```
ans =
```

0	1	2	3	2	1	0	0	0	0	0
0	0	1	2	3	2	1	0	0	0	0
0	0	0	1	2	3	2	1	0	0	0
0	0	0	0	1	2	3	2	1	0	0
0	0	0	0	0	1	2	3	2	1	0
0	0	0	0	0	0	1	2	3	2	1
0	0	0	0	0	0	0	1	2	3	2
1	0	0	0	0	0	0	0	1	2	3

CPLXPAIR

Nhoìm caïc säú phæic thàình caïc càúp liãn hãúp

```
y = cplxpair(x)
```

```
y = cplxpair(x,tol)
```

CREMEZ

Thiết kế bộ lọc FIR với pha tuyến tính

```
b = cremez(n,f,'fresp')
b = cremez(n,f,'fresp',w)
b = cremez(n,f',{'fresp',p1,p2,...},w)
b = cremez(n,f,a,w)
b = cremez(...,'sym')
b = cremez(...,'skip_stage2')
b = cremez(...,'debug')
b = cremez(...,{lgrid})
[b,delta,opt] = cremez(...)
```

CZT

Chirp biến đổi Z

```
y = czft(x,m,w,a)
y = czft(x)
```

DCT

Biến đổi cosine rời rạc

```
y = dct(x)
y = dct(x,n)
```

ELLIP

Thiết kế bộ lọc Elliptic (Cauer)

```
[b,a] = ellip(n,Rp,Rs,Wn)
[b,a] = ellip(n,Rp,Rs,Wn,'ftype')
[b,a] = ellip(n,Rp,Rs,Wn,'s')
[b,a] = ellip(n,Rp,Rs,Wn,'ftype','s')
[z,p,k] = ellip(...)
[A,B,C,D] = ellip(...)
```

ELLIPAP

Nguyên mẫu bộ lọc tầng Elliptic

```
[z,p,k] = ellipap(n,Rp,Rs)
```

ELLIPORD

Chọn mẫu bộ lọc Elliptic

```
[n,Wn] = ellipord(Wp,Ws,Rp,Rs)
[n,Wn] = ellipord(Wp,Ws,Rp,Rs,'s')
```

FFT

Biến đổi Fourier nhanh 1 chiều

```
y = fft(x)
y = fft(x,n)
```

FFT2

Biến đổi Fourier nhanh 2 chiều

```
Y = fft2(X)
Y = fft2(X,m,n)
```

FFTFILT

Lọc FIR-FFT bằng phép chia

```
y = fftfilt(b,x)
y = fftfilt(b,x,n)
```

FFTSHIFT

Sàõp xăúp laûi dăi liăûu xuăút cuía hàm FFT
`y = fftshift(x)`

FILTFILT

Loïc săú pha Zero
`y = filtfilt(b,a,x)`

FILTIC

Tặm âiăöu kiăûn âáöu cho caích cài âaût băü loïc biăún
 âăøi trăuc tiăúp đăung II
`z = filtic(b,a,y,x)`
`z = filtic(b,a,y)`

FREQS

Âăp ăing táön săú cuía caïc băü loïc tăng tău
`h = freqs(b,a,w)`
`[h,w] = freqs(b,a)`
`[h,w] = freqs(b,a,n)`
`freqs(b,a)`

FREQSPACE

Khăng gian táön săú cho âăp ăing táön săú
`f = freqspace(n)`
`f = freqspace(n, 'whole')`
`[f1,f2] = freqspace(n)`
`[f1,f2] = freqspace([m n])`
`[x1,y1] = freqspace(n, 'meshgrid')`
`[x1,y1] = freqspace([m n], 'meshgrid')`

FREQZ

Âăp ăing táön săú cuía caïc băü loïc săú
`[h,w] = freqz(b,a,n)`
`[h,f] = freqz(b,a,n,Fs)`
`[h,w] = freqz(b,a,n, 'whole')`
`[h,f] = freqz(b,a,n, 'whole',Fs)`
`h = freqz(b,a,w)`
`h = freqz(b,a,f,Fs)`
`freqz(...)`

GRPDELAY

Trung bệnh cấng cuía trầu bầu loüc (nhoìm trầu)
 [gd,w] = grpdelay(b,a,n)
 [gd,f] = grpdelay(b,a,n,Fs)
 [gd,w] = grpdelay(b,a,n,'whole')
 [gd,f] = grpdelay(b,a,n,'whole',Fs)
 gd = grpdelay(b,a,w)
 gd = grpdelay(b,a,f,Fs)
 grpdelay(b,a)

HAMMING

Cáia sặ Hamming
 w = hamming(n)
 w = hamming(n,sflag)

HANNING

Cáia sặ Hanning
 w = hanning(n)
 w = hanning(n,sflag)

HILBERT

Biấún âặ Hilbert
 y = hilbert(x)

IMPZ

Ấấp ấng xung cuía cáic bầu loüc sặ
 [h,t] = impz(b,a)
 [h,t] = impz(b,a,n)
 [h,t] = impz(b,a,n,Fs)
 impz(b,a)
 impz(...)

INTFILT

Thiấút kấ bầu loüc FIR nặi suy
 b = intfilt(r,l,alpha)
 b = intfilt(r,n,'Lagrange')

KAISER

Cáia sặ Kaiser
 w = kaiser(n,beta)

KAISERORD

Ấấc lấng cáic tham sặ cho thiấút kấ bầu loüc FIR
 vặi cáia sặ Kaiser
 [n,Wn,beta,ftype] = kaiserord(f,a,dev)
 [n,Wn,beta,ftype] = kaiserord(f,a,dev,Fs)
 c = kaiserord(f,a,dev,Fs,'cell')

LATC2TF

Chuyấn bầu loüc Lattice sang hàm biấún âặ
 [num,den] = latc2tf(k,v)
 [num,den] = latc2tf(k,'iir')
 num = latc2tf(k,'fir')
 num = latc2tf(k)

LATCFILT

```

Cài bấu loüc Lattice vài lattice-ladder
[f,g] = latcfilt(k,x)
[f,g] = latcfilt(k,v,x)
[f,g] = latcfilt(k,1,x)
[f,g,zf] = latcfilt(...,'ic',zi)

```

LEVINSON

Ấũ quy Levinson-Durbin

```

a = levinson(r,p)

```

REMEZ

Thiấut kấu bấuloüc FIR tấúi ău Parks-McClellan

```

b = remez(n,f,a)
b = remez(n,f,a,w)
b = remez(n,f,a,'ftype')
b = remez(n,f,a,w,'ftype')
b = remez(...,{lgrid})
b = remez(n,f,'fresp',w)
b = remez(n,f,'fresp',w,'ftype')
b = remez(n,f',{'fresp',p1,p2,...},w)
b = remez(n,f',{'fresp',p1,p2,...},w,'ftype')
[b,delta] = remez(...)
[b,delta,opt] = remez(...)

```

REMEZORD

Tênh thăi tấu bấuloüc FIR tấúi ău Parks-McClellan

```

[n,fo,ao,w] = remezord(f,a,dev)
[n,fo,ao,w] = remezord(f,a,dev,Fs)
c = remezord(f,a,dev,Fs,'cell')

```

RESIDUEZ

Khai triấùn biấún ấũøi z tằng phấõn

```

[r,p,k] = residuez(b,a)
[b,a] = residuez(r,p,k)

```

SAWTOOTH

Phấit sinh rằg cặa hay tam giaĩc soĩng

```

x = sawtooth(t)
x = sawtooth(t,width)

```

SINC

Hàm Sinc

```

y = sinc(x)

```

SQUARE

Phấit sinh soĩng vuặng

```

x = square(t)
x = square(t,duty)

```

Caĩc pheĩp biấún ấũøi nghẽch ấaĩo Fourier**IFFT**

Nghẽch ấaĩo biấún ấũøi Fourier rằi raũc

IFFT(X) nghẽch ấaĩo biấún ấũøi Fourier rằi raũc cuĩa vectấ X

IFFT(X,N) nghịch đảo biến đổi FFT của N-âm thanh

IFFT2

Nghịch đảo biến đổi Fourier của hai chiều
IFFT2(F) trả về nghịch đảo biến đổi Fourier của hai chiều của ma trận F. Nếu F là vectơ thực
kết quả có cùng hướng
IFFT2(F,MROWS,NCOLS) trả về ma trận F với các số 0
với các MROWSxNCOLS thích hợp khi biến đổi

ABS

Giá trị tuyệt đối của các phần tử
ABS(X) là giá trị tuyệt đối của các phần tử của X.
Nếu X phức thì ABS(X) là các phần tử của X
ABS(S), với S là biến MATLAB thực trả về
các giá trị số của các ký tự ASCII trong chuỗi.
Khả năng thay đổi biểu diễn nhị phân trong mã thay đổi
cách in ra

ANGLE

Góc pha
ANGLE(H) trả về góc pha của ma trận các phần tử
phức, đơn vị radian

UNWRAP

UNWRAP Bung góc pha theo đơn vị radian
UNWRAP(P) bung các góc pha P bằng cách thay đổi
bằng nhau tuyệt đối lần đầu tiên sang phần bù 2π
của chuỗi. Với các ma trận thực bung theo từng
cột
UNWRAP(P,TOL) dùng dung sai bằng nhau TOL khác với
ngang bằng TOL = π

FFTSHIFT

FFTSHIFT chuyển số trung tâm 0 về đầu của phần bù
Dịch chuyển FFT. Đầu tiên các vectơ FFTSHIFT(X)
trả về một vectơ hoán đổi các nửa trái và
phải. Với ma trận, FFTSHIFT(X) hoán đổi góc 1 và
3, 2 và 4
FFTSHIFT áp dụng cho FFT, chuyển số trung tâm 0
về đầu của phần bù

CPLXPAIR

CPLXPAIR Sắp xếp các số thành các cặp phức liên
hợp
Y = CPLXPAIR(X) sắp xếp lại các phần tử của
vectơ X để các số phức liên hợp thành các
cặp liên hợp phù hợp. Các cặp phức sắp xếp
theo thứ tự phần thực
Các phần tử thực áp dụng sau tất cả các cặp
phức. Y = CPLXPAIR(X,TOL) dùng dung sai bằng nhau
của TOL cho các phép so sánh. Ngang bằng TOL =
100*EPS

NEXTPW2

NEXTPOW2 muí 2 tiấúp theo

NEXTPOW2(N) traí vấö P ááöu tiấn màì $2^P \geq |N|$ ($2^P \geq \text{abs}(N)$). Thấìng dùng áãø tặm muí 2 gấön nhấút áãü dài dàiỷ cho cáic pheíp biấún áãøi Fourier nhanh (FFT)
Nấúu X lài vectấ thặ NEXTPOW2(X) gấúng
NEXTPOW2(LENGTH(X))

Caïc hàm læüüŋ giaïc**SIN**

SIN(X) lài sin cuía caïc pháön tæí cuía X

SINH

SINH(X) lài sin hyperbol cuía caïc pháön tæí cuía X

ASIN

ASIN(X) lài arcsin cuía caïc pháön tæí cuía X. Năúu vãi pháön tæí naò ãoĩ mà ABS(x) > 1.0 thç kăút quái lài phæic

ASINH

ASINH(X) lài arcsin hyperbol cuía caïc pháön tæí cuía X

COS

COS(X) lài cosin cuía caïc pháön tæí cuía X

COSH

COSH(X) lài cosin hyperbol cuía caïc pháön tæí cuía X

ACOS

ACOS (X) lài arccos cuía caïc pháön tæí cuía X. Năúu vãi pháön tæí naò ãoĩ mà ABS(x) > 1.0 thç kăút quái lài phæic

ACOSH

ACOSH(X) lài arccos hyperbol cuía caïc pháön tæí cuía X

TAN

TAN(X) lài tang cuía caïc pháön tæí cuía X

TANH

TANH(X) lài tang hyperbol cuía caïc pháön tæí cuía X

ATAN

ATAN(X) lài arctang cuía caïc pháön tæí cuía X

ATANH

ATANH(X) lài arctang hyperbol cuía caïc pháön tæí cuía X

SEC

SEC (X) = 1./SIN(X)

SECH

SECH(X) 1./SINH(X)

ASEC

ASEC(X) lài arcsec cuía caïc pháön tæí cuía X

ASECH

ASECH(X) lài arcsec hyperbol cuía caïc pháön tæí cuía X

CSC

$\text{CSC}(X) = 1. / \text{COS}(X)$

CSCH

$\text{CSCH}(X) = 1. / \text{COSH}(X)$

ACSC

$\text{ACSC}(X)$ là arcsec của các phần thập của X

ACSCH

$\text{ACSCH}(X)$ là arccosec hyperbol của các phần thập của X

COT

$\text{COT}(X)$ là cotang của các phần thập của X

COTH

$\text{COTH}(X)$ là cotang hyperbol của các phần thập của X

ACOT

$\text{ACOT}(X)$ là arccotang của các phần thập của X

ACOTH

$\text{ACOTH}(X)$ là arccotg hyperbol của các phần thập của X

Caïc haìm muï vai logarit**EXP**

EXP(X) haìm e^x

LOG

LOG(X) laì logarithm tæu nhiãn cuía caïc pháön tæí cuía X. Nãúu X khäng đæång thç kăút quái pháïc

LOG10

LOG10(X) logarithm cả sãú 10 cuía caïc pháön tæí cuía X. Nãúu X khäng đæång thç kăút quái pháïc

SQRT

SQRT(X) càn bàüc 2 cuía caïc pháön tæí cuía X. Nãúu $X < 0$ thç kăút quái pháïc

Caïc haìm pháïc**ABS**

Giaỉ trẽ tuyăút âăúi vai âăøi chuăùi sang sãú
ABS(X) laì giaỉ trẽ tuyăút âăúi cuía caïc pháön tæí cuía. Nãúu X pháïc thç ABS(X) laì caïc mấun cuía caïc pháön tæí cuía X

ABS(S), vãi S laì biăún chuăùi MATLAB thç traí văo caïc giaỉ trẽ sãúcuía caïc kyï tæu ASCII trong chuăùi. Khäng thay âăøi biăøu hiăün bãn trong mại chề thay âăøi cạìch in ra

ANGLE

Goïc pha

ANGLE(H) traí văo goïc pha cuía ma tráũn caïc pháön tæí pháïc, âãn về radian

CONJ

CONJ(X) laì sãú pháïc liăĩn hăũp cuía X

IMAG

IMAG(X) laì pháön aío cuía X

REAL

REAL(X) laì pháön thăuc cuía X

Hàm vấ sấ nguyẫn vai thậư

ROUND

ROUND(X) làm tròn các phần tấ của X vấ sấ nguyẫn gấn nhấư

FIX

FIX(X) làm tròn các phần tấ của X vấ sấ nguyẫn gấn nhấư vấ phê 0

FLOOR

FLOOR(X) làm tròn các phần tấ của X vấ sấ nguyẫn gấn nhấư vấ phê $-\infty$

CEIL

CEIL(X) làm tròn các phần tấ của X vấ sấ nguyẫn gấn nhấư vấ phê $+\infty$

REM

REM(x,y) là phần còn lại của phép chia x cho y bằng $x - n \cdot y$ vấ $n = \text{fix}(x./y)$ là phần nguyẫn của thằng $x./y$

SIGN

Vấ mặi phần tấ của X, SIGN(X) trái vấ 1 nấu phần tấ lấi hằ 0, trái vấ 0 nấu bằng 0 và trái vấ -1 nấu nhỏ hằ 0. Vấ X phậư, SIGN(X) = $X ./ \text{ABS}(X)$

Các ma trậ cả bắi

ZEROS

ZEROS(N) là mặi ma trậ cắp N gắm các sấ 0

ZEROS(M,N) hoặư ZEROS([M,N]) là mặi ma trậ cải MxN gắm các sấ 0

ZEROS(SIZE(A)) cùng cải vấi A vài tắư cải âđầ lài các sấ 0

ONES

ONES(N) là mặi ma trậ cắp N gắm các sấ 1

ONES(M,N) hoặư ONES([M,N]) là mặi ma trậ cải MxN gắm các sấ 1

ONES(SIZE(A)) cùng cải vấi A vài tắư cải âđầ lài các sấ 1

EYE

Ma trậ ân vể

EYE(N) là ma trậ ân vể cắp N

EYE(M,N) hoặư EYE([M,N]) là mặi ma trậ cải MxN gắm các sấ 1 trắ

âằi cheồ, còn lại lài các sấ 0

EYE(SIZE(A)) cùng cải vấi A

RAND

Phán bấu âăông bấu caïc sấu vai caïc ma tráun ngáuun nhiên

RAND(N) lai ma tráun cáúp N vãi caïc sấu ngáuun nhiên, thăng thăêng choún tãi viăuc phán bấu trong khoăing (0.0,1.0)

RAND(M,N) hoăuc RAND([M,N]) lai ma tráun câi MxN vãi caïc sấu ngáuun nhiên

RAND(SIZE(A)) cùng câi vãi A

RAND khăng tham sấu lai măt âăui lăăung vă hăăing coi giaĩ trê thay âăui

RAND('seed') trái văo giaĩ trê khăi taô hiăun thăi cuia bấu phaĩt sinh ngáuun nhiên

RAND('seed',s) âăut giaĩ trê khăi taô cuia bấu phaĩt sinh ngáuun nhiên lai s

RAND('seed',0) âăut lai giaĩ trê khăi taô cuia bấu phaĩt sinh ngáuun nhiên

RAND('seed',sum(100*clock)) âăut bấu phaĩt sinh ngáuun nhiên thay âăoi măt luĩc măt giaĩ trê

Ngăom âenh, RAND láuy măt phán bấu âăông bấu. Hàm RANDN phaĩt sinh caïc ma tráun ngáuun nhiên phán bấu chuăon

RAND vai RANDN coi caïc caĩch phaút sinh riăng, măt hàm dùng giaĩ trê khăi taô riăng

Caïc phiăn baĩn trăăc cuia MATLAB cho pheĩ RAND('normal') âăo chuyăon phán bấu âang dùng sang

phán bấu chuăon, RAND('uniform') âăo chuyăon ngăăc lai phán bấu âăông bấu, vai RAND('dist') âăo trái văo măt chuăui chăia phán bấu âang dùng, hoăuc âăông bấu ('uniform') hoăuc chuăon ('normal'). MATLAB 4.0

tiăúp tuĩc cho pheĩ caĩc chuyăon âăoi năy, nhăng cho ra măt khuyăun caĩo

RANDN

Caïc sấu vai caïc ma tráun ngáuun nhiên phán bấu chuăon

RANDN(N) lai măt ma tráun vưăng cáúp NxN coi caĩc phăon tãi ngáuun nhiên, choún tãi

phán bấu chuăon vãi trung bệnh 0.0 vai phăng sai 1.0

RANDN(M,N) hoăuc RANDN([M,N]) lai ma tráun câi MxN vãi caĩc phăon tãi ngáuun

nhiên

RANDN(SIZE(A)) cùng câi vãi A

RANDN khăng tham sấu lai măt âăui lăăung vă hăăing thay âăoi theo tăng lăunh goĩ

RANDN('seed') trái văo giaĩ trê khăi taô hiăun thăi cuia bấu phaĩt sinh chuăon

RANDN('seed',s) âăut giaĩ trê khăi taô cuia bấu phaĩt sinh chuăon vaĩ s

RANDN('seed',0) âăut giaĩ trê khăi taô vaĩ giaĩ trê ban âău

RANDN('seed',sum(100*clock)) âăut vaĩ măt giaĩ trê khăic theo tăng luĩc

RANDN vai RAND coi caĩc caĩch phaút sinh riăng, măt hàm dùng giaĩ trê khăi taô

riêng

Linspace

Linspace vectơ có khoảng cách tuyến tính
 Linspace(x1, x2) phát sinh vectơ dòng gồm 100
 điểm cách đều tuyến tính giữa
 x1 và x2
 Linspace(x1, x2, N) phát sinh N điểm giữa x1 và
 x2

Logspace

Logspace Vectơ có khoảng cách logarit
 Logspace(d1, d2) phát sinh vectơ dòng gồm 50
 điểm cách đều giữa 10d1 và
 10d2. Nếu d2 là pi thì các điểm giữa 10d1
 và pi
 Logspace(d1, d2, N) phát sinh N điểm

Meshgrid

Meshgrid phát sinh các mảng X và Y cho các hàm
 vẽ 3 chiều
 [X,Y] = Meshgrid(x,y) biến đổi hai miền giá trị
 trục x và y sang
 các mảng X và Y để dùng cho việc tính các
 hàm hai biến và các hàm vẽ
 mặt 3 chiều. Các dòng của mảng xuất X là
 các bán kính của vectơ x và các
 cột của mảng xuất Y là các bán kính của
 vectơ y
 Ví dụ, để tính hàm $x \cdot \exp(-x^2 - y^2)$ trên miền $-2 < x < 2$, $-2 < y < 2$,
 [X,Y] = meshgrid(-2:.2:2, -2:.2:2);
 Z = X .* exp(-X.^2 - Y.^2);
 mesh(Z)
 [X,Y] = Meshgrid(x) là cách gọi của [X,Y] =
 Meshgrid(x,x)
 [X,Y,Z] = Meshgrid(x,y,z) cho ra mảng nền 3
 chiều để dùng cho việc tính
 các hàm 3 biến và các hàm vẽ khối 3 chiều

Phân tích ma trận

Cond

Số điều kiện của ma trận
 Cond(X) là tỉ số của giá trị kỳ dị lớn nhất và
 nhỏ nhất của X, cho biết số điều kiện theo chuẩn-
 2

Norm

Chuẩn ma trận hoặc vectơ
 Với ma trận:
 Norm(X) là giá trị kỳ dị lớn nhất của X,
 max(svd(X))
 Norm(X,2) giống Norm(X).

NORM(X,1) là chuẩn-1 của X, tổng tất cả lần nhất, $\max(\sum(\text{abs}((X))))$
 NORM(X,inf) là tổng tất cả lần nhất, $\max(\sum(\text{abs}((X'))))$
 NORM(X,'inf') giống NORM(X,inf)
 NORM(X,'fro') là chuẩn-F, $\sqrt{\sum(\text{diag}(X'*X))}$
 NORM(X,P) coi thử dùng cho ma trận X chế nếu P là 1, 2, inf hoặc 'fro'

Với vectơ:

NORM(V,P) = $\sum(\text{abs}(V)^P)^{(1/P)}$.

NORM(V) = norm(V,2).

NORM(V,inf) = $\max(\text{abs}(V))$.

NORM(V,-inf) = $\min(\text{abs}(V))$.

Trong MATLAB 4.0, nếu X coi các thành phần phức, z, thì $\text{abs}(z) = \sqrt{z*\text{conj}(z)}$, chế khác là $\text{abs}(\text{real}(z)) + \text{abs}(\text{imag}(z))$, dùng trong các phiên bản trước của MATLAB

RCOND

LINPACK ước lượng điều kiện thuận nghịch

RCOND(X) là ước lượng điều kiện thuận nghịch của X theo chuẩn-1 nhân ước lượng điều kiện LINPACK. Nếu X coi điều kiện tốt thì RCOND(X) gần bằng 1.0. Nếu X coi điều kiện xấu thì RCOND(X) gần bằng 0.0

RANK

Số dòng hoặc tất cả lập tùy ý tên

K = RANK(X) là số giá trị kỳ dị của X, lần hạn $\text{MAX}(\text{SIZE}(X)) * \text{NORM}(X) * \text{EPS}$

K = RANK(X,tol) là số giá trị kỳ dị của X, lần hạn tol

DET

Định thức

DET(X) là định thức của ma trận vuông X

TRACE

Tổng các phần tử trên đường chéo

TRACE(A) là tổng các phần tử trên đường chéo của A, cũng là tổng các giá trị riêng của A

NULL

Không gian không

Z = NULL (A) cả số thực chuẩn của không gian không A. $Z'*Z = I$, $A*Z$ coi các phần tử không ảnh hưởng, số tất cả của Z là số khuyết của A

ORTH

Thực giao

Q = ORTH(A) là cả số thực chuẩn của tất cả với điều kiện giá trị riêng của A. $Q'*Q = I$, các tất cả của Q bung ra như các tất cả A vài số tất cả của Q là hàng của A

RREF

Thu gọn dòng dưới bậc thang

$R = \text{RREF}(A)$ cho ra đúng bậc thang thu gọn dòng của A
 $[R, jb] = \text{RREF}(A)$ cũng trả về một vectơ, jb , mà:
 $r = \text{length}(jb)$ là hàng của A , về một thuật toán
 $x(jb)$ là các biến trong hệ phương trình
 tuyến $Ax = b$,
 $A(:, jb)$ là các số của A ,
 $R(1:r, jb)$ mà trận nhân về cấp r
 $[R, jb] = \text{RREF}(A, \text{TOL})$ dùng dung sai cho sai lệch trong
 việc kiểm tra hàng
 Thuật toán này bỏ qua sai số khi tên hàng khác
 nhau là RANK , ORTH và NULL

Phương trình tuyến tính

phép \ và / để giải phương trình tuyến tính

CHOL

Thủ tục Cholesky

$\text{CHOL}(X)$ để dùng để giải các bài toán tam giác trên của X . Tam giác dưới để giải các bài toán về (liên hệ phương trình) của tam giác trên. Nếu X xác định dương thực $R = \text{CHOL}(X)$ cho ra một tam giác trên với $R'R = X$. Nếu X không xác định dương thực một thông báo lỗi sẽ in ra. Với hai đối số xuất, $[R, p] = \text{CHOL}(X)$ không bỏ lỗi. Nếu X xác định dương thực $p = 0$ và R như trên. Nếu X không xác định dương thực p là một số nguyên dương và R là ma trận tam giác trên bậc $q = p-1$ với $R'R = X(1:q, 1:q)$

LU

Tách nhân để dùng phương pháp Gauss

$[L, U] = \text{LU}(X)$ lưu ma trận tam giác trên trong U và L là một
 "ma trận tam giác dưới về tám ly", nghĩa là tích
 ma trận tam giác dưới và ma trận hoán vị, để $X = L*U$
 $[L, U, P] = \text{LU}(X)$ trả về ma trận tam giác dưới L , ma
 trận tam giác trên U , và ma trận hoán vị P
 để $P*X = L*U$
 Chức năng $\text{LU}(X)$ trả về một phuộc với xuất ZGEFA của
 LINPACK

INV

Nghịch đảo ma trận

$\text{INV}(X)$ nghịch đảo của ma trận X . Một khuyến cáo
 sẽ in ra nếu X gần kỳ

QR

Taích tam giaïc trêuc giao

$[Q,R] = QR(X)$ ma tráu tam giaïc trăn R cùng câi vãi X vai ma tráu ân nguyên Q âãø $X = Q \cdot R$

$[Q,R,E] = QR(X)$ cho ra ma tráu hoãn về E , ma tráu tam giaïc trăn R vai ma tráu ân nguyên Q âãø $X \cdot E = Q \cdot R$. Căüt hoãn về E âãüc choün sao cho $\text{abs}(\text{diag}(R))$ tăng

$[Q,R] = QR(X,0)$ cho ra phán taích "kêch thăïc kinh tấ " . Năúu X coĩ câi mxn vãi $m > n$ thç chè coĩ n căüt âăü cuía Q âãüc tênh

$[Q,R,E] = QR(X,0)$ cho ra phán taích "kêch thăïc kinh tấ " trong âoĩ E lài vectă hoãn về âãø $Q \cdot R = X(:,E)$. Căüt hoãn về cuía E âãüc choün sao cho $\text{abs}(\text{diag}(R))$ tăng

Chênh $QR(X)$ trái văö phuêc vü xuát ZGEFA cuía LINPACK. $TRIU(QR(X))$ lài R

QRDELETE

QRDELETE xoĩa măüt căüt trong thăia săü QR

Năúu $[Q,R] = qr(A)$ thăia săü QR cuía A thç $[Q,R] = qrdelete(Q,R,j)$ âăøi Q vai R thănh thăia săü cuía ma tráu vãi $A(:,j)$ bẽ xoĩa

QRINSERT

QRINSERT chên măüt căüt trong thăia săü QR

Năúu $[Q,R] = qr(A)$ thăia săü QR cuía A thç $[Q,R] = qrinsert(Q,R,j,x)$ âăøi Q vai R thănh thăia săü cuía ma tráu nhăün âãüc bằng căich chên căüt x vào trăïc $A(:,j)$. (Năúu A coĩ n căüt vai $j = n+1$, thç x chên sau căüt cuăúi cùng cuía A)

NNLS

Bệnh phăng tăúi thiăü khăng ám

$X = NNLS(A,b)$ trái văö vectă X lài nghiăüm $A \cdot x = b$ theo phăng phăp bệnh phăng tăúi thiăü, thoía măn $x \geq 0$

Dung sai ngăöm âênh cuía $TOL = \text{MAX}(\text{SIZE}(A)) * \text{NORM}(A,1) * \text{EPS}$ âãüc dùng âãø quyăüt âênh khi nào thç căic phăön tăi cuía X nhó hăn 0. Coĩ thăø goũ âêi vãi $X = NNLS(A,b,TOL)$

$[X,W] = NNLS(A,b)$ cũng trái văö vectă âăúi ngăü W vãi $w(i) < 0$ khi $x(i) = 0$ vai $w(i) = 0$ khi $x(i) > 0$

PINV

Giaí âáo

$X = \text{PINV}(A)$ cho ra ma tráu X cùng câi vãi A' sao cho $A \cdot X \cdot A = A$, $X \cdot A \cdot X = X$, à AX vai XA lài Hermitian. Căich tênh đêü vào $\text{SVD}(A)$ vai giaĩ trê kyì đê nhó hăn dung sai âãüc xem lài 0

Dung sai ngăöm âênh lài $\text{MAX}(\text{SIZE}(A)) * \text{NORM}(A) * \text{EPS}$.

Dung sai này coĩ thăø goũ âêi vãi $X = \text{PINV}(A,tol)$

LSCOV

Giaíi bệnh phẻng beỉ nhứt vủi phẻng sai chẻa biẻut $X = \text{LSCOV}(A, b, V)$ traỉ vủo vectỏ X mài caỉc giaỉ trẻ nhỏi nhứt laỉ $(A*X - b)' * \text{inv}(V) * (A*X - b)$. Nẻu A cỏi $M \times N$ thẻ b phảỉ laỉ $M \times 1$ vài V phảỉ laỉ $M \times M$. Lỏi giaỉ X seỉ laỉ $N \times 1$. Ấy laỉ bài toỏn bệnh phẻng beỉ nhứt ($M > N$) vủi phẻng sai V . Lỏi giaỉi ỏỏỏ tẻm thỏủy mài khẻng tẻnh nghẻch ỏỏỏ V . Lỏi giaỉi ỏỏỏ sỏu tuyỏn tẻnh cỏỏ ỏỏỏỏn cho bài toỏn nỏy laỉ:

$$x = \text{inv}(A' * \text{inv}(V) * A) * A' * \text{inv}(V) * b$$
Giaỉ trẻ riẻng vài giaỉ trẻ kỳi đẻ**EIG**

Giaỉ trẻ riẻng vài vectỏ riẻng

$\text{EIG}(X)$ vectỏ riẻng chẻa caỉc giaỉ trẻ riẻng của ma trỏn vủng X

$[V, D] = \text{EIG}(X)$ cho ra mỏt ma trỏn ỏỏỏỏ cheỉo D gỏm caỉc giaỉ trẻ riẻng vài ma trỏn ỏỏỏỏ ỏỏỏ V mài caỉc cỏt laỉ caỉc vectỏ riẻng tẻng ẻng sao cho $X*V = V*D$

$[V, D] = \text{EIG}(X, 'nobalance')$ thỏc hiẻn tẻnh toỏn khẻng cho phẻp cỏn bẻng, ỏỏỏ lỏic cho kỏt quỏi chẻnh ỏỏỏ hỏn.

Caỉc giaỉ trẻ riẻng vài vectỏ riẻng tẻng quỏt

$\text{EIG}(A, B)$ laỉ vectỏ chẻa caỉc giaỉ trẻ riẻng tẻng quỏt của caỉc ma trỏn vủng A vài B

$[V, D] = \text{EIG}(A, B)$ cho ra ma trỏn ỏỏỏỏ cheỉo D gỏm caỉc giaỉ trẻ riẻng tẻng quỏt vài ma trỏn ỏỏỏỏ ỏỏỏ V mài caỉc cỏt laỉ caỉc vectỏ riẻng sao cho $A*V = B*V*D$

POLY

Ảa thỏc ỏỏỏ trẻng

Nẻu A laỉ ma trỏn vủng cỏp N thẻ $\text{POLY}(A)$ laỉ vectỏ đỏing gỏm $N+1$ phỏn tẻi laỉ caỉc hỏu sỏu của ỏỏỏ thỏc ỏỏỏ trẻng, $\text{DET}(\text{lambd}a * \text{EYE}(A) - A)$. Nẻu V laỉ vectỏ $\text{POLY}(V)$ laỉ vectỏ chẻa caỉc phỏn tẻi laỉ caỉc hỏu sỏu của ỏỏỏ thỏc ỏỏỏ trẻng cỏi nghiỏm ỏỏỏ thỏc laỉ caỉc phỏn tẻi của V . Vủi caỉc vectỏ thẻ ROOTS vài POLY laỉ caỉc haỉm ỏỏỏ của nhau

$\text{ROOTS}(\text{POLY}(1:20))$ phỏt sinh vẻ đủi nỏi tiẻng của Wilkinson

POLYEIG

POLYEIG laỉ bài toỏn giaỉ trẻ riẻng ỏỏỏ thỏc

$[X, E] = \text{POLYEIG}(A_0, A_1, \dots, A_p)$ giaỉ bài toỏn giaỉ trẻ riẻng ỏỏỏ thỏc bỏc p :

$$(A_0 + \text{lambd}a * A_1 + \dots + \text{lambd}a^p * A_p) * x = 0$$

Nhỏp $p+1$ ma trỏn vủng bỏc n laỉ A_0, A_1, \dots, A_p . Ma trỏn ỏỏỏ cỏi $n \times n * p$, X , cỏi caỉc cỏt laỉ caỉc vectỏ riẻng, vài mỏt vectỏ ỏỏỏ dài $n * p$, E , cỏi caỉc phỏn tẻi laỉ caỉc giaỉ trẻ riẻng

```
for j = 1:n*p
    lambd = E(j)
    x = X(:, j)
```

```

% (A0 + lambda*A1 + ... + lambda^p*Ap)*x  xấp
xè 0
end
Caïc trắàng hắp àầu biắu:
p = 0, polyeig(A), lài bài toăin giaỉ trề riắng
chúáon, eig(A)
p = 1, polyeig(A,B), lài bài toăin giaỉ trề
riắng táong quắit, eig(A,-B)
n = 1, polyeig(a0,a1,...,ap), cho caïc vắ hắing
a0, ..., ap, lài bài toăin àa thắic chúáon, roots([ap
.. a1 a0])
Nắu cái hai A0 vắ Ap lài kắi đề thặ bài toăin thuắc
loắi táom thắing. Vắo lý thuyắut thặ lắi giaỉi vắ
nghiắum hoắc khắng duy nhắut. Vắo tềnh toăin thặ lắi
giaỉi coỉ thắo khắng àuắng. Thắi taúo ra trắàng hắp
nắy thặ kắut quắi coỉ thắo lài mắut khuyắun caío.
Nắu chề coỉ mắut A0 hay Ap lài kắi đề thặ bài toăin
khắi lài hoắc buắa, nhắng vắi giaỉ trề riắng coỉ thắo
0 hoắc "vắ àềnh"

```

HESS

Đầung Hessenberg

Đầung Hessenberg cuắa mắut ma tráun lài ma tráun coỉ
caïc sắu 0 đắi àắing cheío thắi nhắut. Nắu ma tráun
àắi xắng hay Hermitian thặ đầung lài mắut đắi bắu ba
àắing cheío. [P,H] = HESS(A) cho ra mắut ma tráun àần
nguyắn P vắ mắut ma tráun Hessenberg H vắi $A = P*H*P'$
vắ $P'*P = \text{EYE}(\text{SIZE}(P))$
Chềnh hàm HESS(A) trái vắo H

QZ

Giaỉ trề riắng táong quắit

[AA, BB, Q, Z, V] = QZ(A,B) àắi vắi caïc ma tráun
vuắng A vắ B, cho ra caïc ma tráun tam giaỉc trắn AA
vắ BB, caïc ma tráun Q vắ Z chắa caïc chuyắon vắ
trắi vắ phắi sao cho $Q*A*Z = AA$, vắ $Q*B*Z = BB$, vắ
ma tráun vectắ riắng táong quắit V
Caïc giaỉ trề riắng táong quắit lài caïc phắon tái
àắing cheío cuắa AA vắ BB àắo
 $A*V*\text{diag}(BB) = B*V*\text{diag}(AA)$

Caic haım ma tráuın

EXPM

Muı e ma tráuın

EXPM(X) laı ma tráuın muı e cuıa X. EXPM aããüc tênh bầıng thuáút toaın xáúp xè Pade. Mầıc đui nỏı khắng tênh theo caıch naıy: nầıú X coı mầıt táúp aáoıy aúi caıc vectả riắng vắıı caıc giaı trề riắng tắng ắıng D, thç $[V, D] = \text{EIG}(X)$ vaı

$\text{EXPM}(X) = V * \text{diag}(\exp(\text{diag}(D))) / V$

EXP(X) (khắng coı M) thắc hiắn trắn tắng phắn tắı

EXPM1

Muı e ma tráuın thắng qua xáúp xè Pade

$E = \text{EXPM1}(A)$ laı mầıt bắø sung M-file cuıa thuáút toaın bắn trong MATLAB aáoı vắıı tênh muı e cuıa ma tráuın

EXPM2

Muı e ma tráuın thắng qua chuáúi Taylor

$E = \text{expm2}(A)$ minh hoúa ắnh nhéa cầo ắıắn cho tênh muı e ma tráuın. Nhé mầıt phắng phaıp thắc hắnh sắ, caıch naıy thắıng cháúm vaı thiắúu chềnh xáıc

EXPM3

Muı e ma tráuın thắng qua giaı trề riắng vaı vectả riắng $E = \text{expm3}(A)$ minh hoúa mầıt caıch coı thắø ắắø tênh muı e ma tráuın. Nhé mầıt phắng phaıp thắc hắnh vắø sắ, ắắø chềnh xáıc aããüc xáıc ắnh bắıı ắıắıu kiắn cuıa ma tráuın vectả riắng

LOGM

Lo-ga-rit ma tráuın

$L = \text{LOGM}(A)$ laı ma tráuın logarit cuıa A. Kắıt quía phắıc nầıú A coı caıc giaı trề riắng ắm. Mầıt khuyắún caıo nầıú $\text{expm}(L)$ khắng gắn vắıı A

$[L, \text{esterr}] = \text{logm}(A)$ khắng in ra khuyắún caıo, nhắng trắı vắø mầıt ắắıc lắắng thắıng đắ tắng ắắıı, $\text{norm}(\text{expm}(L) - A) / \text{norm}(A)$. Nầıú A laı ắắıı xắıng thắc hay Hermitian phắıc thç ắıı laı $\text{LOGM}(A)$

Mầıt sắ ma tráuın thắc hoắc phắıc khắng coı loga, nhé $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, vaı LOGM khắng thắø cho kắıt quía

SQRTM

Càn báıc hai ma tráuın

$S = \text{SQRTM}(A)$ càn báıc hai cuıa ma tráuın A

Kắıt quía phắıc nầıú A coı giaı trề riắng ắm. Mầıt khuyắún caıo nầıú tênh $S * S$ khắng gắn vắıı A $[S, \text{esterr}] = \text{sqrtm}(A)$ khắng in khuyắún caıo naıo cáı mắ trắı vắø ắắıc lắắng thắıng đắ tắng ắắıı, $\text{norm}(S * S - A) / \text{norm}(A)$. Nầıú A laı ắắıı xắıng thắc hay Hermitian phắıc thç ắıı laı $\text{SQRTM}(A)$

Mầıt sắ ma tráuın thắc hoắc phắıc khắng coı càn báıc hai, nhé $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, vaı SQRTM khắng thắø cho kắıt quía

FUNM

Tên hàm của mặt ma trận

$F = \text{FUNM}(A, 'fun')$ hoặc $F = \text{FUNM}('fun', A)$, với α là số ma trận A , tên hàm ma trận α là $'fun'$. Ví dụ, $\text{FUNM}(A, 'sin')$ là sin của ma trận A . Với μ , loga và căn bậc hai ma trận thực, thay vào đó, dùng $\text{EXPM}(A)$, $\text{LOGM}(A)$ và $\text{SQRTM}(A)$. FUNM dùng thuật toán khác nhau khác nhau. Nếu A gần với ma trận có nhiều giá trị riêng và vectơ riêng thì FUNM có thể cho kết quả khác nhau.

$[F, \text{ESTERR}] = \text{FUNM}(A, 'fun')$ khác in ra bằng bao nhiêu cái mà trái với mặt α là FUNM sai số bằng α trong kết quả tên toán

Nếu A là α Hermitian thực dùng Schur của nó là α chéo và FUNM có thể cho kết quả khác nhau

$S = \text{SQRTM}(A)$ và $L = \text{LOGM}(A)$ dùng FUNM để tính, nhưng cũng có thể α tin cậy hơn bằng cách so sánh S^2 và $\text{EXPM}(L)$ với A . $E = \text{EXPM}(A)$ dùng mặt thuật toán hoàn toàn khác

Caic biếu vai hàng α biếu**ANS**

Trái lại gần nhất (ANSwer)

ANS là biếu α ra α khi biếu α khác α gần cho biếu nào cái

EPS

Áp dụng các số thực

EPS là mặt biếu α α giá trị α là khoảng cách tại 1.0 α số thực lần nhất tiếp theo. EPS có thể α gần lại giá trị bất kỳ. EPS α dùng α sai cho các hàm PINV và RANK

REALMAX

REALMAX Số thực dạng lần nhất

$x = \text{realmax}$ là số thực dạng lần nhất biếu điều trị mặt tên. Mỗi số lần α bề trị

REALMIN

REALMIN Số thực dạng nhỏ nhất

$x = \text{realmin}$ là số thực dạng nhỏ nhất và α chuẩn hóa trị mặt tên. Mỗi giá trị nhỏ α bề trị α theo chuẩn IEEE

PI

3.1415926535897....

$\text{PI} = 4 * \text{atan}(1) = \text{imag}(\log(-1)) = 3.1415926535897....$

I , J

Ảnh vẽ α

Cái hai biảún i vài j coi giaỉ trẻ ááöu lài sqrt(-1) áãø dùng cho sấu phæíc. Vê duû, caíc biãøu thæíc $3+2i$, $3+2*i$, $3+2i$, $3+2*j$ vài $3+2*\text{sqrt}(-1)$, táút cái áãöu coi cùng măt giaỉ trẻ. Tuy nhiên, cái hai i vài j coi thãø áæãüc gáin cho caíc giaỉ trẻ kháíc, thæång dùng trong caíc vòng lầup FOR vài làm caíc chề sấu

INF

Giaỉ trẻ báút áềnh

INF lài măt biảún cấ áềnh biãøu hiãún giaỉ trẻ báút áềnh đặng theo chuáøn sấu hoüc IEEE. Inf nháún áæãüc tãi caíc pheíp toáin nhæ chia cho 0, nhæ $1.0/0.0$, hoàüc trạin, nhæ $\exp(1000)$

NAN

Khăng lài sấu (Not-a-Number)

FLOPS

Áấúm caíc pheíp toáin sấu thæüc

FLOPS trái vấ sấu tềch lầup caíc pheíp toáin trấ sấu thæüc. FLOPS(0) áầút giaỉ trẻ áấúm vấ 0. Noỉ khăng lài áấúm tuyấút áầúi táút cái caíc pheíp toáin trấ sấu thæüc, nhặg háöu háút caíc pheíp toáin quan troúng áãöu áæãüc áấúm. Caíc pheíp cầng vài trặ lài 1 flop nầú thæüc vài 2 nầú phæíc. Caíc pheíp nhán vài chia lài 1 flop cho mầi pheíp nầú kầút quái lài thæüc vài 6 flop nầú khăng pháỉ. Caíc hàim cả báin áấúm 1 nầú thæüc vài nhiãöu hân nầú lài phæíc

Vài vê duû, nầú A vài B lài caíc ma tráun thæüc vắng cáúp N, thç $A + B$ áấúm N^2 flop, $A * B$ áấúm $2*N^3$ flop, $A ^ 100$ áấúm $99*2*N^3$ flop, LU(A) áấúm thắ $(2/3)*N^3$ flop

NARGIN

Sấu áầúi sấu nhắup cho măt hàim

Bắ cầnh thán hàim đo ngắỉ dùng áềnh nghéa (UDF), biảún cấ áềnh NARGIN biãøu hiãún sấu áầúi sấu nhắup dùng áãø gồỉ hàim, vài NARGOUT biãøu hiãún sấu áầúi sấu xuấút

NARGOUT

NARGOUT Său ăăúi său xuăút cuía măút hăim

Băn cănh thăn hăim đơ ngăăi dùng ăềnh nghéa (UDF), biăún cău ăềnh NARGIN biăêu hiăún său ăăúi său nhăúp dùng ăăô gôúi hăim, vại NARGOUT biăêu hiăún său ăăúi său xuăút

COMPUTER

COMPUTER hoúi kiăôu măiy tênh

COMPUTER trăi văô măút chuăúi chăĩa tăn măiy ăang lăim MATLAB. Căic măiy côi thăô lăi:

PCWIN -	MS-Windows	DEC_RISC	-
	DecStation		
MAC2 -	Macintosh	VAX_VMSD	-
	VAX/VMS D_float		
SUN4 -	Sun SPARC	VAX_VMSG -	VAX/VMS
G_float			
SOL2 -	Solaris 2	ALPHA	-
	Dec Alpha		
HP700 -	HP 9000/700	HP300	-
	HP 9000/300		
SGI -	Silicon Graphics	CRAY -	Cray
IBM_RS -	IBM RS6000	CONVEX -	
	Convex		

[C,MAXSIZE] = COMPUTER cūng trăi văô său nguyăn MAXSIZE chăĩa său tăúi ăa căic phăôn tăi cho pheíp trong 1 mă trăun trăn phiăn băin năy cuía MATLAB

ISIEEE

Ăuĩng cho măiy tênh văii chuăôn său hoüc IEEE

ISIEEE trăi văô 1 trăn căic măiy tênh văii chuăôn său hoüc IEEE vại 0 trăn căic măiy tênh khăng theo chuăôn său hoüc IEEE. Căic măiy theo chuăôn IEEE lăi: IBM PC, hăou hăút căic măiy trăum cuía UNIX, Macintosh. Căic măiy khăng theo chuăôn IEEE lăi: VAX, Cray

ISSTUDENT Ăuĩng văii phiăn băin MATLAB đănh cho sinh viăn (Student Edition)

WHY

Cung căúp căic trăi lăi ngăôn gôun cho căic cău hoúi băút kyì

VERSION

Phiăn băin cuía MATLAB

VERSION trăi văô măút chuăúi chăĩa său phiăn băin MATLAB

Thời gian vai nhất kỳ

CLOCK

Ấyng hăo tằing. CLOCK trái vắ vectả đoiंग gắm 6 phắn tắ chắi giắ vai nhất kỳ hiắu tắi trong đắng thắp phắ:

```
CLOCK = [năm thaingngay giắ phứt giắ]
Năm phắn tắ áắu lài nguyắ. Phắn tắ giắ chềh
xắ áắu vắi chắ sắu sau đắu chắu thắp phắ
FIX(CLOCK) làm troiंग sang đắng hiắo thề nguyắ
```

CPUTIME

CPUTIME CPU thắi gian theo ánn vắ giắ
CPUTIME trái vắ thắi gian CPU theo ánn vắ giắ áắ
đùng bắi MATLAB tắ khi khắi áắu
về đắ:

```
t=cputime; thao tắ; cputime-t
trái vắ thắi gian cpu dùng cho thao tắ
Giắ trề trái vắ coi thắo trắ biắu hiắu bắ trong
```

DATE

Lềh

S = DATE trái vắ mắu chuắi chắi nhất kỳ đắng ngay-thaing-năm (dd-mmm-yy)

ETIME

Thắi gian trắ qua

ETIME(T1,T0) trái vắ thắi gian trắ qua giắ 2 vectả T1 vài T0 theo ánn vắ giắ. Hai vectả phắi coi áắing 6 phắn tắ, theo đắng trái vắ bắi CLOCK:

```
T = [Năm Thaing Ngay Giắ Phứt Giắ]
Áắ lài mắu về đắ dùng ETIME áắo tềh thắi gian
thao tắ: t0 = clock; thao tắ; etime(clock,t0)
Lắu yắ: Nhắ lài phắn bắ sung, nắ khắng làm viắu
qua cáic ngay cuắi thaing hoắc năm. Lài mắu M-file,
vắ vắu coi thắo sắi áắo nắu cáo
```

TIC

Bắo áắu cho bắ tềh thắi gian

```
Đắy lắnh TIC; thao tắ; TOC
in ra thắi giau thắu hiắu
```

TOC

Áắo bắ tềh thắi gian

```
TOC, in thắi gian trắ qua tềh tắ khi dùng TIC
t = TOC; lắu thắi gian trắ qua trong t, thay cho
viắu in nắ ra
```

Thao tắ trắ ma trắu

DIAG

Tắo mắi hoắc trềh ra cáic áắing chềo

Nếu V là một vectơ dòng hoặc cột với N phần tử thì $\text{DIAG}(V, K)$ là một ma trận vuông cấp $N + \text{ABS}(K)$ với các phần tử của V trên đường chéo thứ K . $K = 0$ là đường chéo chính, $K > 0$ là trên đường chéo chính và $K < 0$ là dưới đường chéo chính. $\text{DIAG}(V)$ nhân giá trị V vào đường chéo chính. Về đầu, $\text{DIAG}(-M:M) + \text{DIAG}(\text{ONES}(2*M,1),1) + \text{DIAG}(\text{ONES}(2*M,1),-1)$ cho ra một ma trận 3 đường chéo bậc $2*M+1$.
 Nếu X là ma trận thì $\text{DIAG}(X, K)$ là vectơ cột tạo nên các phần tử của đường chéo thứ K của X . $\text{DIAG}(X)$ là đường chéo chính của X . $\text{DIAG}(\text{DIAG}(X))$ là một ma trận đường chéo

FLIPLR

Lần ngược các dòng ma trận theo hướng trái/phải
 $\text{FLIPLR}(X)$ trái với X với dòng đảo ngược và các cột đảo theo hướng trái/phải

$X = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{matrix}$ thành $\begin{matrix} 3 & 2 & 1 \\ 6 & 5 & 4 \end{matrix}$

FLIPUD

Lần ngược các cột ma trận theo hướng trên/dưới
 $\text{FLIPUD}(X)$ trái với X với cột đảo ngược và các dòng đảo theo hướng trên/dưới

Về đầu,
 $X = \begin{matrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{matrix}$ thành $\begin{matrix} 3 & 6 \\ 2 & 5 \\ 1 & 4 \end{matrix}$

RESHAPE

RESHAPE Thay đổi kích thước
 $\text{RESHAPE}(X, M, N)$ trái với ma trận cỡ $M \times N$ với các phần tử được lấy tại X theo từng cột. Một kết quả là nếu X không có ứng $M \times N$ phần tử
 $\text{RESHAPE}(X, [M \ N])$ cũng làm việc

ROT90

Quay ma trấu 90°

rot90(A) lài quay 90° cuía ma trấu A.

rot90(A,k) lài quay (k*90) ° ma trấu A, k = ±1, ±2,

...

Vê duû,

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad B = \text{rot90}(A) = \begin{bmatrix} 3 & 6 \\ 2 & 5 \\ 1 & 4 \end{bmatrix}$$

sau àoĩ mesh(B) biãøu hiãũn quay theo chiãøu đãøng 90o cuía mesh(A)

TRIL

Tam giaĩc đãĩĩ

TRIL(X) lài pháõn tam giaĩc đãĩĩ cuía X

TRIL(X,K) lài caĩc pháõn táĩ bãn đãĩĩ âẽàng cheĩo thãĩ K cuía X. K = 0 lài âẽàng cheĩo chẽnh, K > 0 lài trãn âẽàng cheĩo chẽnh vaĩ K < 0 lài đãĩĩ âẽàng cheĩo chẽnh

TRIU

Tam giaĩc trãn

TRIU(X) lài pháõn tam giaĩc trãn cuía X

TRIU(X,K) lài caĩc pháõn táĩ bãn trãn âẽàng cheĩo thãĩ K cuía X. K = 0 lài âẽàng cheĩo chẽnh, K > 0 lài trãn âẽàng cheĩo chẽnh vaĩ K < 0 lài đãĩĩ âẽàng cheĩo chẽnh

: láúy chẽ sãú ma trấu , sãõp xãúp laũi ma trấu

Caĩc hàm coĩ âũĩ sãú laĩ hàm**ODE23**

Giaĩĩ hãũ phãøng trẽnh vi phãn, phãøng phaĩp báũc tháũp ODE23 tẽch phãn hãũ phãøng trẽnh vi phãn thẽàng bàõng caĩch đũng cãng thẽĩc Runge-Kutta báũc 2 vaĩ 3

[T,Y] = ODE23('yprime', T0, Tfinal, Y0) tẽch phãn hãũ phãøng trẽnh vi phãn thẽàng mã táĩ báĩi tãũp M-file YPRIME M, trãn khoaĩng tãĩ T0 âũũn Tfinal, vãĩi âĩãũ kiãũn âãũ Y0

[T, Y] = ODE23(F, T0, Tfinal, Y0, TOL, 1) đũng đũng sai TOL vaĩ hiãũn thẽ trãũng thãĩi trong khi tẽnh tẽch phãn

Âũĩ sãú nhãũp:

F - Chuãũi chãĩa tãn cuía mã táĩ báĩi toaĩn cuía ngẽãĩi đũng

Goũĩ: yprime = fun(t,y) vãĩi F = 'fun'

t - Thãĩi gian (vã hẽĩng)

y - Vectã cãũt lãĩi giaĩĩ

yprime - Vectã âãũo hàm trãĩ vãũ; yprime(i)

= dy(i)/dt

t0 - Giaĩ trẽ âãũo cuía t

```

tfinal- Giaỉ trẻ cuáúi cuía t
y0      - Giaỉ trẻ ááöu cuía vectả cäüt
tol      - Ääü chênh xaíc muáún ðùng (Ngáöm äênh: tol =
1.e-3)
trace - Năúu khaíc 0 thç in ra mäuì bæäic (Ngáöm
äênh: trace = 0)

```

Ääúi säú xuáút:

```

T - Äiäøm thài gian têch phán trái vảo (vectả cäüt)
Y - Lài giaíi trái vảo, mäuì vectả cäüt lài giaíi
trăn mäuì giaỉ trẻ tout (giaỉ trẻ xuáút trong t)
Kăút quai coí thăø äæäüc hiäøn thê bái: plot(tout,
yout)

```

ODE23P

ODE23P Giaíi caíc phæång trçnh vi phán bàòng phæång phaíp báüc tháúp vai hiäøn thê hçnh veí. Têch phán mäuì hăü phæång trçnh vi phán thæång bàòng caích ðùng cäng thæic Runge-Kutta báüc 2 vai 3 vai veí caích tênh thài gian cuía 2 hoàüc 3 pháön táí ááöu cuía lài giaíi

ODE23P('FUN', T0, Tf, Y0) phaít sinh mäuì hçnh veí mäuì phàóng 2 hoàüc 3 pha cuía lài giaíi hăü phæång trçnh vi phán mã táí bái tăüp M-file YPRIME.M trăn khoiàng T0 ääin Tf vai ðùng äiäöu kiäün ááöu Y0

Træäic khi göi ODE23P, thiăút láúp giăi haùn caíc truêch thêch hăüp văi

```

axis([y1min y1max y2min y2max]);
hold

```

hoàüc

```

axis([y1min y1max y2min y2max y3min y3max]);
hold

```

Răöi göi ODE23P('FUN',t0,tfinal,y0) văi

FUN - Tăn haìm xaíc äên phæång trçnh vi phán

yprime = fun(t,y) văi

t - Thài gian (vă hæäing)

y - Vectả cäüt lài giaíi

yprime - Vectả cäüt ääü haìm; yprime(i) =

dy(i)/dt

t0 - Giaỉ trẻ ááöu cuía t

tfinal- Giaỉ trẻ cuáúi cuía t

y0 - Giaỉ trẻ ááöu cuía vectả cäüt

```

tol - Ääü chênh xaíc muáún ðùng (Ngáöm äênh: tol =
1.e-3)

```

ODE45

Giaíi các phương trình vi phân phương phaíp bậc cao
 ODE45 Têch phân măt hâu phương trình vi phân thằìng
 bàòng cách dùng cằg thằìc Runge-Kutta bậc 4 vài 5
 [T,Y] = ODE45('yprime', T0, Tfinal, Y0) têch phân hâu
 phương trình vi phân thằìng mà tái bằi tập M-file
 YPRIME M, trằn khoằng tằi T0 ằằn Tfinal, vài ằằu
 kiằn ằằu Y0

[T, Y] = ODE45(F, T0, Tfinal, Y0, TOL, 1) dùng dùng
 sai TOL vài hiằn thằ traằng thằi trong khi tằn têch
 phân

Ầằi sằ nhằp:

F - Chuằi chằi tằn củi mà tái bằi toằn củi
 ngằi dùng

Goằ: yprime = fun(t,y) vài F = 'fun'

t - Thằi gian (và hằìng)

y - Vectằ củt lằi giaỉ

yprime - Vectằ ằằu hằn trằi vằ; yprime(i)

= dy(i)/dt

t0 - Giaỉ trằ ằằu củi t

tfinal- Giaỉ trằ củi củi t

y0 - Giaỉ trằ ằằu củi vectằ củt

tol - Ầằ chằn xằ muằn dùng (Ngằ ằn: tol =
 1.e-3)

trace - Nằu kằc 0 thằ in ra mằi bằìc (Ngằ
 ằn: trace = 0)

Ầằi sằ xuằt:

T - Ầằ thằi gian têch phân trằi vằ (vectằ củt)

Y - Lằi giaỉ trằi vằ, mằi vectằ củt lằi giaỉ
 trằ mằt giaỉ trằ tằt (giaỉ trằ xuằt trong t)

Kằt quằ cằ thằ ằằ hiằn thằ bằi: plot(tout,
 yout)

QUAD

Tằn giaỉ trằ sằ củi mằt têch phân, phương phaíp
 bậc thằp

Q = QUAD('F',A,B) xằp xằ têch phân củi F(X) tằi A
 ằằn B sai sằ tằng ằằi 1e-3. 'F' lài chuằi chằi
 tằn hằn. Hằn F phằi trằi vằ mằt vectằ củi giaỉ
 trằ xuằt nằu củi giaỉ trằ nhằp lài vectằ

Q = QUAD(F,A,B,TOL) têch phân sai sằ TOL

Q = Inf ằằ trằi vằ nằu gằp phằi ằằ quy, biằu
 hiằn têch phân kằi dằ

Q = QUAD(F,A,B,TOL,TRACE) têch phân sai sằ TOL vài
 lằn vằ củi ằằ lằng hằn vài mằt ằằ củi
 hằn bằ tằn têch phân

QUAD dùng phằi hằp quy tằc ằằ quy Simpson

Q = QUAD('F',A,B,TOL,TRACE,P1,P2,...) cho phằi củi
 hằ sằ P1, P2, ... ằằ truyằn trằi tiằp vào hằn
 F: G = F(X,P1,P2,...). Ầằ dùng giaỉ trằ ngằ
 ằn cho TOL hòu TRACE thằ cằ thằ truyằn vào mà
 trằn rằng ([])

QUAD8

Tên gọi trả về số của mặt tích phân, phải phải bậc cao

$Q = \text{QUAD8}('F', A, B)$ xấp xỉ tích phân của $F(x)$ tại A đến B sai số bằng 1e-3. 'F' là chuỗi chứa tên hàm. Hàm F phải trả về mặt vectơ các giá trị xuất nếu các giá trị nhập là vectơ

$Q = \text{QUAD8}(F, A, B, \text{TOL})$ tích phân sai số TOL

$Q = \text{Inf}$ nếu trả về nếu gặp phải lỗi quy, biểu thức tích phân kỳ

$Q = \text{QUAD8}(F, A, B, \text{TOL}, \text{TRACE})$ tích phân sai số TOL và luôn vẽ các lần lặp hàm với mặt tích phân

QUAD8 dùng phương pháp quy tắc 8 điểm Newton Cotes cho phép các hàm số P1, P2, ... được truy cập trực tiếp vào hàm F: $G = F(x, P1, P2, \dots)$. Để dùng giá trị trả về gồm ảnh cho TOL hoặc TRACE thì coi thử truy cập vào ma trận rỗng ([])

FMIN

Các giá trị hàm mặt biểu

$\text{FMIN}('F', x1, x2)$ tìm giá trị x là các giá trị của hàm $F(x)$ khoảng $x1 < x < x2$. 'F' là chuỗi chứa tên hàm mục tiêu cần tìm các giá trị.

$\text{FMIN}('F', x1, x2, \text{OPTIONS})$ dùng mặt vectơ các tham số ảnh hưởng. Nếu $\text{OPTIONS}(1)$ khác 0 thì các bước trung gian trong lần gọi ảnh hưởng; gồm ảnh $\text{OPTIONS}(1) = 0$. $\text{OPTIONS}(2)$ là dung sai cuối cùng với x; gồm ảnh bằng 1.e-4. $\text{OPTIONS}(14)$ là số lần nhất của các bước; gồm ảnh $\text{OPTIONS}(14) = 500$. Các thành phần khác của OPTIONS không được như các tham số ảnh hưởng nhập bởi FMIN

$\text{FMIN}('F', x1, x2, \text{OPTIONS}, P1, P2, \dots)$ cung cấp thêm 10 ảnh số để truy cập cho hàm mục tiêu, $F(x, P1, P2, \dots)$

Về đầu:

$\text{fmin}('cos', 3, 4)$ tên của pi đến vài chữ số thập phân

$\text{fmin}('cos', 3, 4, [1, 1.e-12])$ hiển thị các bước lấy ảnh tên của pi đến 12 chữ số thập phân

FMINs

Các giá trị hàm nhiều biến

$\text{FMIN}('F', X0)$ tìm vectơ x là các giá trị của hàm $F(x)$ gần với giá trị ảnh X0. 'F' là chuỗi chứa tên hàm mục tiêu cần tìm các giá trị. $F(x)$ phải là hàm lấy giá trị và hằng của ảnh số vectơ.

$\text{FMIN}('F', x1, x2, \text{OPTIONS})$ dùng mặt vectơ các tham số ảnh hưởng. Nếu $\text{OPTIONS}(1)$ khác 0 thì các bước trung gian trong lần gọi ảnh hưởng; gồm ảnh $\text{OPTIONS}(1) = 0$. $\text{OPTIONS}(3)$ là dung sai cuối cùng với $F(x)$; gồm ảnh bằng 1.e-4. $\text{OPTIONS}(14)$ là số lần nhất của các bước; gồm ảnh $\text{OPTIONS}(14) = 500$. Các thành phần khác của OPTIONS không được như các tham số ảnh hưởng nhập bởi FMIN

FMIN('F',x1,x2,OPTIONS,P1,P2,...) cung cấp thãm 10
 âäúi sấu âãø truyãn cho hàm muüc tiãu, F(X,P1,P2,...)
 FMINS dùng phãng phãip tçm ăn

FZERO

Tçm giáỉ trẽ 0 của hàm 1 biãún
 FZERO(F,X) Tçm giáỉ trẽ 0 của hàm f(x). 'F' lài
 chuãúi chãia tãn hàm muüc tiãu trái vãø giáỉ trẽ thãüc
 của mãüt âäúi sấu. X lài giáỉ trẽ ảoãn ban áãøu.
 Giáỉ trẽ trái vãø gãõn vãúi âiãøm mài hàm F thay áãøi.
 Vẽ duũ, FZERO('sin',3) lài pi. Lãu yĩ đáuú nhãiy ăn
 quanh sin. Thãng thãeing cãic hàm áãüc ănẽnh nhẽa
 trong tãúp M-file
 Mãüt áãúi sấu tuỹ choũn thãĩ ba áãút dung sai tãẽng
 áãúi cho viãüc kiãøm tra hãúi tuũ. Coĩ mãüt áãúi sấu
 tuỹ choũn thãĩ tã áãø in vãút của cãic bããic

FPLOTT

Veĩ áãø thẽ mãüt hàm
 FPLOTT(FNAME,LIMS) veĩ áãø thẽ hàm chẽ ănẽnh bãĩ biãún
 chuãúi FNAME giãia cãic giáỉi haũn trãn truũc x chẽ
 ănẽnh bãĩ LIMS = [XMIN XMAX].LIMS = [XMIN XMAX YMIN
 YMAX] cho cãic giáỉi haũn trãn truũc y vài lài tuỹ
 choũn
 Hàm FNAME phãĩ phũi hãúp vãúi chuáõn tãẽng cãüt của
 MATLAB. Vẽ duũ, nãuu FNAME lài hàm biãøu hiãũn bãĩ f
 = [f1(x) f2(x)]thç nhãúp '[x1 x2 x3]', hàm trái vãø ma
 trãũn

```
_f1(x1)_f2(x1)
f(x) =_f1(x2)_f2(x2)
_f1(x3)_f3(x3)
```

Cãich khãic, FNAME coĩ thãø lài chuãúi thay áãøi vãúi
 biãún x, nhã 'sin(x)', 'diric(x,10)' hoãuc
 '[sin(x),cos(x)]'. FPLOTT chãúp nhãũn 2 áãúi sấu tuỹ
 choũn lài MARKER vài TOL. MARKER lài chuãúi chẽ ănẽnh
 kiãøu ănẽnh đáuú dùng trong áãø thẽ. Kiãøu ngãøm ănẽnh
 lài '-'. Thãm vào kiãøu veĩ chuáõn, FPLOTT coĩn
 chãúp nhãũn '-+', '-x', '-o', '-*' (hoãuc '+-', 'x-',
 'o-', vài '*-'). TOL lài dung sai tãẽng áãúi. Ngãøm
 ănẽnh lài 2e-3. Sấu cãuc áãúi cãic bããic của x lài
 (1/TOL)+1

[X,Y] = fplot(FNAME,LIMS,...) trái vãø abscissae vài
 cãic toũa áãũ dùng áãø veĩáãø thẽ FNAME trong vectã
 cãüt X vài cãic cãüt của ma trãũn Y

Vẽ duũ:

```
fplot('sin',[0 4*pi])
fplot('sin(x)',[0 4*pi],'-+')
fplot('[sin(x),cos(x)]',[0 4*pi],'-x')
fplot('abs(exp(-j*x*(0:9)))*ones(10,1))',[0 2*pi],'-o')
fplot('tan',[-2*pi 2*pi -2*pi 2*pi],'-*')
fplot('[tan(x),sin(x),cos(x)]',[-2*pi 2*pi -2*pi 2*pi])
fplot('sin(1 ./ x)', [0.01 0.1],1e-3)
```

MATLAB lại mặt ngăn ngại láúp trạnh**SCRIPT**

Caïc tấúp nguyẫn baín M-file
 Măt tấúp nguyẫn baín lại tấúp bắn ngoàì coỉ năúi dung
 lại đăy caïc lăũnh cuía MATLAB. Bằong caích âaĩnh vaìo
 tắn tấúp thặ caïc lăũnh trong âoĩ âăăc thăc hiăũn
 tuăõn tắu. Tấúp nguyẫn baín phaĩ coỉ tắn mắi răũng
 ".m" vaì thăẽng goũì lại tấúp "M-file"

FUNCTION

Tấúp haìm M-file
 Caïc tấúp mắi coỉ thăõ thăm vaìo MATLAB năũu chuĩng
 âăăc taũ nhă caïc haìm âaĩ coỉ. Caïc lăũnh vaì haìm
 chăĩa haìm mắi phaĩ âăũt trong tấúp truĩng tắn haìm
 mắi vaì tắn mắi răũng lại '.m'. Doĩng âăũ tiăũn cuía
 tấúp phaĩ lại doĩng âẽnh nghẽa cuĩ phaĩp cho haìm
 mắi. Vẽ duũ, tấúp STAT.M trắn âẽa coỉ năúi dung:

```
function [mean,stdev] = stat(x)
    n = length(x);
    mean = sum(x) / n;
    stdev = sqrt(sum((x - mean).^2)/n);
```

âẽnh nghẽa haìm STAT âăũ tẽnh trẽ trung bẻnh vaì âăũ
 lăũch chuăõn cuía mắũ vectă . Caïc biăũn bắn trong
 haìm lại caïc biăũn cuĩc băũ

EVAL

Thăc hiăũn chuăũi kyĩ tắu chăĩa biăũu thăic MATLAB
 EVAL(s), s lài mắũ chuăũi, thặ MATLAB thăc hiăũn
 chuăũi nhă mắũ biăũu thăic hoăũc mắũ lăũnh
 EVAL(s1,s2) cho khaĩ năng bắũt lăũi. Nổi thi haĩnh
 chuăũi s1 vaì traĩ văũ năũu thao taĩc thănh căũg. Năũu
 thao taĩc bẻ lăũi thặ chuăũi s2 âăăc ăaĩc lăũũng
 trăăic khi traĩ văũ

[X,Y,Z,...] = EVAL(s) traĩ văũ caĩc âăũi săũ xuăũt tặ
 biăũu thăic trong chuăũi s
 Caĩc chuăũi nhăũp vaìo EVAL thăẽng âăăc taũ ra bằong
 caĩch năũi caĩc chuăũi con vaì caĩc biăũn bắn trong
 cằũp ngoăũc văũũg. Vẽ duũ: Phaĩt sĩnh đăy ma phăũũg
 coỉ tắn tặ M1 âăũn M12:

```
for n = 1:12
    eval(['M' num2str(n) ' = magic(n)'])
end
```

Chăũy mắũ tấúp nguyẫn baín M-file. Caĩc chuăũi taũ
 năũ caĩc doĩng cuía ma trăũn phaĩ cũĩng âăũ đăĩ

```
D = ['odedemo '
      'quaddemo '
      'fitdemo '];
n = input('Select a demo number: ');
eval(D(n,:))
```

Âoũc vaì ăaĩ lyĩ caĩc tấúp coỉ tắn data1.dat,
 data2.dat, ...

```
k = 0;
while 1
    k = k+1;
    datak = ['data' int2str(k)];
```

```

filename = [datak '.dat'];
if ~exist(filename), break, end
eval(['load ' filename]);
X = eval(datak);
% Xáỉ lyỉ đái liău ma tráun X
end

```

FEVAL

Ảáỉc lăăung hàỉm

Năúu F làỉ măút chuăúi chăỉa tăn hàỉm (thăằng âềnh nghéa trong măút M-file), thặ FEVAL(F,x1,...,xn) ảáỉc lăăung hàỉm ảỏỉ văỉ cáỉc ảắủỉ sắủ ảắỉ cho. Về đú, F = 'foo', FEVAL(F,9.64) giáúng nhắ foo(9.64) FEVAL thăằng ảắắủc đúng bắn trong cáỉc hàỉm lắủy tăn cáỉc cáỉc hàỉm khaỉc làỉm ảắủỉ sắủ

GLOBAL

Ăềnh nghéa cáỉc biăủn toàỉn cuűc

GLOBAL X Y Z ảềnh nghéa X, Y, vạỉ Z trong phaűm vỉ toàỉn cuűc. Thằng thăằng măúi hàỉm của MATLAB, ảềnh nghéa bàòng M-file, củỉ cáỉc biăủn cuűc bắỉ riăng của nỏỉ, chuűng ảắắủc phắn biăủt văỉ cáỉc biăủn của cáỉc hàỉm khaỉc vạỉ của cáỉc hàỉm trong vủng làỉm vỉăủc cả sắỉ vạỉ cáỉc nguyăn bắỉn kắng làỉ hàỉm

Túy nằn, năúu măút sắủ hàỉm, củỉ thắủ trong vủng làỉm vỉăủc cả sắỉ, mắỉ tắút cáỉ ảắủu khaỉ bắỉo măút tăn nằỏ ảỏỉ làỉ GLOBAL thặ tắút cáỉ chuűng ảắủu ảắắủc phắn bắủ măút bắỉn sao của biăủn ảỏỉ. Mỏúi lăủnh gắỉn cho biăủn ảỏỉ ảắỉ măút hàỉm thặ củỉ thắủ đúng trong tắút cáỉ cáỉc hàỉm khaỉc củỉ khaỉ bắỉo nỏỉ làỉ GLOBAL Vắủ phỏng cáỉch lắủp trệnh thặ cáỉc biăủn toàỉn cuűc ảắủu làỉ chắỉ hoa, nằng kắng bắủt bắủc

NARGCHK

NARGCHK Kỉăỏm tra sắủ ảắủỉ sắủ nắủp

NARGCHK (low,high,number) tráỉ vắủ thằng bắỉo lăúi năúu number kắng phắỉ gắỉa low vạỉ high. Năúu vắủy thặ tráỉ vắủ ma tráun rằng

LASTERR

LASTERR thằng bắỉo lăúi củắủỉ củng

LASTERR('') ảắủt lăúi hàỉm LASTERR ảắủ tráỉ vắủ ma tráun rằng cho ảắủn khaỉ xắỉy ra lăúi tíắủp theo. LASTERR thăằng đúng trong vỉăủc kắủt hắủp hai ảắủỉ sắủ trong đắủng của EVAL: EVAL('try','catch'). 'catch' củỉ thắủ xem xềỉt chuăúi LASTERR ảắủ xắỉc ảềnh củỉ gắỉy ra lăúi vạỉ lắủy hoắủt ảắủng tắằng ắỉng

Ăắắủu khaỉắn lắắủng**IF**

Thỉ hàỉnh lăủnh củỉ ảắắủu kắắủn

Đắủng chung của cáủ lăủnh IF làỉ:

```
IF variable, statements, END
```

Các câu lệnh statements được thi hành nếu phần thực của biểu variable có tất cả các phần tại khác không. Biểu thường dùng kết quả của các toán tại quan hệ là `==`, `<`, `>`, `<=`, `>=`, hay `~=`

Vê du:

```

        if i==j
            A(i,j) = 2;
        elseif abs (i-j) == 1
A(i,j) = -1;
        else
            A(i,j) = 0;
        end

```

ELSE

Dùng với IF

ELSEIF

ELSEIF Dùng với IF nhưng khác cần END nhưng ELSE IF thực phải có END

END

Kết thúc các câu lệnh FOR, WHILE và IF. Khác có END thực FOR, WHILE và IF phải nhập thêm. Mỗi END được ghép cặp với FOR, WHILE, IF gần nhất mà chưa có cặp mở kết thúc phạm vi của lệnh

FOR

Câu lệnh lặp có chế độ số lần lặp. Dùng chung của lệnh FOR là:

```

        FOR variable = expr, statement, ..., statement
END

```

Các câu của biểu thực expr được lưu vào biểu từng bước và các lệnh được thi hành, Biểu thực thường có dạng X:Y, trong đó các câu của nó là các vế hằng số. Với về du (giá trị N và biểu được gần một giá trị nhỏ)

```

for i=1:n,
    for j=1:n,
        A(i,j) = 1/(i+j-1);
    end
end

```

```

for s = 1.0: -0.1: 0.0, end các bước của s thay đổi -0.1 (giá trị)

```

```

for E = eye(N), ... end tạo E vào các vectơ đơn vị N chiều

```

WHILE

Câu lệnh lặp với số lần lặp khác xác định

Dùng chung của lệnh WHILE là:

```

WHILE variable, statement, ..., statement, END

```

Các lệnh statement, ..., statement được thi hành trong khi biểu có tất cả các phần tại của variable khác không. Về du (giá trị A và biểu được tăng)

```

        E = 0*A; F = E + eye (E); N = 1;

```

```
while norm(E+F-E,1) > 0, E = E + F;  
    F = A*F/N;  
    N = N + 1;  
end
```

BREAK

Kiểm tra việc thoát khỏi vòng lặp. Nếu tạo ra các vòng lặp thực BREAK sẽ thoát khỏi lần lặp lần cuối trong nhóm

RETURN

Trả về lần gọi hàm

ERROR

Hiển thị thông báo vai trò qua hàm

ERROR('MSG') hiển thị và báo MSG vai trò ra lần thoát trong một M-file. Nếu chuỗi lại mà trước rằng thực hành hoạt động

INPUT

Nhà nhập dùng nhập dữ liệu

INPUT('prompt:') cho một câu nhắc 'prompt:' và chỉ nhập dùng nhập tài bàn phím. Dữ liệu nhập có thể lại một biểu thức MATLAB, nó sẽ được tính, dùng các biểu thức trong vùng làm việc, và lại kết quả trả về của hàm

INPUT(' prompt:', 's') cho một câu nhắc 'prompt:' và chỉ nhập dùng nhập tài bàn phím. Dữ liệu nhập sẽ được tính toán; các ký tự trả về sẽ ẩn giá trị một chuỗi ký tự của MATLAB

KEYBOARD

KEYBOARD gửi bìa thêm nầu lại tấp nguyãn bảin. Khi ầut trong tấp M-file, KEYBOARD ngợng thì hảnh tấp vại trái ầiầu khiầon vầ bảin thêm. Ầut biầut lại ầầut biầu hiầun bầong dầu nhầoc keip. Cầc biầun coỉ thầ ầầut xem xeit hay thay ầầi - mầi lầnh MATLAB ầầu hầp lầ. Chầ ầầ bảin thêm kầut thuic khi vầo lầnh RETURN (nghéa lại ầầnh vầo 6 chầ R-E-T-U-R-N vầ ần thêm ENTER). Ầầu khiầon trái vầ lầnh gửi M-file

MENU

Phầt sinh mầt menu cầc lầu choïn cho ngầi dùng nhầp

K = MENU('Choose a color','Red','Blue','Green') hiầun trầ mầ hợnh

----- Choose a color -----

- 1) Red
- 2) Blue
- 3) Green

Choïn mầt sầ

Sầ nhầp bầi ngầi dùng ầầut trái vầ. Sầ lầnh nhầut cầc ầầ mầc lại 32

PAUSE

Tầ dầing

PAUSE chầi ngầi dùng ần thêm bầut kyì

PAUSE(n) tầ dầing n giầ

PAUSE OFF biầu hiầun mầi lầnh PAUSE hoầc PAUSE(n) khầ thầc hiầun. Lầnh nầy ầầ cầc nguyãn bảin chầy tầ ầầing

PAUSE ON ngầut lầi

UIMENU

UIMENU tầ menu giao điầu ngầi dùng

UIMENU('PropertyName1',value1,'PropertyName2',value2,...) tầ menu trầ menu bar ầi trầ ầềnh cuía cầa sầ hợnh ầnh hiầun thầi vại trái vầ mầt thei cho nầ

UIMENU(H,...) tầ mầt menu mầi vầi H lại cha. H phầi lại mầt thei hợnh ầnh hoầc menu. Nầu H lại thei hợnh ầnh thợ UIMENU menu trầ menu bar ầi trầ ầềnh cuía cầa sầ. Nầu H lại mầt thei menu trầ menu bar thợ menu mầi lại menu thầp hầ ầầ mầc cha trầ menu bar

Cầc ầầut tềnh cuía menu coỉ thầ ầầut ầi lầ tầ ra bầong cầch dùng tầing cầup ầầi sầ PropertyName/PropertyValue trong UIMENU, hoầc thay ầầi sau ầi bầong cầch dùng lầnh SET Thầc hiầun GET(H) ầầ xem danh sầch cầc ầầut tềnh cuía UIMENU vại cầc giầ trầ hiầun thầi cuía nầ Thầc hiầun SET(H) ầầ xem danh sầch cầc ầầut tềnh cuía UIMENU vại cầc giầ trầ hầp lầ

UICONTROL

UICONTROL taôo âiãöu khiãön giao điãün ngæãi dùng

UICONTROL

('PropertyName1',value1,'PropertyName2',value2,...)

taôo âiãöu khiãön giao điãün ngæãi dùng trong cæía sãø hçnh áính hiãün thãi vai trái vãö mãüt theí cho noĩ

Caïc ââuuc tênh cuía menu coĩ thãø âæãuc ââuut áí luĩc taôo ra bàòng caĩch dùng tæng cầup ââuĩ sãu PropertyName/PropertyValue trong UICONTROL, hoàuc thay ââuĩ sau âoĩ bàòng caĩch dùng lãũnh SET. Thæuc hiãün GET(H) âãø xem danh saĩch caĩc ââuuc tênh cuía UICONTROL vai caĩc giaĩ trẽ hiãün thãi cuía noĩ. Thæuc hiãün SET(H) âãø xem danh saĩch caĩc ââuuc tênh cuía UICONTROL vai caĩc giaĩ trẽ hãũp lãũ

Hàm vãö âa thæic**ROOTS**

Tçm caĩc nghiãũm cuía âa thæic

ROOTS(C) tênh nghiãũm cuía âa thæic coĩ caĩc hãũ sãu laĩ caĩc pháön tæĩ cuía vectã C. Nãũu C coĩ N+1 pháön tæĩ thç âa thæic laĩ $C(1)*X^N + \dots + C(N)*X + C(N+1)$

POLY

Âa thæic ââuuc træng

Nãũu A laĩ ma tráũn vuãng cáup N, thç POLY(A) laĩ vectã ðoĩng gãõm N+1 pháön tæĩ laĩ caĩc hãũ sãu cuía âa thæic ââuuc træng, $\text{DET}(\text{lambdã}*\text{EYE}(A) - A)$. Nãũu V laĩ vectã, thç POLY(V) laĩ vectã coĩ caĩc pháön tæĩ laĩ caĩc hãũ sãu cuía âa thæic coĩ caĩc nghiãũm trong V. Vãĩi caĩc vectã, ROOTS vai POLY laĩ caĩc hàm ngæãuc nhau vãö thæĩ tæũ, chia tè lãũ vai sai sãu

ROOTS(POLY(1:20)) phaĩt sinh vẽ ðuũ nãøi tiãúng cuía Wilkinson

POLYVAL

POLYVAL Æaĩc læũng âa thæic

Nãũu p laĩ mãüt vectã ââu dài d+1 coĩ caĩc pháön tæĩ laĩ caĩc hãũ sãu cuía mãüt âa thæic, thç

$y = \text{POLYVAL}(p,x)$ laĩ giaĩ trẽ cuía âa thæic tênh taũĩ x

$$y = p(1)*x^d + p(2)*x^{(d-1)} + \dots + p(d)*x + p(d+1)$$

Mãũu X laĩ mãüt ma tráũn hay vectã, thç âa thæic âæãuc æaĩc læũng taũĩ táút cáĩ caĩc âiãøm trong X

[y,delta] = POLYVAL(p,x,S) dùng giaĩ trẽ xuát tuĩ choũn phaĩt sinh báĩ POLYFIT âãø âaĩnh giaĩ caĩc sai sãu, $y \pm \text{delta}$. Nãũu caĩc sai sãu trong sãu liãũu nhãũp vaĩo POLYFIT âæuc lãũp vãĩi biãũn hàòng, thç $y \pm \text{delta}$ chæĩa êt nhãũt 50% caĩc giaĩ trẽ ðæũ âoain

POLYVALM

POLYVALM Mãic læąung ăa thăic ma tráun

Năuu V lài vectă coĩ caĩc pháon tăi lài caĩc hău său cuĩa ăa thăic, thç POLYVALM(V,X) lài giaĩ trẽ cuĩa ăa thăic taũĩ ăăũĩ său ma tráun X. Xem POLYVAL văĩi caĩch tênh ăa thăic trong trăəng hăp ăăũĩ său thăəng hoăuc maĩng

RESIDUE

RESIDUE Tênh khai triăon təng pháon hay thăung dă

[R,P,K] = RESIDUE(B,A) tçm thăung dă, căuc vại hăəng cuĩa khai triăon təng pháon cuĩa chia ăa thăic, B(s) vại A(s). Năuu khăng coĩ nghiăum keĩp,

$$K(s) = \frac{B(s)}{A(s)} = \frac{R(1)}{s - P(1)} + \frac{R(2)}{s - P(2)} + \dots + \frac{R(n)}{s - P(n)}$$

Caĩc vectă B vại A lài caĩc hău său cuĩa caĩc ăa thăic theo luỹ thăia thoăĩ cuĩa s. Thăung dă traĩ văo trong vectă căut R, caĩc vẽ trẽ căuc trong vectă căut P, vại caĩc hăəng trong vectă doĩng K. Său caĩc căuc lài $n = \text{length}(A) - 1 = \text{length}(R) = \text{length}(P)$

Vectă hău său hăəng lài răung năuu $\text{length}(B) < \text{length}(A)$; ngăəuc laũĩ thç

$$\text{length}(K) = \text{length}(B) - \text{length}(A) + 1$$

Năuu $P(j) = \dots = P(j+m-1)$ lài măut căuc cuĩa multplicity m, thç khai triăon bao găom caĩc haũng tăi thuăuc dăung

$$\frac{R(j)}{(s - P(j))} + \frac{R(j+1)}{(s - P(j))^2} + \dots + \frac{R(j+m-1)}{(s - P(j))^m}$$

[B,A] = RESIDUE(R,P,K), văĩi 3 ăăũĩ său nhăup vại 2 ăăũĩ său xuăut, chuyăon khai triăon phán său təng pháon thăinh caĩc ăa thăic văĩi caĩc hău său trong B vại A.

Khuyăun caĩo: Văo măut său, khai triăon phán său caĩc ăa thăic lài biăou hiăũn cuĩa măut bài toăĩn yău. Năuu ăa thăic thăic mău, A(s), găon ăa thăic coĩ nghiăum băũĩ, thç caĩc thay ăăoĩ nhoĩ trong său liăũu, kăo cáĩ sai său, coĩ thăo làm thay ăăoĩ lăĩn trong caĩc căuc vại thăung dă kăut quăĩ. Caĩc căng thăic cuĩa bài toăĩn làm cho viăuc dùng caĩch biăou điăun căuc trăung hoăuc 0 lài thêch hăp hăn

POLYFIT

POLYFIT Khăĩp ăăəng cong ăa thăic

POLYFIT(x,y,n) tçm caĩc hău său cuĩa ăa thăic p(x) báuc n ăăo khăĩp său liăũu, $p(x(i)) \sim y(i)$, theo phăang phaĩp bçnh phăang beĩ nhăut

[p,S] = POLYFIT(x,y,n) traĩ văo caĩc hău său ăa thăic p vại ma tráun S ăăo dùng văĩi POLYVAL ăăo cho ra caĩc ăăĩc læąung sai său trong caĩc đău ăoăĩn. Năuu caĩc sai său trong său liăũu, y, ăăuc lăup văĩi biăũn

hàòng, thợ POLYVAL seỉ cho ra caỉc giảỉi haủn sai sảủ
chæỉa êt nhấủt 50% cuỉa đæủ àoạỉn

POLYDER

POLYDER Âaûo haìm âa thaïc

POLYDER(P) traí vảo âaûo haìm cuía âa thaïc coi caïc hâu sáu lai caïc pháõn táí cuía vectả P POLYDER(A,B) traí vảo âaûo haìm cuía âa thaïc $A*B$. $[Q,D] = \text{POLYDER}(B,A)$ traí vảo âaûo haìm cuía âa thaïc thẳng B/A , biểu diểu thẳng Q/D

CONV

Têch chấu vai nhán âa thaïc

$C = \text{CONV}(A, B)$ têch chấu caïc vectả A vai B. Vectả kút quái coi âaû dài $\text{LENGTH}(A) + \text{LENGTH}(B) - 1$. Nẫu A vai B lai caïc vectả lai caïc hâu sáu cuía âa thaïc, thç têch chấu chẳg tẳng âẳng nhán hai âa thaïc

DECONV

Taích chấu vai chia âa thaïc

$[Q,R] = \text{DECONV}(B,A)$ taích chấu caïc A ra B. Kút quái traí vảo trong vectả Q vai pháõn đẳ trong vectả R âa $B = \text{conv}(Q,A) + R$

Nẫu A vai B lai caïc vectả lai caïc hâu sáu cuía âa thaïc, thç taích chấu tẳng âẳng vắi chia âa thaïc. Kút quái chẳ B cho A âẳ thẳng Q vai đẳ R

Nẫui suy sáu liẫu**INTERP1**

INTERP1 nẫui suy sáu liẫu 1 chiểu (tra cẩu baíng)

$YI = \text{INTERP1}(X,Y,XI)$ traí vảo vectả YI chẳa caïc pháõn táí tẳng ẳg vắi caïc pháõn táí trong XI vai xẳc ẳnh bắi pheỷ nẫui suy bẳn trong caïc vectả X vai Y

Pheỷ nẫui suy cẳg thao táic nhẳ " tra cẩu baíng ". Mẳ táí trong caïc sáu haỷng " tra cẩu baíng ", "baíng" lai $[X,Y]$ vai INTERP1 " caïc tra cẩu " lai caïc pháõn táí cuía XI trong X, vai, đẳu vào caïc vẳ trẳ cuía chẳg, caïc giẳ trẳ traí vảo YI nẫui suy bẳn trong caïc pháõn táí cuía Y

INTERP1 thẳc hiểu xuẩt nhiểu giẳ trẳ tra cẩu baíng nẫu Y lai ma trẩn. Nẫu Y lai ma trẩn coi $\text{length}(X)$ đẳg, vai N cẩt, thç INTERP1 traí vảo ma trẩn YI cẩ $\text{length}(XI) \times N$ chẳa nhiểu kút quái xuẩt cuía tra cẩu baíng

Ngắo ẳnh, INTERP1 đẳg nẫui suy tuyẩn tẳnh

$YI = \text{INTERP1}(X,Y,XI, 'method')$ chẳ ẳnh caïc phẳng phaỷ kẳc

Caïc phẳng phaỷ coi thẳ lai:

'linear' - nẫui suy tuyẩn tẳnh

'spline' - nẫui suy spline lắu phẳng

'cubic' - nẫui suy lắu phẳng

Tẩt cá caïc phẳng phaỷ nẫui suy ỏi hoỷ X phắi lai ẳn ẳiẫu. Phẳng phaỷ lắu phẳng 'cubic' cẳg ỏi hoỷ caïc ẳiẫm trong X phắi cẳch ẳu

Vẳ đẳ, phẩt sinh ẳẳg cẳg hỷn sin thẳ vai nẫui suy trẳ mẩt hoẳnh ẳu tẩt hẳ:

```
x = 0:10; y = sin(x); xi = 0:.25:10; yi =
interp1(x,y,xi); plot(x,y,'o',xi,yi)
```

INTERP2

INTERP2 năüi suy săú liăöu 2 chiăöu (tra căü baíng)
 ZI = INTERP2(X,Y,Z,XI,YI) traí văö ma tráün ZI chăüa
 cáic pháön táí tăng äíng văü cáic pháön táí cuía XI
 vài YI vài xăic äënh bàong pheíp năüi suy băn trong
 hăim 2 chiăöu mã táí báü cáic ma tráün X, Y, vài Z.
 Cáic giaí trê băn ngoăüi miăön traí văö NaN. X coí
 thăö lài măüt vectă ðoíng, trong trăäng hăüp năüi
 äăüc giaí thiăút äíp ðuöng vào cáic căüt cuía Z.
 Tăng tăü, Y coí thăö lài vectă căüt vài cáic pháön táí
 cuía năüi äăüc giaí thiăút äíp ðuöng qua cáic ðoíng
 cuía Z.

Pheíp năüi suy cüíng thao táic nhă "table lookup". Mã
 táí trong cáic săú hăung " tra căü baíng ", "baíng"
 lài TAB = [NaN,Y; X,Z] vài INTERP2 "cáic tra căü "
 lài cáic pháön táí cuía XI trong X, YI trong Y, vài,
 ðăüa vào vê trê cuía chuíng, cáic giaí trê traí văö ZI
 năüi suy băn trong cáic pháön táí cuía Z

Ngăöm äënh, INTERP2 ðuöng pheíp năüi suy song tuyăún
 tên

```
ZI = INTERP2(X,Y,Z,XI,YI,'method') chë äënh phăäng  
phaíp
```

Cáic phăäng phaíp coí thăö lài:

- 'linear' or 'bilinear' - năüi suy song tuyăún tên
- 'cubic' or 'bicubic' - năüi suy song láúp phăäng
- 'nearest' - năüi suy căün găön nhăüt

Tăüt cáí cáic pheíp năüi suy äăöu äoüi hoíi X vài Y
 lài äân äiăüu vài lăäüi vüăng (năüi chuíng äăüc tăü
 ra bàong cáich ðuöng MESHGRID). Khoaíng cáich biăún
 äăüi äăüc xăi lyí bàong äính xău cáic giaí trê äăi cho
 trong X,Y,XI, vài YI vào măüt miăön giaí trê cáich
 äăöu trăăc khi năüi suy

INTERPFT

INTERPFT Năüi suy 1 chiăöu bàong cáich ðuöng phăäng
 phaíp FFT (biăún äăüi Fourier nhanh)

Y = INTERPFT(X,N) traí văö vectă Y äăü ðăüi N nhăün
 äăüc bàong cáich năüi suy trong biăön äăüi Fourier
 cuía X. Giaí săi x(t) lài hăim tuăön hoăn theo t
 văüi chu kyí p, tăü măüu táí cáic äiăöm cáich äăü,
 $X(i) = x(T(i))$ văüi $T(i) = (i-1)*p/M$, $i = 1:M$, $M =$
 $\text{length}(X)$. Răüi y(t) lài măüt hăim tuăön hoăn khaíc
 cüíng chu kyí vài $Y(j) = y(T(j))$ văüi $T(j) = (j-1)*p/N$,
 $j = 1:N$, $N = \text{length}(Y)$. Năüi N lài măüt báüi nguyăn
 cuía M, thç $Y(1:N/M:N) = X$

Năüi X lài măüt ma tráün, pheíp năüi suy äăüc thăüc
 hiăün theo tăng căüt

INTERPFT, khăng tham săú, lài măüt chăäng trçnh măüu

GRIDDATA Læåïï dæî liãûu

[XI,YI,ZI] = GRIDDATA(X,Y,Z,XI,YI) traí vảo XI và YI taỏu nỏn cỏch nỏy, ỏỏi lại giỏỏng cỏc ma traỏu traí vảo bỏỏi MESHGRID. GRIDDATA đủng phỏỏng phỏỏp kỏỏỏng cỏch ỏỏỏỏ

SPLINE

PP = spline(x,y) traí vẫ đường-pp của phép nối suy
spline lát phẳng, ấở sau ấở dùng vẫ ppval,
v.v...

```
v = ppval(pp,xx)   traí vãõ giaĩ trẽ pp cuía haìm pp
taûi xx
```

STRINGS

Vào lầnh "help strfun" cho danh sách các hàm xử lý chuỗi

Láúy giaï trẽ tuyãút âäúi vai chuyãøn chuäúi sang sãú

ABS(X) lài giáị trẽ tuyăút âăúi cuía caïc pháön tặi cuía X. Năúu X pháic thặ ABS(X) lài mắun cuía caïc pháön tặi cuía X. Năúu S lài biăún chuăúi MATLAB thặ ABS(S) traí văö caïc giáị trẽ săú ASCII thăúp phán cuía caïc kyị tặu trong chuăúi. Khăng thay âăúi biăöu hiăún bắn trong mắi chề thay âăúi caích in ra

SETSTR

Xem caïc giáị trẽ săú lài chuăúi

X = SETSTR(X) coi thăö dùng cho caïc vectă chặi caïc săú nguyăn tặi 0 âăún 255 âăö biăöu hiăún mắ ASCII cho vắn bắn. Hắm traí văö vectă âăăc hiăön thề nhặ vắn bắn chặi khăng lài caïc săú nguyăn

ISSTR

Ăuăng năúu lài chuăúi

ISSTR(S) traí văö 1 năúu lài chuăúi vắn bắn, ngắăc lăúi traí văö 0

BLANKS

Măüt chuăúi gắm caïc kyị tặu trắng

BLANKS(n) lài măüt chuăúi gắm n kyị tặu trắng. Dùng vắi DISP, nhặ DISP(['x' BLANKS(20) 'y'])

DISP(BLANKS(n)') đỉ chuyăön con troí xuăúng n đòing

DEBLANK

DEBLANK Xổi caïc kyị tặu trắng ái cuăúi măüt chuăúi

DEBLANK(S) Xổi caïc kyị tặu trắng ái cuăúi măüt chuăúi vắi mắi kyị tặu null trong chuăúi S. Kyị tặu null lài măüt kyị tặu coi trẽ tuyăút âăúi bắng 0

STR2MAT

STR2MAT Taôu đăung ma trắn vắn bắn tặi caïc chuăúi riăng biăút

S = STR2MAT(T1,T2,T3,...) taôu ma trắn S chặi caïc chuăúi vắn bắn T1,T2,T3,... lài caïc đòing. Tặ âăăng thắ mắi chuăúi caïc kyị tặu trắng nhằm taôu ra măüt ma trắn hắp lyị. Cói thăö dùng âăún 10 chuăúi âăö taôu S. Mắi tham săú vắn bắn, Ti, coi thăö chềnh nố lài măüt ma trắn chuăúi. Caích nắy cho pheíp taôu ra caïc ma trắn chuăúi lắn tuy yị

EVAL

Thăc hiăún chuăúi chặi biăöu thăc MATLAB

EVAL(s), vắi s lài măüt chuăúi, lắm cho MATLAB thăc hiăún nhặ măüt biăöu thăc hoăc măüt lắnh

EVAL(s1,s2) cung cáúp khắ nắg bắt lắi. Nố thăc hiăún s1 vắ traí văö năúu thao tắc thắnh cắng. Năúu thao tắc bắ lắi thặ s2 âăăc ắắc lắăng trắắc khi traí văö. Nghề nhặ lắnh EVAL('try','catch')

[X,Y,Z,...] = EVAL(s) traí văö caïc âăúi săú xuăút tặi biăöu thăc trong chuăúi s. Caïc chuăúi nhắp vắo EVAL thắăng âăăc taôu ra bắng caích nắi caïc chuăúi con vắ caïc biăún bắn trong cắp ngoăc vắng. Vồ đú: Phắt sinh măüt đắy ma trắn coi tắn tặi M1 âăún M12:

```

        for n = 1:12 ,                      eval(['M' num2str(n) ' =
magic(n)']) ,                               end
        Châuy măt tấp M-file nguyăn bản.  Căc chuăi taô
        nă căc dòng của ma trắn D phắi cùng âă dài
        D = ['odedemo '
              'quaddemo'
              'fitdemo '];
        n = input('Select a demo number: ');
        eval(D(n,:))
        Âuc vắ xắ lý căc tấp data1.dat, data2.dat, ...
        k = 0;
        while 1
            k = k+1;
            datak = ['data' int2str(k)];
            filename = [datak '.dat'];
            if ~exist(filename), break, end
            eval(['load ' filename]);
            X = eval(datak);
            % Xắ lý đắ liău trong ma trắn X
        end

```

STRCMP

So sắnh chuăi

STRCMP(S1,S2) trắ vắ 1 năú S1 vắ S2 giắng nhau vắ
ngắc lắ thắ trắ vắ 0

FINDSTR

FINDSTR Tắ măt chuăi trong chuăi kắc

K = FINDSTR(S1,S2) trắ vắ căc chề sắ bắt áău của
mỗi xuát hiău S2 trong S1
Vê du:

```

s = 'How much wood would a woodchuck chuck?';
findstr(s,'a')      trắ vắ 21
findstr(s,'wood')   trắ vắ [10 23]
findstr(s,'Wood')   trắ vắ []
findstr(s,' ')      trắ vắ [4 9 14 20 22 32]

```

UPPER

Chuyăon chuăi sang chắ hoa

A = upper(a) chuyăon mỗi kắ tắ thắng trong A
thắnh kắ tắ hoa tắng ắng, căc kắ tắ kắc kắng
thay áăi

LOWER

Chuyăon chuăi sang chắ thắng

b = lower(A) chuyăon mỗi kắ tắ hoa trong A thắnh
kắ tắ thắng tắng ắng, căc kắ tắ kắc kắng
thay áăi

ISLETTER

ISLETTER ắng năú lắ chắ cắ

Vắi măt chuăi S, thắ ISLETTER(S) lắ 1 cho căc chắ
cắ, cò lắ lắ 0

ISSPACE

ISSPACE âüĩng năúu lại kyĩ tău trăúg, sang ðoĩng, văö âăöu ðoĩng, tab, tab âăĩng, hay sang trang. Văĩi chuăüi S, thç ISSPACE(S) lại cho caĩc kyĩ tău trăúg vai 0 cho kyĩ tău khaĩc

STRREP

STRREP tçm vai thay thăü chuăüi

S = STRREP(S1,S2,S3) thay thăü S2 trong S1 văĩi chuăüi măĩi S3

Vê ðu:

```
s1='This is a good example';
strrep(s1,'good','great')   traĩ văö 'This is a
great example'
strrep(s1,'bad','great')    traĩ văö 'This is a
good example'
strrep(s1','','great')      traĩ văö 'This is a
good example'
```

STRTOK

Tçm ðăúu hiăüu trong chuăüi (ðăúu hiăüu lại kyĩ tău khăng lại chăĩ caĩi hay săü)

STRTOK(S,D) traĩ văö ðăúu hiăüu âăöu tiăĩn trong S âăăüc âăĩnh ðăúu băĩi D. Năĩ cuĩng coĩ caĩc traĩ văö tuĩy choũn lại phăĩn coĩn laũi cuĩa chuăüi găúg. D lại măüt măĩng caĩc kyĩ tău ðăúu hiăüu. Năúu khăng cho ðăúu hiăüu thç giaĩ thiăút lại kyĩ tău trăúg

NUM2STR

NUM2STR Chuyăĩn săü sang chuăüi

T = NUM2STR(X) chuyăĩn săü vă hăĩĩng X sang chuăüi biăĩu hiăũn trong T văĩi khoaĩng 4 chăĩ săü vai coĩ măüt phăĩn muĩ năúu cáĩn. Hăĩm năĩy thăĩĩng ðuĩng âăĩ âăĩnh nhăĩn trong caĩc hçnh veĩ văĩi caĩc lăũnh TITLE, XLABEL, YLABEL vai TEXT. Măüt âăüĩ săü tuĩy choũn coĩ thăĩ cho âăĩ biăĩu hiăũn măüt âăü chçnh xăĩc thay âăüi
T = NUM2STR(X,PREC) chuyăĩn săü vă hăĩĩng X sang chuăüi văĩi măüt âăü chçnh xăĩc tăüi âă chç âçĩnh băĩi PREC

INT2STR

INT2STR Chuyển số nguyên sang chuỗi

S = INT2STR(N) chuyển số nguyên và hằng N sang chuỗi

STR2NUM

STR2NUM chuyển chuỗi sang số

X = STR2NUM(S) chuyển chuỗi S, nấn lại bộ nhớ ký tự ASCII của mặt giá trị số, sang số của MATLAB. Chuỗi có thể chứa các chữ số, dấu chấm thập phân, mặt dấu dấu âm đầu là + hoặc -, mặt chữ 'e' cho mũ của 10, và 'i' cho nhân về phức

STR2NUM chuyển ma trận chuỗi sang ma trận số về đầu

```
A = ['1 2'          str2num(A) => [1 2;3 4]
      '3 4']
```

Nếu chuỗi S không là biểu thức của ma trận hoặc số thì thực STR2NUM(S) trả về ma trận rỗng

HEX2NUM

HEX2NUM IEEE chuyển số dùng thập lục sang dùng số thập kỷ

HEX2NUM(S), với S là chuỗi 16 ký tự chứa số dùng hex thực trả về số thập kỷ chuẩn IEEE Nếu ít hơn 16 ký tự thực thêm vào bên phải với các số 0

Về đầu HEX2NUM('400921fb54442d18') trả về Pi

HEX2NUM('bff') trả về -1

Các NaN, inf và các số hữu tỉ tên cũng

HEX2DEC

HEX2DEC Chuyển số dùng hex sang dùng thập phân

HEX2DEC(D) trả về số dùng hex D trong dùng thập phân.

Về đầu HEX2DEC('12B') và HEX2DEC('12b') cái hai trả về 299

DEC2HEX

DEC2HEX Chuyển số thập phân sang dùng hex

DEC2HEX(D) trả về số nguyên thập phân D về dùng hex. Về đầu, DEC2HEX(2748) trả về 'ABC'

SPRINTF

SPRINTF Ghi dữ liệu có dùng thực vào mặt chuỗi

[S,ERRMSG] = SPRINTF(FORMAT,A,...) lệnh dùng dữ liệu trong ma trận A (và trong mỗi ô số ma trận thêm nữa), để khi nào lệnh trong chuỗi lệnh dùng FORMAT, và trả về nội dung biểu chuỗi S của MATLAB. ERRMSG là ô số tùy chọn để trả về mặt chuỗi thông báo nếu có và mặt ma trận rỗng nếu không

SPRINTF giống như FPRINTF ngoài trả về dữ liệu trong biểu chuỗi chữ không ghi vào tệp. FORMAT là mặt chuỗi chứa các lệnh chuyển kiểu của ngăn ngắt C. Các lệnh chuyển kiểu bao gồm ký tự %, cải, để in ra và số chữ

sáu thập phân, chế độ hiển thị con, vài các ký tự chuyển đổi d, i, o, u, x, X, f, e, E, g, G, c, và s. Xem giáo trình C để coi đầy đủ chi tiết. `SPRINTF` như trong ANSI C với một số các ngoại lệ vài mã riêng,

Bao gồm:

1. Nếu MATLAB không chuyển đổi chế độ hiển thị sang kiểu chế độ hiển thị đúng e để dễ dàng. Phải chuyển đổi kiểu rồi rằng các giá trị không nguyên của MATLAB sang giá trị nguyên nếu muốn dùng chế độ hiển thị chuyển đổi nguyên như d vài lược cách xử lý của ANSI C

2. Các chế độ hiển thị con không chuẩn sau đây để cung cấp cho các ký tự chuyển đổi o, u, x, và X

t - Kiểu cả bán của C là thực chế độ không

b - Kiểu cả bán của C là thực kép chế độ không phải là nguyên không đúng

Về đầu, để in ra đúng hex một giá trị thực kép dùng một đúng như '%bx'

`SPRINTF` khác với hàm tương tự trong C một điều quan trọng là - nó "vectơ hóa" với trường hợp A là và hằng. Chuỗi để đúng để của vòng qua các phần tử của A cho đến khi tất cả các phần tử để dùng hết. Rồi nó của vòng theo cách tăng từ mà không bắt đầu lại qua mỗi lần sáu mà tăng tiếp theo

Về đầu, như

```
S = sprintf('rho is %5.3f',(1+sqrt(5))/2)
```

cho ra chuỗi

```
S = 'rho is 1.618'
```

SSCANF

Để chuỗi để điều khiển khi nào coi đúng thực

`[A,COUNT,ERRMSG,NEXTINDEX] = SSCANF(S,FORMAT,SIZE)` để để liêu tài biến chuỗi MATLAB S, chuyển đổi để vào chuỗi để hiển thị đúng FORMAT, vài trái với trong mà trước A. COUNT là một để sáu xuất tùy chọn để trái với sáu phần tử để thành cần. ERRMSG là một để sáu xuất tùy chọn để trái với một chuỗi tăng ba lần nếu có lần hoặc một mà trước rằng nếu không lần. NEXTINDEX là một để sáu tùy chọn chế độ hiển thị một hoặc nhiều lần sáu ký tự để que trong S.

`SSCANF` giống như `FSCANF` ngoại lệ trừ để để liêu trong biến chuỗi MATLAB chế độ không để tài tập. SIZE là tùy chọn; nó để một giá trị hạn vào sáu phần tử coi thấy que tài chuỗi; nếu không chế độ hiển thị để xem là toàn bộ chuỗi; nếu coi thực các giá trị tiếp theo

N để nhiều nhất N phần tử vào một vectơ cột

inf để để của chuỗi

`[M,N]` để nhiều M * N phần tử lược để hết nhất một mà trước MxN, theo thực tập cột. N coi thấy là inf, nhưng M thực không. Nếu mà trước A coi kết

quái chè tài caích dùng caic chuyãøn kiãøu vai SIZE khãng coi ãaung [M,N] thç tráí vãø mãüt vectã ãoing. FORMAT laì mãüt chuãùi chãia caic chè ãenh chuyãøn kiãøu cuía ngãn ngãi C. Caic chè ãenh chuyãøn kiãøu bao gãøm kyĩ tãu %, boí læu choün * vai trãeing ããü rãung, vai caic kyĩ tãu chuyãøn kiãøu d, i, o, u, x, e, f, g, s, c, vai [. . .] (tãúp queĩt). Toain bãü ANSI C cung cáúp caic kyĩ tãu chuyãøn kiãøu naìy ããø dùng trong MATLAB. Xem giaõo trçnh C ããø coi ããõy ãuĩ chi tiãút. Nãúu mãüt kyĩ tãu chuyãøn kiãøu ãããüc dùng thç viãüc ãoüc pháõn tãí coi thãø dùng cho nhiãõu pháõn tãí cuía ma tráün MATLAB, mãüi pháõn tãí læu mãüt kyĩ tãu.

Viãüc hoìa trãün caic chè ãenh chuyãøn kyĩ tãu vai sãü laim cho ma tráün kãút quái laì sãü vai mãüi læõn ãoüc caic pháõn tãí biãøu hiãün bàõng caic giaĩ trẽ ASCII cuía chuĩng, mãüi kyĩ tãu ãing vãĩi mãüt pháõn tãí cuía ma tráün. Queĩt cho ããün hãút chuãùi nãúu NEXTINDEX læĩn hãn kêch thããic cuía S. SSCANF khaic vãĩi haím trũng tãn trong C ãĩ chãu quan troũng laì: noĩ "vectã hoìa " ããø tráí vãø mãüt ããüi sãü ma tráün. Chuãùi ãenh ãaung ãããüc cuãün voĩng qua chuãùi cho ããün hãút hoãüc sãü ããĩ læĩu chè ãenh bãĩi SIZE ãããüc chuyãøn hãt.

Vẽ ãu, caic læũnh

```
S = '2.7183 3.1416';
```

```
A = sscanf(S,'%f')
```

ãoüc mãüt vectã gãøm 2 pháõn tãí xáúp xẽ e vai pi

Ããõ hoĩa X-Y cả bãĩn

PLOT

Veĩ caic vectã hoãüc ma tráün

PLOT(X,Y) veĩ vectã Y theo X. nãúu X hoãüc Y laì ma tráün thç vectã ãããüc veĩ theo caic ãoing hoãüc cãüt cuía ma tráün vãĩi hãẽĩng tàng theo chè sãü

PLOT(Y) veĩ caic cãüt cuía Y theo chè sãü. Nãúu Y phãic thç PLOT(Y) tãẽng ããẽng PLOT(real(Y),imag(Y)). Trong tãút cáĩ caic caích dùng khaic cuía PLOT thç pháõn áo ãããüc boĩ qua. Caic kiãøu ããẽng vai mãüi veĩ khaic nhau coi thãø nhãün ãããüc vãĩi læũnh PLOT(X,Y,S) vãĩi S laì 1, 2 hoãüc 3 chuãùi kyĩ tãu taũo nãn tãĩ caic kyĩ tãu sau:

	Y	vãĩng	.	ãĩãøm
	m	tẽm	o	ããü
troĩn	c	xanh thiãnthanh	x	ããü
x	r	ãoĩ	+	
cãũng	g	xanh læu	-	ããüc
	b	xanh ããẽng	*	sao
	w	trãõng	:	
chãúm				

gaûch k âen - . cháúm

--

gaûch
Vê duû, plot (X,Y,'c+') veî dáúu cǎung maìu thiǎn thanh ǎi maùí ãiǎøm ðǎi liǎûu
PLOT(X1,Y1,S1,X2,Y2,S2,X3,Y3,S3,...) tǎø háúp caïc hñh veî xaïc ãenh báíi báü ba (X,Y,S), vǎíi X vaiY lai caïc vectǎ hoàuc ma tráun vai S lai chuǎúi. Vê duû, plot (X,Y,'y-',X,Y,'go') veî hai láön ðǎi liǎûu, vǎíi ãǎng vaìng cuìng vǎíi voìng troìn xanh luúc ǎi maùí ãiǎøm. Lǎûnh PLOT nǎuu khǎng chè ãenh maìu thç ãǎúc tǎu ãǎung ðùng ãàuc tênh ColorOrder cuía truúc. Ngǎøm ãenh ColorOrder ãǎúc liǎút kǎ trong báíng trǎn ãáúi vǎíi háu maìu mai coi ngǎøm ãenh máüt ãǎng lai maìu vaìng vai nhiǎõu ãǎng thç cuǎun voìng qua 6 maìu ãáõu trǎng báíng. Vǎíi háu ãǎn sǎõc thç PLOT cuǎun voìng qua ãàuc tênh LineStyleOrder cuía truúc. PLOT tráí vǎõ máüt vectǎ cǎüt caïc theí cuía caïc ãáúi tǎûng LINE,máüt theí máüt ðoìng. Caïc cǎúp X,Y hoàuc báü ba X,Y,S coi thǎõ theo sau lai caïc cǎúp tham sǎu/giaíi trẽ ãǎõ chè ãenh caïc ãàuc tênh thǎm nǎia cuía caïc ãǎng

LOGLOG

Veî hñh bàòng caìch chia truúc loga

LOGLOG(...) giǎúng PLOT(...), ngoaúi trǎi ðùng caìch chia logarit cho cái hai truúc XY

SEMILOGX

SEMILOGX veî hñh bàòng caìch chia truúc báín loga

SEMILOGX(...) giǎúng PLOT(...), ngoaúi trǎi chia truúc logarit (cǎ sǎu 10) ðùng cho truúc X

SEMILOGY

SEMILOGY veî hñh bàòng caìch chia truúc báín loga

SEMILOGY(...) giǎúng PLOT(...), ngoaúi trǎi chia truúc logarit (cǎ sǎu 10) ðùng cho truúc Y

FILL

Tǎ ãa giaíc 2 chiǎõu

FILL(X,Y,C) tǎ ãa giaíc 2 chiǎõu xaïc ãenh báíi caïc vectǎ X vai Y vǎíi maìu chè ãenh báíi C. Caïc ãenh cuía ãa giaíc chè ãenh báíi caïc cǎúp thǎnh pháõn cuía X vai Y. Nǎuu cáõn, thç ãoìng ãa giaíc bàòng caìch nǎúi ãenh cuǎúi vǎíi ãenh ãáõu. Nǎuu C lai máüt chuǎúi kyí tǎu ãǎn choùn tǎi danh saìch 'r','g','b','c','m','y','w','k', hoàuc máüt báü ba vectǎ ðoìng RGB, [r g b], thç ãa giaíc ãǎúc tǎ vǎíi hàòng maìu chè ãenh. Nǎuu C lai máüt vectǎ cuìng ãáü ðài nhǎ X vai Y thç caïc pháõn tǎi cuía noi ãǎúc chia truúc báíi CAXIS vai ðùng nhǎ chè sǎu trong COLORMAP hiǎun hǎnh ãǎõ chè ãenh caïc maìu ǎi caïc ãenh; maìu bǎn trong ãa giaíc nháun ãǎúc bàòng nǎúi suy song tuyǎún tênh trong caïc maìu ãenh

Nǎuu X vai Y lai caïc ma tráun cuìng cáí thç veî maùí ãa giaíc ǔìng vǎíi máüt cǎüt. Trong trǎng háúp nǎiy,

C là vectơ dòng với các màu đã gia cố "beut", và C là một ma trận với các màu cho đã gia cố "núi suy". Nếu X hoặc Y là ma trận thì thành phần khác là một vectơ cột cùng số dòng thực của vectơ cột để được lặp lại cho ra một ma trận có kích thước mới

FILL(X1,Y1,C1,X2,Y2,C2,...) là một cách khác vẽ chế ảnh từ nhiều vùng

FILL ảnh của các mảng PATCH là FaceColor có ảnh tên 'nón', 'núi suy', hay màu chế ảnh tùy thuộc vào giá trị từ 1 đến 3. FILL vẽ một vectơ cột các thể chế để các ảnh của mảng PATCH, màu thể một mảnh. Ba X, Y, C có thể theo sau là các cặp tham số/giá trị để chế ảnh các ảnh tên để cho các mảnh. FILL không ảnh hưởng cho ảnh tên trục NextPlot

Caic lănh ảnh thể X-Y ảnh biể

POLAR

Vẽ ảnh thể trong hệ tọa độ cực

POLAR(THETA, RHO) vẽ một ảnh thể bằng cách dùng các tọa độ cực của góc THETA, ảnh về radian, bán kính RHO

POLAR(THETA, RHO, S) dùng chế ảnh kiểu ảnh vẽ trong chuỗi S

Xem lệnh PLOT ảnh biể các kiểu ảnh vẽ hình ảnh

BAR

Vẽ ảnh thể thanh ảnh

BAR(Y) vẽ một ảnh thể của các phần tử của vectơ Y

BAR(X,Y) vẽ một ảnh thể của các phần tử của vectơ Y với các vẽ từ chế ảnh trong vectơ X. Các giá trị của X phải theo thứ tự tăng dần. Nếu các giá trị của X không chia đều thực hiện cho các khoảng không đều nhau qua một ảnh để li. Thay vào đó, các thanh ảnh vẽ ở giữa các giá trị X gần nhau. Các ảnh một ảnh chứa nhiều ảnh theo các khoảng trong cho các khoảng ngoài cần thiết

Nếu X và Y là hai ma trận cùng cỡ thì vẽ một ảnh thể ảnh ảnh với một vectơ

[XX,YY] = BAR(X,Y) không vẽ một ảnh thể ảnh ảnh của các vectơ X và Y ảnh PLOT(XX,YY) là lệnh vẽ ảnh thể ảnh

Xem thêm STAIRS, HIST

BAR(X, 'linetype') hoặc BAR(X,Y, 'linetype') dùng kiểu ảnh vẽ chế ảnh là linetype. Xem PLOT ảnh biể thêm chi tiết

STEM

Vẽ ảnh thể ảnh li ảnh rời ảnh

STEM(Y) vẽ ảnh thể ảnh li ảnh Y như ảnh ảnh theo trục x kết thúc với vòng tròn nhỏ với các giá trị

STEM(X,Y) vẽ dãy dữ liệu Y tại các giá trị chèn
ên trong X. Có mặt axis sáu chuôi tùy chọn cuối
cùng là kiểu axis vẽ cho các đây. Như
STEM(X,Y,'-.') hoặc STEM(Y,':')

STAIRS

Biểu đồ bậc thang (biểu đồ không có các mảng
bên trong)

STAIRS(Y) vẽ biểu đồ bậc thang của các phần tử
của vectơ Y

STAIRS(X,Y) vẽ biểu đồ bậc thang của các phần
tử của vectơ Y theo các trục để ảnh hưởng của X. Các
giá trị của X phải theo thời gian tăng dần và cách
nhau

[XX,YY] = STAIRS(X,Y) không vẽ mặt biểu đồ mà
lấy giá trị trái và phải của các vectơ X và Y để
PLOT(XX,YY) là lệnh vẽ biểu đồ

ERRORBAR

ERRORBAR vẽ đồ thị với các thanh sai số

ERRORBAR(X,Y,L,U,SYMBOL) vẽ biểu đồ của vectơ Y
theo X với các thanh sai số để ảnh hưởng của các vectơ
L và U. The vectors X,Y, L và U phải cùng số
lượng. Nếu X,Y, L và U là các ma trận thực hoặc
cộng cho ra một dòng riêng biệt. Các thanh sai số
được vẽ cách U(i) ở trên và L(i) ở dưới mỗi
điểm trong (X,Y) để mỗi thanh có độ dài L(i) +
U(i). SYMBOL là một chuỗi để ảnh hưởng khi hiển thị
ảnh, các ký hiệu và màu cho hình vẽ X-Y

ERRORBAR(X,Y,L) vẽ Y theo X với một thanh sai số
để ảnh hưởng Y có độ dài 2*L(i)

ERRORBAR(Y,L) vẽ Y với thanh sai số [Y-L Y+L]

Về đây,

```
x = 1:10;
```

```
y = sin(x);
```

```
e = std(y)*ones(size(x));
```

```
errorbar(x,y,e)
```

vẽ các thanh sai số để ảnh hưởng theo độ lệch
chuẩn ảnh

HIST

Vẽ biểu đồ

HIST(Y) vẽ một biểu đồ để gồm 10 điểm cách nhau
giữa các giá trị nhỏ nhất và lớn nhất trong Y,
trên bề mặt phân bố của các phần tử trong vectơ
Y

HIST(Y,N), với N là một ảnh hưởng và ảnh hưởng,
đúng N ảnh hưởng

HIST(Y,X), với X là vectơ, vẽ một biểu đồ để đúng
các ảnh hưởng chỉ ảnh hưởng trong X

[N,X] = HIST(...) không vẽ biểu đồ mà trái và
các vectơ X và N ảnh hưởng BAR(X,N) vẽ biểu đồ

ROSE

Vẽ biểu đồ hoa hồng hay biểu đồ góc

ROSE(THETA) vẽ biểu đồ góc cho các góc trong
THETA. Các góc trong vectơ THETA phải ảnh hưởng theo
ảnh hưởng radian

ROSE(THETA,N) với N là ảnh hưởng và ảnh hưởng, đúng
N ảnh hưởng cách ảnh hưởng tại 0 ảnh hưởng 2*PI. Ảnh hưởng ảnh hưởng N = 20

[T,R] = ROSE(...) khăng veỉ mài trái vẫo caỉc vectỏ T
vài R ấẫo POLAR(T,R) veỉ biẫo ấẫo

COMPASS(Z, 'S') và COMPASS(X,Y, 'S') dùng để vẽ đồ thị của hàm số y = f(x) trong miền xác định của x. Khi đó, trục hoành là trục x và trục tung là trục y.

FEATHER(Z, 'S') và FEATHER(X, Y, 'S') dùng để vẽ trục
'S' và 'S' là trục hoành và trục tung trong đồ thị PLOT

[X,Y] = fplot(FNAME,LIMS,...) traí vảo caïc hoanh
 âäü tung âäüü dùng âäø veỉ FNAME trong vectỏ cẩt X
 vại trong caïc cẩt cuỉa ma trẩn Y
 Vê duừ:


```
fplot('sin',[0 4*pi]) ; fplot('sin(x)',[0 4*pi],'-
+') ;
fplot('[sin(x),cos(x)]',[0 4*pi],'-x')
fplot('abs(exp(-j*x*(0:9)))*ones(10,1))',[0 2*pi],'-
o')
fplot('tan',[-2*pi 2*pi -2*pi 2*pi],'-*')
fplot('[tan(x),sin(x),cos(x)]',[-2*pi 2*pi -2*pi
2*pi])
fplot('sin(1 ./ x)', [0.01 0.1],1e-3)
```

COMET

Veê âăö thê vãi buít veê hoaút âăüng nhê sao chăoi trong 2 chiăöu

COMET(Y) hiăön thê măüt hợnh veê comet cuía vectă Y

COMET(X,Y) hiăön thê măüt hợnh veê comet cuía vectă Y theo X

COMET(X,Y,p) duėng comet âăü dài p*length(Y). Ngăöm âềnh p = 0.10

Chênh lăüh COMET lai măüt chăẻng trệnh măùu

Vê duú:

```
t = -pi:pi/200:pi;
```

```
comet(t,tan(sin(t))-sin(tan(t)))
```

Chũi giaíi trăn âăö thê**TITLE**

Ghi tiău âăö cho caíc hợnh veê 2 vại 3 chiăöu

TITLE('text') thăm đơng vản bảín áí âềnh caíc trưê hiăün thăi

TITLE('text','Property1',PropertyValue1,'Property2',PropertyValue2,...) âăüt caíc giaí trê cuía caíc âăüc tênh chề âềnh cuía tiău âăö

XLABEL

Âăinh nhăin trưê X cho caíc hợnh veê 2 vại 3 chiăöu

XLABEL('text') thăm đơng vản bảín bản căüh trưê X cuía caíc trưê hiăün thăi

XLABEL('text','Property1',PropertyValue1,'Property2',PropertyValue2,...) âăüt caíc giaí trê cuía caíc âăüc tênh chề âềnh cuía nhăin trưê x

YLABEL

Âăinh nhăin trưê Y cho caíc hợnh veê 2 vại 3 chiăöu

YLABEL('text') thăm đơng vản bảín bản căüh trưê X cuía caíc trưê hiăün thăi

YLABEL('text','Property1',PropertyValue1,'Property2',PropertyValue2,...) âăüt caíc giaí trê cuía caíc âăüc tênh chề âềnh cuía nhăin trưê y

TEXT

Thă vản bảín văo hợnh veê hiăün thăi

TEXT(X,Y,'string') thăm vản bảín trong căúp đăúu nhăy văo vê trê (X,Y) trăn caíc trưê hiăün thăi, vãi (X,Y) lai caíc âăn vê cuía hợnh veê hiăün thăi.

Nếu X và Y là các vectơ, TEXT ghi các và bán kính tất cả các về trục hoành. Nếu 'string' là một mảng cùng số dòng với axis dài của X và Y, TEXT ảnh dấu màu ảnh với các dòng tăng ứng của mảng 'string'

TEXT(X,Y,Z,'string') thêm và bán kính trong hồ của axis 3 chiều

TEXT trả về một vectơ cột gồm các thể chế axis các axis tăng TEXT, màu thể một axis tăng và bán kính. Các axis tăng TEXT là con của các axis tăng AXES. Cặp X,Y (bộ ba X,Y,Z cho 3 chiều) coi theo sau bởi các cặp tham số/giải trục chế trục các trục tên thêm của và bán kính. Cặp X,Y (bộ ba X,Y,Z cho 3 chiều) coi theo bộ qua toàn bộ, và tất cả các chế trục axis tên bằng cách dùng các cặp tham số/giải trục. Thúc hiện GET(H), với H là thể và bán kính axis xem danh sách các axis tăng và bán kính giải trục hồ của chúng

GTEXT

Axis và bán kính vào hình vẽ 2 chiều bằng cách dùng chuốt

GTEXT('string') hiện thể của số axis thể, axis ra con troi chuốt, và chữ số nút chuốt hoặc số thêm. Con troi chuốt coi theo thay axis vẽ trục bằng chuốt (hoặc bằng các thêm mũi tên trên vài loaị máy). Khi số nút chuốt hoặc số thêm nút ký tự và bán kính axis vào axis thể trục vẽ trục axis cho

GRID

Vẽ các trục lưới cho các hình vẽ 2 và 3 chiều

GRID ON thêm các trục lưới vào các trục hiện tại

GRID OFF tắt các trục lưới

GRID, chế lưới này lưới trục thể hiện lưới

GRID axis các trục tên XGrid, YGrid, và ZGrid của các trục hiện tại

Caic lănh vẽ ânh vai tă vùng

PLOT3

Vẽ các trục vài ảnh trong không gian 3 chiều

PLOT3() tăng trục 3 chiều của PLOT()

PLOT3(x,y,z), với x, y và z là 3 vectơ cùng axis dài, vẽ một trục trong không gian 3 chiều qua các ảnh coi hồ của axis là các phần tử của x, y và z

PLOT3(X,Y,Z), với X, Y và Z là 3 ma trận cùng cỡ, vẽ nhiều trục như trục trục tại các cột tăng ứng của X, Y và Z.

Các kiểu trục khác nhau, dấu hiện vẽ vài màu coi theo như trục với PLOT3(X,Y,Z,s), s là một chuốt gồm 1, 2 hoặc 3 ký tự trục trục tại các ký tự trục như trong lưới PLOT

PLOT3 traí vảo maut vecta cút caic thei ché aân caic aân tæung LINE, maut thei maut dòng. Bâu ba X,Y,Z, hoac bâu bân X,Y,Z,S coi thã tiáp theo lai caic cút tham sáu/gia trê aân ché aên caic aân tên thãm cuá caic aân

FILL3

Veî vai tă măüt âa giaïc 3 chiăöu trong khăng gian 3
chiăöu

[illegible]

Năúu măút trờng X, Y hoàúc Z lài ma tráúñ, vại caíc tham săú khaíc lài caíc vectă căút cùng săú đoiğ, thç caíc âăúi săú vectă căút âăăúc tăi taúo âăø cho ra caíc ma tráúñ coi kêch thăăic hăúp lăú

FILL3 (X1,Y1,Z1,C1,X2,Y2,Z2,C2,...) lài mǎut caïch
khaïc âãø chè âênh tă nhiãöu vùng

FILL3 ààùt àäúi tǎǎung PATCH ààùc tênh FaceColor lài 'flat', 'interp', hoàùc mǎüt chè àěnh màìu phuû thuǎüc vaìo giaỉ trẽ cuía ma tráùn C. FILL3 traí vǎo mǎüt vectǎ cǎüt gǎòm caíc theí chè àǎøn caíc àäúi tǎǎung PATCH, mǎùi theí mǎüt màính. Bǎu bǎùn X,Y,Z,C coỉ thǎø tiǎúp theo lài caíc cǎúp tham sǎù/giaỉ trẽààø chè àěnh caíc ààùc tênh thǎm nǎià cuía caíc màính. FILL3 khǎng tǎng ǎěng vǎi ààùc tênh truúc cuía NextPlot

COMET3

Veî âăö the vãi buít veî hoaút âăüng nhæ sao chăøi trong 3 chiăöu

COMET3(Z) hiăøn the măt hñnh veî comet 3 chiăöu của vectă Z

COMET3(X,Y,Z) hiăøn the măt hñnh veî comet 3 chiăöu của âăüng cong qua các âiăøm [X(i),Y(i),Z(i)]

COMET3(X,Y,Z,p) dùng comet âăü dài p*length(Z). Ngăøm âênh p = 0.1

COMET3, chênh lăünh này lại chăêng trñnh màu Vê duú:

```
t = -pi:pi/500:pi;
comet3(sin(5*t),cos(3*t),t)
```

Veî âăüng măic vai các hñnh veî khaíc 2 chiăöu của đăi liăüu 3 chiăöu

CONTOUR

CONTOUR veî âăüng măic

CONTOUR(Z) lại măt hñnh veî âăüng măic của ma trăün Z xem các giáỉ trê của Z lại chiăöu cao trăn măt măt phăóng

CONTOUR(X,Y,Z), vãi X vài Y lại các vectă, chê âênh X vài Y lại các truúc dùng trăn hñnh veî

CONTOUR(Z,N) vài CONTOUR(X,Y,Z,N) veî N âăüng măic, âêi lăn các giáỉ trê ngăøm âênh măt cách tău âăüng.

CONTOUR(Z,V) vài CONTOUR(X,Y,Z,V) veî các âăüng măic âăü dài LENGTH(V) tăui các giáỉ trê chê âênh trong vectă V

CONTOUR(...,'linetype') veî vãi màu vài kiăøu âăüng chê âênh, nhæ lăünh PLOT

C=CONTOUR(...) trăi văö ma trăün C nhæ trñnh báy trong lăünh CONTOURC vài dùng bảí CLABEL

[C,H] = CONTOUR(...) trăi văö vectă căt H găøm các theí chê âăün các âăúi tăüng LINE, măt theí măt âăüng

CONTOUR3

CONTOUR3 veî âăüng măic 3chiăöu

CONTOUR3(Z) veî các âăüng măic của Z trong 3 chiăöu

CONTOUR3(Z,N) veî N âăüng măic trong 3 chiăöu. Ngăøm âênh N=10

CONTOUR3(X,Y,Z) hoăüc CONTOUR3(X,Y,Z,N) dùng các ma trăün X vài Y âă í xăic âênh giáỉ haün của các truúc

C = CONTOUR3(...) trăi văö ma trăün âăüng măic C nhæ trong lăünh CONTOURC

[C,H] = CONTOUR3(...) trăi văö vectă căt H găøm các theí chê âăün các âăúi tăüng LINE, măt theí măt âăüng

CLABEL

Thăm các nhain âằing mẵc vaio hẻnh veỉ âằing mẵc

CLABEL(CS) thăm nhain âầu cao vaio hẻnh veỉ âằing mẵc hiầun thằi bằong caỉch đùng cáúu truỉc CS xuấut tằi phuỏc vuỏ CONTOUR. Các vẻ trẻ âằnh nhain âằủc choủn mắut caỉch ngầu nhiầ.

Vẻ đườ: cs = contour(rand(10)); clabel(cs)

CLABEL(CS,V) âằnh các nhain âủing âằing mẵc àoỉ vằi các bắuc cho trong vectỏ V. Ngắom âẻnh thẻ hoắut âằung nằy âằnh nhain cho táút cáỉ các âằing mẵc âằ biắút

CLABEL(CS,'manual') âầut các nhain âằing mẵc taủi các vẻ trẻ áún chuắut. Áún <ENTER> âầ kắut thuỉc viắuc âằnh nhain. Đùng thanh trắung (phẻm space bar) âầ vaio các âằing mẵc vằi các phẻm muỉ tắn âầ di chuyắon con troỉ chuắut nằu khắng đùng âằủc chuắut

CONTOURC

CONTOURC Tẻnh âằing mẵc

CONTOURC tẻnh ma trắun âằing mẵc C âầ đùng CONTOUR veỉ âằing mẵc

C = CONTOURC(Z) tẻnh ma trắun âằing mẵc cho mắut hẻnh veỉ âằing mẵc cuỉa ma trắun Z bằong caỉch xem các gắỉ trẻ trong Z lằi âầ cao trắn hẻnh veỉ

C = CONTOURC(X,Y,Z), vằi X vằi Y lằi các vectỏ, chẻ âẻnh X vằi Y lằi các truỉc âầ ỏ tẻnh âằing mẵc

CONTOURC(Z,N) vằi CONTOURC(X,Y,Z,N) tẻnh N âằing mẵc, âẻi lắn các gắỉ trẻ ngắom âẻnh mắut caỉch tằủ âằung

CONTOURC(Z,V) vằi CONTOURC(X,Y,Z,V) tẻnh LENGTH(V) âằing mẵc taủi các gắỉ trẻ chẻ âẻnh trong vectỏ V

Ma trắun âằing mẵc C lằi ma trắun 2 đỏing cuỉa các âằing mẵc. Mằui àoắun veỉ liắn tiắúp chằỉa gắỉ trẻ cuỉa âằing mẵc, sằu cằup veỉ (x,y), vằi veỉ chẻnh các cằup

Các àoắun liắn tiắúp nằu nhằ

```
C = [level1 x1 x2 x3 ... level2 x2 x2 x3 ...;
      pairs1 y1 y2 y3 ... pairs2 y2 y2 y3 ...]
```

PCOLOR

Veỉ gắỉ mằu (bắing kiắom tra)

PCOLOR(C) lằi mắut hẻnh veỉ gắỉ mằu hay lằi " bắing kiắom tra " cuỉa ma trắun C. Các gắỉ trẻ cuỉa các phắon tắỉ cuỉa C chẻ âẻnh mằu trắn mằui ắ cuỉa hẻnh veỉ. Trong chắủ âầ âầut boẻng ngắom âẻnh, 'faceted', mằui ắ coỉ mắut hằong mằu vằi đỏing vằi cắut cuắui cuỉa C khắng âằủc đùng. Vằi boẻng nằui suy, 'interp', mằui ắ coỉ mằu lằi kắut quắ tằi pheỉp nằui suy song tuyắún tẻnh cuỉa mằu áỉ bắún âẻnh vằi táút cáỉ các phắon tắỉ cuỉa C âầủ âằủc đùng. Các phắon tắỉ lắnn nhắut vằi nỏỏ nhắut cuỉa C âằủc gắnn cho các mằu áầủ vằi cuắui cho trong bắing mằu; các mằu cho các phắon tắỉ coỉn lằui trong C âằủc xắc âẻnh bắỉ bắing - tẻm bắn trong phắon coỉn lằui cuỉa bắing mằu

PCOLOR(X,Y,C), vãi X và Y là các vectơ và ma trận, tạo ra mặt hình vẽ giải màu trên lưới xác định bởi X và Y. X và Y có thể xác định lưới cho mặt "áa" chằng hân.

PCOLOR thực sự là SURF vãi hình ảnh của nó ảnh hưởng tới lần trên

PCOLOR trái với mặt thể chế ảnh hưởng tới các SURFACE

QUIVER

QUIVER vẽ ảnh rung (hay trục ảnh)

QUIVER(X,Y,U,V) vẽ các vectơ trục ảnh vãi các thành phần (u,v) tại các điểm (x,y). Các ma trận X,Y,U,V phải cùng cỡ và chứa các về trên cùng vãi các thành phần trục ảnh (X và Y cũng có thể là các vectơ chế ảnh mặt lưới ảnh). QUIVER tạo ảnh phản ánh các vectơ trục ảnh cho phù hợp bản trong lưới

QUIVER(U,V) vẽ các vectơ trục ảnh tại các điểm cách ảnh trong mặt phẳng x-y

QUIVER(X,Y,S) hay QUIVER(X,Y,U,V,S,...) tạo ảnh phản ánh các vectơ trục ảnh cho phù hợp bản trong lưới rồi nhân chằng vãi S. Dùng S=0 ảnh vẽ các vectơ trục ảnh mà không tạo ảnh phản ánh

QUIVER(...,STYLE) dùng hình vẽ vãi kiểu vẽ ảnh chế ảnh bởi chuỗi STYLE cho các vectơ trục ảnh. Áa vào mặt dấu '>' trong STYLE ảnh vẽ các ảnh mới lần trong các vectơ trục ảnh

Xem PLOT vãi các kiểu vẽ ảnh

H = QUIVER(...) trái với vectơ gồm các thể ảnh

Vê du:

```
[x,y] = meshgrid(-2:.2:2,-1:.15:1);
z = x .* exp(-x.^2 - y.^2); [px,py] =
gradient(z,.2,.15);
contour(x,y,z), hold on
quiver(x,y,px/5,py/5), hold off, axis image
```

Caic lănh vẽ bđ mặt vài lăi

MESH

Mặt lưới 3 chiều

MESH(X,Y,Z,C) vẽ lưới tham số màu xác định bởi 4 ảnh số ma trận. Ảnh xem ảnh chế ảnh bởi VIEW. Các nhân trục ảnh xác định bởi miền giải trẻ của X, Y và Z, hoặc bởi các ảnh hướng trục của AXIS. Vãi chia màu ảnh xác định bởi miền giải trẻ của C, hoặc bởi các ảnh hướng trục của CAXIS. Các giải trẻ màu ảnh chia ảnh dùng nh các chế số cho COLORMAP hướng trục

MESH(X,Y,Z) dùng C = Z, vẽ vãi màu ảnh cán xằng vãi ảnh cao của lưới

MESH(x,y,Z) và MESH(x,y,Z,C), vãi 2 ảnh số vectơ thay 2 ảnh số ma trận ảnh tiên, phải có length(x) = n và length(y) = vãi [m,n] = size(Z). Trong trường hợp này, các ảnh của của các ảnh lưới là các bộ ba (x(j), y(i), Z(i,j)). Lưu ý

SURF(X,Y,Z,C) vẽ mặt có tham số màu xác định bởi 4 trục số ma trận. Axis xem các trục vẽ bởi VIEW. Các nhân trục các trục xác định bởi miền giá trị của X, Y và Z, hoặc bởi cái axis hiển thị của AXIS. Vẽ chia màu các trục xác định bởi miền giá trị của C, hoặc bởi cái axis hiển thị của CAXIS. Các giá trị màu sẽ chia các trục dùng như các chế độ số cho COLORMAP hiển thị. Mã lệnh bằng các trục axis SHADING

SURF(X,Y,Z) dùng C = Z, vẽ váûy màu âãüç cán xẽing
vãii âãü cao củia lããii

SURF(x,y,Z) và SURF(x,y,Z,C), với 2 âäúi säú vectå thay 2 âäúi säú ma tråûn åäöu tiån, pháúi coi length(x) = n và length(y) = với [m,n] = size(Z). Trong tråëng hãüp này, cáic åênh cuía cuía cáic máinh bãö màût là cáic bãü ba (x(j), y(i), Z(i,j)). Låu yĩ ràöng x tåëng åïng với cáic cåüt cuía Z và y tåëng åïng với cáic ðöng

SURF(Z) và SURF(Z,C) dùng $x = 1:n$ và $y = 1:m$.
 Trong trường hợp này, chiều cao, Z, là một hàm
 giá trị âm, xác định trên một lưới hình chữ
 nhật.

SURF traí vǎö mäüt theí chè âǎún mäüt âǎúi tǎǎüng
SURFACE

AXIS, CAXIS, COLORMAP, HOLD, SHADING và VIEW ảnh hưởng đến hình ảnh, các trục, và các thuộc tính bề mặt đồ họa ảnh hưởng đến hiển thị bề mặt

SURFC

Tạo hẩu lẫnh vẽ SURF/CONTOUR

SURFC(...) giǎúng SURF(...) ngoaũ trặi mǎüt âşǎing
mǎic âşǎüc veĩ bẵ dǎĩi bǎõ mǎüt

Vç CONTOUR khäng xại lyï dãi liăũu kyì dẻ, năn phuôc
vuũ nạy chẻ laim viăũc âăúi vãi căic băo măt xăic
ăẻnh trăn lăăii chăi nhăt. Căic ma trătun hoăuc vectă
X vại Y chẻ xăic ăẻnh căic giăăii haun truôc

SUREFL

Veî màût 3 chiãöu coi âäü saing

SURFL(...) giáng nhæ SURF(...) ngoài trãi về bề mặt với các đầu sáng tại một nguồn sáng. SURFL(Z), SURFL(X,Y,Z), SURFL(Z,S), và SURFL(X,Y,Z,S) rút ra đầu hộp lá. Nếu coi chế ảnh S, thç 3 vectơ $S = [S_x, S_y, S_z]$ chế ảnh hằng nguồn sáng. S cũng coi thç chế ảnh trong hộp toà đầu mai, $S = [A_Z, E_L]$. Bổng hçnh đưa vào tạo hộp các kiểu đầu sáng khuếch tán, phân chiếu và bao quanh. Giá trị ngòm ảnh cho S là 450 cùng chiếu kim ảnh hç tại hằng xem hiên thãi. Dùng CLA, HOLD ON, VIEW(AZ,EL), SURFL(...), HOLD OFF ảnh về mặt coi đầu sáng với hằng quan sát (AZ,EL). Phân bầu tăng ảnh ảnh coi đầu sáng chung quanh(ka), khuếch tán(kd) , phân chiếu(ks) và ảnh trong suất (spread) coi thç ảnh ảnh bải 5 ảnh số SURFL(X,Y,Z,S,K) với $K=[k_a, k_d, k_s, spread]$

Dễ dàng vào thời tậ các aiếm trong các ma trận X, Y , và Z aiếm aênh bản trong vài bản ngoài của các bề mặt tham sấu. Thới lầnh $SURFL(X', Y', Z')$ nầu baùn khâng thêch kẩt quai của hàm này. Do cách tên các vectơ chuẩn của bề mặt, nầ $SURFL$ àoii hoii các ma trận êt nhẩt lài báuc 3

VIEWMTX

VIEWMTX Phaït sinh caïc ma tráùn biấún âäøi quan saït
 A=VIEWMTX(AZ,EL) traí vảo ma tráùn biấún âäøi trảo
 giao báúc 4, A, dùng âãø chiấúu caïc vectả 3 chiấồ
 lần bảồ màút 2 chiấồ. Dùng cùng goïc phảong về vai
 goïc náng nhæ VIEW, âầúc biấút, AZ vai EL phaỉ theo
 âản về âầ. Traí vảo cùng ma tráùn biấún âäøi nhæ caïc
 lầủnh

_VIEW(AZ,EL)

_A = VIEW

nhạng khẳng thay âäøi VIEW hiầủn thẳi

A=VIEWMTX(AZ,EL,PHI) traí vảo ma tráùn biấún âäøi
 phẩúi caỉnh báúc 4 dùng âãø chiấúu caïc vectả
 3 chiấồ lần bảồ màút 2 chiấồ. PHI laỉ goïc trảo
 cuía khẩúi chuáỏn (âản về âầ) vai âiầồ khiầỏn sầ
 âầ lầủch phẩúi caỉnh:

_PHI = 00 laỉ pheỉp chiấúu trảo giao

_PHI = 100 giẩúng camera

_PHI = 250 nhæ thẩú kềnh thẳằng

_PHI = 600 nhæ thẩú kềnh goïc rẳủng

Ma tráùn A coỉ thẳø dùng âãø dầút pheỉp biầỏn âäøi
 quan saït bầỏng lầủnh VIEW(A)

A=VIEWMTX(AZ,EL,PHI,XC) traí vảo ma tráùn biấún âäøi
 phẩúi caỉnh bầỏng caỉch dùng XC laỉ âiầỏm âềch bắ
 trong khẩúi. XC=[xc,yc,zc] chề âềnh âiầỏm (xc,yc,zc)
 trong khẩúi chuáỏn. Giaỉ trề ngắỏm âềnh laỉ âiầỏm gắỏn
 khẩúi chuáỏn nhẩút,

$$XC = 0.5 + \sqrt{3} / 2 * [\cos(EL) * \sin(AZ), -\cos(EL) * \cos(AZ), \sin(EL)]$$

HIDDEN

Ầầút chẩú âầ xỏi âềằng áỏn cuía lầẩỉ

HIDDEN ON báút chẩú âầ xỏi âềằng áỏn cuía lầẩỉ
 hiầủn hằnh

HIDDEN OFF tầỏt chẩú âầ xỏi âềằng áỏn vặ vầủ coỉ
 thẳø thẩúy qua lầẩỉ hiầủn hằnh
 Chềnh lầủnh HIDDEN lầút trầủng thẳỉ xỏi âềằng áỏn

SHADING

SHADING âầút chẩú âầ boỉng hẻnh

SHADING âiầồ khiầỏn maỉu boỉng cuía caïc âầúi tẳằng
 SURFACE vai PATCH. Caïc âầúi tẳằng SURFACE vai PATCH
 âầúc taồ ra bắỉ caïc haỉm SURF, MESH, PCOLOR, FILL,
 vai FILL3.

SHADING FLAT âầút boỉng cuía âầồ hoỉa hiầủn hằnh
 vaồ nầỏn

SHADING INTERP âầút boỉng vaồ pheỉp nầủ suy

SHADING FACETED âầút boỉng vaồ âa dỉầủn vai laỉ
 ngắỏm âềnh. Boỉng nầỏn laỉ hầỏng theo tằng maỉnh;
 mầủ âoảủn cuía âềằng lầẩỉ hay maỉnh bảồ màút coỉ
 mầút maỉu hầỏng xỏi âềnh bắỉ giaỉ trề maỉu áỉ mầủ
 âầồ muỉt cuía mầủ âoảủn hoầúc goïc cuía mầủ maỉnh
 coỉ chề sầủ nhoỉ nhẩút trong caïc chề sầủ. Boỉng nầủ
 suy, coỉn goỉi laỉ boỉng, laỉ song tuyẩủn tềnh theo
 tằng maỉnh; maỉu trong mầủ âoảủn hoầúc caïc maỉnh

Caïc hàm `AXIS` , `CAXIS` , `COLORMAP` nhæ trong `plotxy`

vê duû, [0 0 0] lài hoàn toàn âen, [1 1 1] lài
trắng, [1 0 0] lài ôóí, [.5 .5 .5] lài xám, và
[127/255 1 212/255] lài ngoc đẳng. Các âáúi táúng

ããø hoüa ðùng giaí maüu -- laì caïc ääúi tæüng SURFACE vaì PATCH, äæüç taüo ra báíi caïc haím MESH, SURF, vaì PCOLOR -- aính xaü müüt ma tráün maüu, C, coì caïc giaí trë trong miãön [Cmin, Cmax], vaìo müüt máng caïc chè sáu k, trong miãön [1, m]. Caïc giaí trë Cmin vaì Cmax laì min(min(C)) vaì max(max(C)), hoüc laì chè äënh báíi CAXIS. Aính xaü laì tuyáün tênh, vãi Cmin aính xaü sang chè sáu 1 vaì Cmax aính xaü sang chè sáu m. Sau aoì caïc chè sáu äæüç ðùng vãi báng maüu äãø xaïc äënh maüu tæng æng vãi müüi pháön táí cuía ma tráün. Xem CAXIS äãø biáút thãm chỉ tiáút. Vaìo lãüh HELP COLOR äãø xem müüt sáu caïc báng maüu thëng ðùng. COLORMAP laì müüt M-file äãø ääút thuäc tênh báng maüu cuía hënh aính hiãün thãi

CAXIS

Chia truüc giaí maüu

CAXIS(V), vãi V laì vectå 2 pháön táí [cmin cmax], ääút caích chia truüc thuí cëng giaí maüu cho caïc ääúi tæüng SURFACE vaì PATCH taüo ra báíi caïc lãüh nhæ MESH, PCOLOR, vaì SURF. cmin vaì cmax äæüç gäin vaìo caïc maüu äãøu vaì cuúi cuía báng maüu hiãün thãi. Caïc maüu cho PCOLOR vaì SURF äæüç xaïc äënh báíi báng tra cæu trong miãön naìy. Caïc giaí trë ngoài miãön äæüç xeín bàòng caích taüo ra chuëng müüt caích roì ràng. CAXIS('auto') ääút caích chia truüc ngæüç vãø miãön giaí trë tæü ääüng. Chënh haím CAXIS tráí vãø vectå ðoìng gäöm 2 pháön táí [cmin cmax] äang coì hiãüu læüc. CAXIS laì müüt M-file äãø ääút caïc ääüt tênh truüc CLim vaì CLimMode

SHADING

SHADING Chäü ääü maüu cuía boìng

SHADING äiãøu khiãøn maüu boìng cuía caïc ääúi tæüng SURFACE vaì PATCH. Caïc ääúi tæüng SURFACE vaì PATCH äæüç taüo ra báíi caïc haím SURF, MESH, PCOLOR, FILL, vaì FILL3

SHADING FLAT ääút boìng cuía hënh veì hiãün thãi vaìo nãön

SHADING INTERP ääút boìng vaìo nãüi suy

SHADING FACETED ääút boìng nhiãøu màüt, vaì laì ngäöm äënh

Boìng nãön laì hàòng tæng mánh; müüi äoaün thàóng cuía læüi hoüc mánh màüt coì müüt hàòng maüu äæüç xaïc äënh báíi giaí trë maüu taüi äãøu müüt cuía äoaün hoüc goïc cuía mánh coì chè sáu nhoí nháút. Boìng nãüi suy còin goüi laì boìng Gouraud, laì song tuyáün tênh tæng mánh; maüu trăn müüi äoaün hoüc müüi mánh khäic vãi caïc giaí trë áí äãøu müüt hoüc goïc theo caích tênh tuyáün tênh hoüc nãüi suy. Boìng nhiãøu màüt laì boìng nãön vãi caïc äæüç læüi äen äèi læn. Äáy thæüng laì caích hiãüu læüc nháút vaì laì màüc äënh. SHADING laì müüt M-file äãø ääút caïc ääüt tênh EdgeColor vaì FaceColor cuía táút cá caïc ääúi tæüng SURFACE trong caïc truüc hiãün thãi. Nãi ääút chuëng

vaio caic giaï trë âuïng àaõn phuû thuäüc vaio caic
 ääúi tæäüng SURFACE lai læäïi hay maut

Caic baíng màu

HSV

Baíng màu HSV (Saíng-Äáûm-Giaï trë)

HSV(M) traí vảo ma tráûn câi Mx3 chæïa maut baíng màu HSV

HSV, cùng ääü dài vảo baíng màu hiäûn thãi. Maut baíng màu HSV khaiç thành pháõn màu sảo vảo mã hçnh HSV. Caic màu bắt äáüu vảo màu áoí, ääún vaìng, xanh luüc, thiãn thanh, xanh dæång, tøm, vai tráíí vảo áoí. Baíng màu hæü êch ääüc biäút cho viäüc hiäõn thë caic chæic nàng coi chu kyì

Vê duû, ääø ääüt laûi baíng màu cuía hçnh ánh hiäûn thãi:
 colormap(hsv)

GRAY

Baíng màu chia truüc xaim tuyäún tên

GRAY(M) traí vảo maut ma tráûn câi Mx3 chæïa maut baíng màu chia truüc xaim tuyäún tên

GRAY, cùng ääü dài vảo baíng màu hiäûn thãi

Vê duû, ääø ääüt laûi baíng màu cuía hçnh ánh hiäûn thãi:
 colormap(gray)

HOT

Baíng màu Äen-Äáoí-Vaìng-Träõng (Black-red-yellow-white)

HOT(M) traí vảo maut ma tráûn câi Mx3 chæïa maut baíng màu "hot"

HOT, cùng ääü dài vảo baíng màu hiäûn thãi

Vê duû, ääø ääüt laûi baíng màu cuía hçnh ánh hiäûn thãi:
 colormap(hot)

COOL

Taûo dæĩng cuía baíng màu thiãn thanh vai tøm

COOL(M) traí vảo maut ma tráûn câi Mx3 chæïa maut baíng màu "cool"

COOL, cùng ääü dài vảo baíng màu hiäûn thãi

Vê duû, ääø ääüt laûi baíng màu cuía hçnh ánh hiäûn thãi:
 colormap(cool)

BONE

Chia truüc xaim vảo maut baíng màu pha maut êt xanh dæång

BONE(M) traí vảo maut ma tráûn câi Mx3 chæïa maut baíng màu "bone"

BONE, cùng ääü dài vảo baíng màu hiäûn thãi

Vê duû, ääø ääüt laûi baíng màu cuía hçnh ánh hiäûn thãi:
 colormap(bone)

COPPER

Baíng màu màu ôông tuyáun tên
 COPPER(M) trái vảo măt ma tráu cấ Mx3 chĩa măt
 baíng màu "copper"
 COPPER, cùng ôău dài vắ baíng màu hiăun thắ
 Về đư, ôă ôăt láu baíng màu của hợnh ánh hiăun
 thắ: colormap(copper)

PINK

Các bợng tợng lam nhăt của baíng màu hợng
 PINK(M) trái vảo măt ma tráu cấ Mx3 chĩa măt baíng
 màu "pink"
 PINK, cùng ôău dài vắ baíng màu hiăun thắ
 Về đư, ôă ôăt láu baíng màu của hợnh ánh hiăun
 thắ: colormap(pink)

PRISM

Baíng màu coỉ các màu lợng kợnh
 PRISM(M) trái vảo măt ma tráu cấ Mx3 chĩa cách
 đợng lợp 6 màu: ôó, da cam, vợng, xanh lợc, xanh
 đợng, tợ violet. Giắ trợ mợc ôợnh của M lá ôău
 dài của baíng màu hiăun thắ
 PRISM, kợ coỉ tham sắ nhắ hợ hợ xúút, thay ôăi
 các màu của các ôăi tợ đợ bắ kợ trong các
 trợ hiăun thắ sang màu lợng kợ. Cách này ôăc
 biăt hợ ôợ ôăi vắ măt hợnh vẽ CONTOUR
 contour(data,20)
 prism

Các màu trong baíng PRISM cùng ôăc ôă ra giợng
 nhắ baíng HSV. Tuy nhắ, PRISM đợng lợp các baíng
 sao của 6 màu, ngợc láu HSV thay ôăi các màu
 của nợ măt cách măt mắ

JET

Biăt thắ của HSV
 JET(M), măt biăt thắ của HSV(M), lá baíng màu
 đợng vắ hợnh ánh ôen tuyăon NCSA
 JET, cùng ôău dài vắ baíng màu hiăun thắ.
 Đợng COLORMAP(JET).

FLAG

Baíng màu trợ ôó, trợ, xanh đợng
 FLAG(M) trái vảo măt ma tráu cấ Mx3 chĩa măt baíng
 màu "flag". Viăt tợ M làm tợ sắ nợi hắ của
 baíng FLAG, cùng ôău dài vắ baíng màu hiăun thắ
 Về đư, ôă ôăt láu baíng màu của hợnh ánh hiăun
 thắ: colormap(flag)

Các hàm baíng màu liă quan**COLORBAR**

COLORBAR Hiăon thợ thanh mắ (thắ màu)
 COLORBAR('vert') thắ măt thắ màu ôợng vào các
 trợ hiăun thắ

COLORBAR('horiz') thãm măt thắic mầu ngang
 COLORBAR(H) àút thanh mầu vào hấ toĩa àũ H. Thanh
 mầu sẽ nòm ngang nấu hấ H coi chiấo rấng lấi
 hần chiấo cao (trong hấ toĩa àũ chuáon)
 COLORBAR khắg tham sấu thặ hoắc lầi thãm măt thắic
 mầu àểg, hoắc lầi cúp nhắt măt thanh mầu ài
 coi
 H = COLORBAR(...) trái vấ măt thei chề àũn cáic
 trườ thanh mầu

HSV2RGB

HSV2RGB Chuyấon HSV sang RGB
 M = HSV2RGB(H) chuyấon măt báing mầu HSV sang măt
 báing mầu RGB. Mắi báing mầu lầi măt ma tráun coi
 sấu ðoing báut kyi vai coi àũg 3 cắu, vai cáic
 phắon tái trong àoaun [0, 1]. Cáic cắu trong ma tráun
 nhắp, H, biấu hiấu H-S-V tắg ểg. Cáic cắu cuía
 ma tráun kắu quái, M, biấu hiấu cắg àũ cuía R-G-
 B tắg ểg
 Nhæ H(:,1), àũ saing, biấu thiắn tặ 0 àũn 1, thặ
 mầu kắu quái biấu thiắn tặ àoi sang vaing, xanh
 lườ, thiắ thanh, xanh ðắg vai tề, àen sang àoi.
 Khi H(:,2), àũ àũm, lầi 0, thặ cáic mầu khắg àũm;
 àn giắi chũg coi ðoing xắm. Khi H(:,2) lầi 1, thặ
 cáic mầu cắc àũm; chũg khắg chắa thắnh phắon
 trắg. Khi H(:,3), giắ trề mầu, biấu thiắn tặ 0
 àũn 1, thặ tắg àũ saing
 Báing mầu ngắm àềh, HSV, lầi hsv2rgb([h s v]) vắi
 h lầi măt biấu thiắ tuyấu tềh tặ 0 àũn 1 vai
 cá hai s vai v àũ lầi 1

RGB2HSV

RGB2HSV Chuyấon àũi RGB sang HSV
 H = RGB2HSV(M) chuyấon măt báing mầu RGB sang măt
 báing mầu HSV. Mắi báing lầi măt ma tráun coi
 sấu ðoing báut kyi vai coi àũg 3 cắu, vai cáic
 phắon tái trong àoaun [0, 1]. Cáic cắu cuía ma tráun
 nhắp, M, biấu hiấu cắg àũ cuía R-G-B tắg ểg. Cáic
 cắu cuía ma tráun xuắu ,H , biấu hiấu H-S-V tắg
 ểg

CONTRAST

CONTRAST Chia trườ xắm báing mầu àũ tắg àũ tắg
 phắu cuía hặh áing
 CMAP = CONTRAST(X,M) trái vấ măt báing mầu chia
 trườ xắm, lầi măt ma tráun cái Mx3 vắi 3 cắu àũg
 nhắu, àũ IMAGE(X)
 COLORMAP(CMAP) coi măt biấu àũ chề tề lầ phắn bá
 thầ àũg àũ. Nấu ði qua M thặ ðuing àũ ðài mắc
 àềh lầ 64

BRIGHTEN

BRIGHTEN Baíng màu sáng hơn tối
 BRIGHTEN(beta) thay baíng màu hiện thời mới baíng
 màu xám hơn tối giảm cả baín các màu giảm
 nhau. Baíng màu sáng hơn nếu $0 < \text{beta} \leq 1$ và tối
 hơn

$-1 \leq \text{beta} < 0$

BRIGHTEN(beta), sau đó BRIGHTEN(-beta) phục hồi
 baíng màu gốc

map = BRIGHTEN(beta) trái với mới phần baín sáng
 hơn tối của baíng màu đang dùng mà không thay
 đổi cách hiển thị

newmap = BRIGHTEN(map,beta) trái với mới phần baín
 sáng hơn tối của baíng màu chế ảnh mà không
 thay đổi cách hiển thị

SPINMAP

SPINMAP Xoay baíng màu

SPINMAP quay tròn baíng màu khoảng 3 giây

SPINMAP(T) quay tròn baíng màu khoảng T giây

SPINMAP(inf) là một vòng lặp vô hạn, ngắt với
 phím <^C>

SPINMAP(T,inc) dùng để tăng chế ảnh. Ngắt ảnh thực
 inc = 2, với vậy inc = 1 là một cách quay chậm hơn
 , inc = 3 là nhanh hơn, inc = -2 là theo hướng
 khác, ...

Để tránh vẽ lại liên tục, có
 set(gcf,'sharecolors','no')

RGBPLOT

RGBPLOT Vẽ baíng màu

RGBPLOT(MAP) vẽ một baíng màu, nhúng lại một ma
 trận các mx3 là ảnh sau nhập vào để cho
 COLORMAP. Ba cột của ma trận ảnh vẽ trong 3
 ảnh đơn, xanh lục, và xanh dương

Các mã lệnh sáng**SURFL**

Một coi bóng 3 chiều với ảnh

SURFL(...) giống SURF(...) ngoài trừ vẽ một với
 ảnh sáng tại một nguồn sáng.

SURFL(Z), SURFL(X,Y,Z), SURFL(Z,S), và SURFL(X,Y,Z,S)
 rút lại ảnh hộp lâu. Nếu coi chế ảnh S, thực vectơ
 3 chiều $S = [S_x, S_y, S_z]$ ảnh chế ảnh hướng của nguồn
 sáng. Cũng coi thêm chế ảnh S trong hộp toạ độ
 cầu, $S = [AZ, EL]$. Bóng của ánh sáng vào tạo hộp
 các mã lệnh sáng với ảnh khuếch tán, ảnh phản
 chiếu và ảnh toạ sáng. Giải trẻ ngấm ảnh cho S
 là 45o theo chiều ngang tên tại hướng quan sát
 hiện tại. Dùng CLA, HOLD ON, VIEW(AZ,EL),
 SURFL(...), HOLD OFF để vẽ vẽ một chiếu với
 hướng quan sát (AZ,EL). Để nhúng ảnh vẽ lâu khác
 nhau với ảnh phủ sáng, ảnh phản chiếu và hộp sau

toái coĩ thãø ðùng SURFL(X,Y,Z,S,K) vãi K=[ka,kd,ks,spread]. Ðũa vào thãi tẽu của các âiãøm trong các ma trãun X,Y, vai Z ãø xãi ãnh các mặt nãm bản trong hay bản ngoãi. Thãi SURFL(X',Y',Z') nãu khãng thẽch ðùng kãut quãi nãy. ãø tên ãũc các phãip vectã của các mặt thç SURFL ãoĩ hoĩ cúp của các ma trãun ẽt nhãt là 3

SPECULAR

SPECULAR ãũ phãin chiãu

R = SPECULAR(Nx,Ny,Nz,S,V) trái vã ãũ phãin chiãu của mặt mặt coĩ các thành phãon của phãip vectã là [Nx,Ny,Nz]. S vai V ãnh hãĩng nguãon sãĩng hãĩng nhçn tãĩng ãĩng. Chuĩng coĩ thãø là các vectã 3 chiãu [x,y,z] hoãc các vectã 2 chiãu [Az, El] vãi Az là goĩc phãĩng về vai El là goĩc nãĩng trong hũ toũ ãũ cẽc. Phãin chiãu mãnh nhãt khi phãip vectã theo hãĩng (S+V)/2, vãi S là hãĩng nguãon vai V là hãĩng nhçn. Thành phãon toĩa sãĩng của mặt coĩ thãø ãũc chẽ ãnh bàong cãĩch thãm mặt tham sãu thãi 6 là SPECULAR (Nx,Ny,Nz,S,V,spread)

DIFFUSE

DIFFUSE ãũ khuãĩch tãĩn

R = DIFFUSE(Nx,Ny,Nz,S) trái vã ãũ khuãĩch tãĩn của mặt mặt coĩ các thành phãon của phãip vectã là [Nx,Ny,Nz]. S là mặt vectã 3 chiãu ãø xãi ãnh hãĩng nguãon sãĩng. S cũĩng coĩ thãø là mặt vectã 2 chiãu S = [Az,El] ãnh hãĩng hãĩng goĩc phãĩng về vai goĩc nãĩng.

ãnh luãt Lambert: R = cos(theta) vãi theta là goĩc giãĩa phãip vectã của mặt vai nguãon sãĩng

SURFNORM

SURFNORM Các phãip vectã của mặt 3 chiãu

[Nx,Ny,Nz] = SURFNORM(X,Y,Z) trái vã các thành phãon của phãip vectã của mặt 3 chiãu coĩ các thành phãon là (X,Y,Z). Phãip vectã chuãon hoĩa coĩ ãũ dài 1

[Nx,Ny,Nz] = SURFNORM(Z) trái vã các thành phãon của phãip vectã của mặt Z

Khãng coĩ ãũ sãu xuãt bản trãi thç SURFNORM(X,Y,Z) hoãc SURFNORM(Z) vẽ mặt vãi các phãip vectã vẽ ra tãi nũ. Các phãip vectã của mặt Các phãip vectã của mặt trái vã ðũa vào cãĩch ãiãũ chẽnh song lãũ phãĩng ðãi liãũ. ðùng SURFNORM(X',Y',Z') ãø ãũ hãĩng các phãip vectã

Taũ cãĩa sãø hçnh áĩnh vai các ãiãũ khiãon

FIGURE

FIGURE taũ các hçnh áĩnh (các cãĩa sãø ãũ thẽ)

Chẽnh lãũnh FIGURE mãĩ mặt hçnh áĩnh mãĩ (cãĩa sãø ãũ thẽ), vai trái vã mặt thẽi nguyãĩn

FIGURE(H) taôo măt hẹnh ánh thại H trăn hẹnh ánh hiấu taui của dăy lănh PLOT. Nău hẹnh ánh H chặ coi thẹ măt hẹnh ánh măt âăuc taôo ra bằng cặh đùng thei hẹnh ánh âăou tiăn coi thặ coi GCF traí vặ thei của hẹnh ánh hiấu taui. Thặc hiấu GET(H) âăo xem danh sặh cặc âăuc tên của cặc hẹnh ánh vặ cặc gặi trặ hiấu taui của chặng. Thặc hiấu SET(H) âăo xem danh sặh cặc âăuc tên của cặc hẹnh ánh vặ cặc gặi trặ hặp lặ của chặng

GCF

Lắy thei hẹnh ánh hiấu thặi

H = GCF traí vặ thei của thẹnh ánh hiấu thặi . Hẹnh ánh hiấu thặi lặ hẹnh ánh mặ cặc lănh vặ nhặ PLOT, TITLE, SURF, ... cho ra. Đùng lănh FIGURE âăo thay âăoi hẹnh ánh hiấu taui sang hẹnh ánh khặc, hoặc âăoi taôo ra măt hẹnh ánh măt

CLF

Xoặ hẹnh ánh

CLF xoặ táút cá cặc âăui tặng trăn hẹnh ánh hiấu taui

CLF RESET xoặ mặi thặ vặ cặng âăut lặi cặc âăuc tên của hẹnh ánh vặ cặc gặi trặ ngặc âềnh, trặ vặ trặ

CLOSE

Ăoặg cặc sặ chặ âềnh

CLOSE(H) ăoặg cặc sặ coi thei H

Chềnh lănh CLOSE ăoặg cặc sặ hẹnh ánh hiấu taui

CLOSE ăoặg cặc sặ khặg âiăou kiấu vặ khặg nhặc cặng nhặu

CLOSE('name') ăoặg cặc sặ coi tăn chặ âềnh

CLOSE ALL ăoặg táút cá cặc cặc sặ âang mặ

Taôo cặc trưc vặ cặc âiăou khiặ**SUBPLOT**

SUBPLOT taôo trưc coi tiău âăo

SUBPLOT (m,n,p), hoặc PLOT (mnp), phán cặc sặ hẹnh ánh thặnh măt ma trău cặ mxn cặc trưc nhặ, choũn trưc thặ p cho âăo thặ hiấu taui, vặ traí vặ thei trưc. Cặc trưc âăuc âăum đặc theo đặng trăn âềnh của cặc sặ hẹnh ánh, rặi đặng thặ 2,

...

Vặ đặ,

SUBPLOT(2,1,1), PLOT(income)

SUBPLOT(2,1,2), PLOT(outgo)

Vặ âăo thặ income ái nặ trăn cặc sặ vặ outgo ái nặ đặi

SUBPLOT(m,n,p), nău trưc âi coi thặ taôo ra trưc hiấu taui

SUBPLOT(H), vặi H lặ măt thei trưc thặ áy lặ cặh khặc âăo taôo ra măt trưc hiấu taui cho

âẽng vai âaĩnh sấu tãi trẽn xuấng. Truớc j ngang vai âaĩnh sấu tãi traĩi qua pháĩi

AXIS('xy') âầt MATLAB chấu âầ truớc ngắom âẽnh lại hầ toĩa âầ "Ầồ-caĩc". Gắc toĩa âầ áĩ goĩc trẽn bắ traĩi. Truớc x ngang vai âaĩnh sấu tãi traĩi qua pháĩi. Truớc y âẽng vai âaĩnh sấu tãi đầĩi lắ

AXIS('equal') thay âầĩ kếp thắĩc hầp truớc âầ x vai y bầng nhau. Lầnh nầy làm cho lầnh PLOT(SIN(X),COS(X)) trắg nhắ hẹnh trồĩn thay cho hẹnh oval

AXIS('square') taồ hầp truớc vuắg

AXIS('image'), vầĩ caĩc hẹnh áĩnh, taồ tề lầ phắn giáĩ nhắ hẹnh áĩnh

AXIS('normal') phườc hầĩ hầp truớc hiầũn taũĩ vầ kếp thắĩc áầỹ âũĩ vai xoĩa mầĩ haũn chấu trẽn caĩc âũn

về chia tề lầ. Lầnh nầy phườc hầĩ hiầũ lầc AXIS('square') vai AXIS('equal')

AXIS('image') âầ tề lầ phắn giáĩ vai caĩc giáĩ haũn truớc sao cho hẹnh áĩnh trong caĩc truớc hiầũn taũĩ coĩ caĩc âầĩom áĩnh hẹnh vuắg

AXIS('off') tầt táút cáĩ caĩc nhắĩn truớc vai caĩc âaĩnh dầũ

AXIS('on') bầt lắ táút cáĩ caĩc nhắĩn truớc vai caĩc âaĩnh dầũ

[S1,S2,S3] = AXIS('state') traĩ vầ caĩc chuầĩ biầũ hiầũn cáĩ âầt hiầũn taũĩ cuĩa 3 âầc tềnh truớc

S1 = 'auto' hoầc 'manual'

S2 = 'on' hoầc 'off'

S3 = 'xy' hoầc 'ij'

CAXIS

Giaĩ chia tề lầ truớc

CAXIS(V), vầĩ V làĩ vectắ 2 phầĩn táĩ [cmin cmax], âầt cáĩch chia tề lầ thuĩ cắg cuĩa giáĩ mầũ cho caĩc âầĩ tắũg SURFACE vai PATCH taồ ra bắĩ caĩc lầnh nhắ MESH, PCOLOR, vai SURF. cmin vai cmax âầc gắĩn cho mầũ áầũ vai mầũ cuầĩ cuĩa báĩg mầũ. Caĩc mầũ cho PCOLOR vai SURF âầc xắĩ âẽnh bầng báĩg tra cầũ bắ trong mầũn nầy. Caĩc giáĩ trề bắ ngoạĩ mầũn âầc xểĩ cho roĩ rắĩg

CAXIS('auto') âầt cáĩch chia truớc ngắũc vầ tầ âầũg

CAXIS, traĩ vầ vectắ đồĩng 2 phầĩn táĩ chắĩa [cmin cmax] âầg coĩ hiầũ lầc

CAXIS làĩ mầt M-file áầ âầt cáĩc âầc tềnh truớc làĩ CLim vai CLimMode

HOLD

Giãỉ hợnh veỉ hiãũn thẳi

HOLD ON giãỉ PLOT vài táút cáỉ cáỉc àầũc tênh trũc àầũ cáỉc lầũn veỉ tiấũp theo thẳm hợnh áỉnh vào hợnh áỉnh àaỉ coỉ

HOLD OFF traỉ vẫũ chấũ àầũ màũc àềnh, cáỉc lầũn PLOT xoỉa cáỉc hợnh áỉnh trẳaỉc vài àầũt lầũi táút cáỉ cáỉc àầũc tênh trũc trẳaỉc khi veỉ hợnh áỉnh mắỉ

HOLD, lầũt trầũng thẳi giãỉ hợnh áỉnh

HOLD khẳng áỉnh hẳaỉng cáỉc àầũc tênh chia miãũn tẳũ àầũng cuỉa cáỉc trũc

Lầũ yỉ thẳũt toaỉn: HOLD ON àầũt àầũc tênh NextPLOT cuỉa hợnh áỉnh hiãũn thẳi vài cáỉc trũc vào tênh chấũt "thẳm hợnh áỉnh" ("add"); HOLD OFF àầũt àầũc tênh NextPLOT cuỉa cáỉc trũc hiãũn thẳi vào tênh chấũt "thay thẳũ" ("replace")

Caỉc àầũi tẳũng theỉ àầũ theỉ

FIGURE , AXES, AXIS nhẳ trẳn

LINE

Tầũ cáỉc àầũi tẳũng àềaỉng thầũng báũc thấũp

LINE(X,Y) thẳm àềaỉng thầũng trong cáỉc vectỏ X vài Y vào cáỉc trũc hiãũn thẳi. Nầũ X vài Y laỉ cáỉc ma trầũn cũng cắi thợ màũi àềaỉng trẳn mắũt cầũt

LINE(X,Y,Z) tầũ cáỉc àềaỉng thầũng trong hầũ toũa àầũ 3 chẳũũ

LINE traỉ vẫũ mắũt vectỏ cầũt cáỉc theỉ chề àầũn cáỉc àầũi tẳũng LINE, màũi theỉ mắũt àềaỉng. Cáỉc àầũi tẳũng LINE laỉ con cuỉa cáỉc àầũi tẳũng AXES. Cầũp X,Y (bầũ ba X,Y,Z àầũi vẫũ 3 chẳũũ) coỉ thẳũ theo sau laỉ cáỉc cầũp tham sầũ/giaỉ trề àầũ chề àềnh cáỉc àầũc tênh thẳm cuỉa cáỉc àềaỉng. Cầũp X,Y (X,Y,Z àầũi vẫũ 3 chẳũũ) coỉ thẳũ boỉ qua toaỉn bầũ, vài táút cáỉ cáỉc chề àềnh àầũc tênh bàũng cáỉch đũng cáỉc cầũp tham sầũ/giaỉ trề. Thẳũc hiãũn GET(H), vẫũ H laỉ theỉ àềaỉng, àầũ xem đẳnh sắch cáỉc àầũc tênh àầũi tẳũng àềaỉng vài cáỉc giaỉ trề hiãũn thẳi cuỉa chũũng. Thẳũc hiãũn SET(H) àầũ xem đẳnh sắch cáỉc àầũc tênh àầũi tẳũng àềaỉng vài cáỉc giaỉ trề hầũp lầũ

TEXT

Thẳm vàn báỉn vào hợnh áỉnh hiãũn thẳi

TEXT(X,Y,'string') thẳm vàn báỉn trong cầũp dầũu nhẳũ vào tầũi về trề (X,Y) trẳn cáỉc trũc hiãũn thẳi, vẫũ (X,Y) theo àỉn về lầũy trong lầũn PLOT hiãũn thẳi. Nầũ X vài Y laỉ cáỉc vectỏ , thợ TEXT gắ vàn báỉn tầũi táút cáỉ cáỉc về trề àềũc cho. Nầũ 'string' laỉ mắũt máỉng cũng sầũ đũng vẫũ àầũ đẳi cuỉa X vài Y, thợ TEXT àaỉnh dầũu màũi àỉầũm vẫũ cầũt tẳũng àỉũg cuỉa máỉng 'string'

TEXT(X,Y,Z,'string') thẳm vàn báỉn trong hầũ toũa àầũ 3 chẳũũ

AXES

theo

tää 2

lat'.

 \hat{a}_{ii}

caic

 $\hat{\mu} \hat{\sigma}^2$

caic

SURFACE(X,Y,Z) dùng $C = Z$, vẽ vảy cán xẻng vãi
 ầu cao mặt

Xem SURF ầu coi ầu ầu cái dưng khải nhau mai
 X,Y,Z,C coi thỡ lúy

SURFACE trái vớ mặt thei chề ầun mặt ầui tểung
 SURFACE. Cái ầui tểung SURFACE lại con của cái
 ầui tểung AXES

Cái ầui sủ cho SURFACE coi thỡ theo sau lại cái
 cùp tham sủ/giải trề ầu chề ềnh cái ầuc tênh thm
 cho mặt. Cái ầui sủ X,Y,Z,C cho SURFACE coi thỡ
 boi qua toàn bủ, vai tút cái cái ầut tênh chề ềnh
 bông cách dùng cái cùp tham sủ/giải trề
 AXIS, CAXIS, COLORMAP, HOLD, SHADING vai VIEW ầut cái
 ầuc tênh hnh ánh, cái trưc, vai mặt coi hiều
 lể cho viể hiể thề của SURFACE

Thức hiể GET(H), vãi H lại mặt thei mặt, ầu xem
 danh sách cái ầuc tênh ầui tểung mặt vai cái
 giải trề hiể thằi của chểng. Thức hiể SET(H)
 ầu xem danh sách cái ầuc tênh vai cái giải trề
 ầuc tênh hủp lủ của ầui tểung mặt

IMAGE

Hiể thề hnh ánh (tao ầui tểung hnh ánh)

IMAGE(C) hiể thề ma trậ C dưng hnh ánh. Mũi
 phỏn tẻi của C chề ềnh mũi của mặt mnh chể
 nhủt trong hnh ánh. Mũi phỏn tẻi của C ầuc
 dùng nhề cái chề sủ trong bẩng mũi hiể thằi ầu
 xải ềnh mũi. Nủ [M N] = SIZE(C), thề cái giải
 hủn trưc lại [0.5 N+0.5 0.5 M+0.5 bẩi vủ mũi
 phỏn tẻi trong C lại tám coi toa ầu nguyể giể 1
 vai M hoặ N

IMAGE(X,Y,C), vãi X vai Y lại cái vectỏ, chề ềnh
 viể ầunh nhủn cái trưc X vai Y, nhể cho ra
 cùng hnh ánh nhề IMAGE(C). Chề coi cái phỏn tẻi
 ầu vai cuểi của X vai Y ầuc dùng cho cái giải
 hủn trưc. Trong cái trể hủp này, để
 ầu/cuểi của C lại tám trể giải trề ầu/cuểi của
 Y, vai cểt ầu/cuểi của C lại tám trể giải trề
 ầu/cuểi của X, vãi cách thm vào ải cái ầu
 mũi ầu ầum bẩ cách ầu để theo cái trưc X
 vai Y

IMAGE trái vớ mặt thei chề ầun mặt ầui tểung
 IMAGE. C hoặ bủ ba X,Y,C coi thỡ theo sau lại cái
 cùp tham sủ/giải trề ầu chề ềnh cái ầuc tênh thm
 cho hnh ánh. C hoặ bủ ba X,Y,C coi thỡ boi qua
 toàn bủ, vai tút cái cái ầut tênh chề ềnh bông
 cách dùng cái cùp tham sủ/giải trề. IMAGE ầut
 ầuc tênh trưc YDir vào giải trề ầu ('reverse'),
 vai ầuc tênh View vào [0 90]. ầui tểung hnh ánh
 sei khể ầu ra cái goể View khải. ầu lúy ầuc
 hiể lể tể tể quay mặt hnh ánh, dùng SURF
 vãi ầu xủ trể hoặ PCOLOR
 Thức hiể GET(H), vãi H lại mặt thei hnh ánh, ầu
 xem danh sách cái ầuc tênh ầui tểung hnh ầu vai
 cái giải trề hiể thằi của chểng. Thức hiể

SET(H) sẽ xem danh sách các thuộc tính và các giá trị thuộc tính hấp thụ của đối tượng hính

UICONTROL

UICONTROL tạo đối tượng khi tạo giao diện người dùng
UICONTROL

('PropertyName1',value1,'PropertyName2',value2,...)

tạo đối tượng khi tạo giao diện người dùng trong cửa sổ hính hính hính và trái vảo mặt thế cho nó

Các thuộc tính của UICONTROL có thể được đặt bất cứ lúc nào ra bằng cách dùng tên cấp đối số PropertyName/PropertyValue trong UICONTROL, hoặc thay đổi sau đó bằng cách dùng lệnh SET. Thử hính GET(H) sẽ xem danh sách các thuộc tính của UICONTROL và các giá trị hính của nó. Thử hính SET(H) sẽ xem danh sách các thuộc tính của UICONTROL và các giá trị hấp thụ

UIMENU

UIMENU tạo menu giao diện ngoài dùng
 UIMENU('PropertyName',value1,'PropertyName2',value2,...
 .) tạo menu trên menu bar ở trên bên của cửa sổ
 hệ thống hiển thị tại trái vỏ màn hình cho nó
 UIMENU(H,...) tạo màn hình menu mới với H là cha. H
 phải là màn hình hệ thống hoặc menu. Nếu H là
 hệ thống thì UIMENU là menu trên menu bar ở
 trên bên của cửa sổ. Nếu H là màn hình menu trên
 menu bar thì menu mới là menu thả xuống bên
 dưới menu bar

Các thuộc tính của menu có thể được đặt
 ra bằng cách dùng từng cặp `propertyName/propertyValue`
 trong UIMENU, hoặc thay
 đổi sau đó bằng cách dùng lệnh SET. Thuộc
 tính GET(H) để xem danh sách các thuộc
 tính của UIMENU và các giá trị hiển thị của nó
 Thuộc tính SET(H) để xem danh sách các
 thuộc tính của UIMENU và các giá trị
 hiện tại

Các thao tác vẽ thể tạo**SET**

Đặt các thuộc tính của

SET(H,'PropertyName',PropertyValue) đặt
 giá trị thuộc tính để đặt cho các thuộc
 tính tạo tạo vẽ. H có thể là màn hình
 vectơ các thể, trong trường hợp này SET
 đặt các giá trị thuộc tính cho
 tất cả các thuộc tính

SET(H,'PropertyName1',PropertyValue1,'PropertyName2',Pr
 opertyValue2,...) đặt nhiều giá trị thuộc
 tính với
 một lần

SET(H,'PropertyName') hiển thị các giá trị
 có thể đặt cho thuộc tính để đặt của
 thuộc tính vẽ H

SET(H) hiển thị tất cả các thuộc tính và các
 giá trị có thể đặt của chúng cho thuộc
 tính vẽ H. Giá trị mặc định để đặt
 thuộc tính vẽ trong thuộc tính cha, ứng
 với tên của thuộc tính bằng cách đặt
 propertyName vào bên phải cách với chuỗi
 'Default', kiểu thuộc tính, và tên thuộc
 tính. Về
 mặt, để đặt màu nền thuộc tính của các
 thuộc tính và
 bên vào màu nền trong cửa sổ hệ thống
 hiển thị:

```
set(gcf,'DefaultTextColor','red')
```

Ba chuỗi có ý nghĩa để đặt cho các giá trị
 thuộc tính:

'default' - dùng giá trị mặc định (từ thuộc
 tính cha)

'factory' - dùng giá trị mặc định của hãng

'remove' - xóa giá trị mặc định

GET

Lấy các thuộc tính của đối tượng

V = GET(H, 'PropertyName') trả về giá trị của thuộc tính cho trước cho đối tượng đã cho với tên H

GET(H) hiển thị tất cả các tính chất của đối tượng đã cho hiển thị của chúng cho đối tượng đã cho với tên H

RESET

Đặt lại trục hoành

RESET(H) đặt lại tất cả các thuộc tính, trả về trục, của đối tượng với tên H về giá trị mặc định. Về đầu, RESET(GCA) đặt lại các thuộc tính của các trục hiển thị. RESET(GCF) đặt lại các thuộc tính của trục hiển thị

DELETE

Xoá một tập hợp một đối tượng đã cho

DELETE file_name xoá tập con tất cả

DELETE(H) xoá đối tượng đã cho với tên H. Nếu đối tượng là rỗng, tập rỗng được tạo ra xoá tập (không chắc chắn)

GCO

GCO Tên của đối tượng hiển thị

OBJECT = GCO trả về đối tượng hiển thị trong trục hiển thị

OBJECT = GCO(FIGURE) trả về đối tượng hiển thị trong trục của trục. Đối tượng hiển thị cho một trục để lại đối tượng cuối cùng của trục

DRAWNOW

DRAWNOW Hoàn thành một trục vẽ chưa xử lý

DRAWNOW "đọc biểu đồ hàng đầu" và làm MATLAB cập nhật màn hình

Có 4 biểu đồ của MATLAB đọc biểu đồ hàng đầu và vẽ trên màn hình:

- trả về tài liệu của MATLAB
- gộp phải là PAUSE
- thức hiển thị là GETFRAME
- thức hiển thị là DRAWNOW

DRAWNOW('DISCARD') làm cho trục tin được cập nhật như các biểu đồ hàng đầu được đọc ra, nhưng không đọc chúng thức. Điều này cần thiết cho việc in bản sao cũng khi trục trục trục InvertHardcopy là bất


```

PRINT <filename> -s<system name> in hầu
SIMULINK chè àenh
Cui phaíp: PRINT [ -ddevice] [ -options ]
<filename>

```

Caïc thiăút bi PostScript coi thăø duìng lại:

```

-dps      - PostScript cho caïc maÿ in àen-
trầong
-dpsc     - PostScript cho caïc maÿ in màu
-dps2     - PostScript mæic 2 cho caïc maÿ
in àen-trầong
-dpsc2    - PostScript mæic 2 cho caïc maÿ
in màu
-deps     - Encapsulated PostScript (EPSF)
-depsc    - Encapsulated Color PostScript
(EPSF)
-deps2    - Encapsulated Level 2 PostScript
(EPSF)
-depsc2   - Encapsulated Level 2 Color
PostScript (EPSF)

```

Caïc thiăút bề gầon liăõn thăm lại:

```

-dhpgl    - HPGL tẳang thêch vắi maÿ veì
Hewlett-Packard 7475A
-dill     - Adobe Illustrator 88 compatible
illustration file
-dmfile   - M-file chuí yăúu taôu laúi hợnh
aính vại con cuía noì

```

Khăng coi chặic nàng àáõy àuì, vại tẳang khăng cung cúp

Caïc thiăút bề GhostScript thăm lại:

```

-dlaserjet - HP LaserJet
-dljetplus - HP LaserJet+
-dljet2p   - HP LaserJet IIP
-dljet3    - HP LaserJet III
-dcdeskjet - HP DeskJet 500C vắi màu 1
bit/âiăøm aính
-dcdjcolor - HP DeskJet 500C vắi màu 24
bit/âiăøm aính vại màu đao àăũng chấút lắũng cao
(Floyd-Steinberg)
-dcdjmono  - HP DeskJet 500C chè in àen
-ddeskjet  - HP DeskJet vại DeskJet+
-dpaintjet - Maÿ in màu HP PaintJet
-dpjetxl   - Maÿ in màu HP PaintJet XL
-dbj10e    - Canon BubbleJet BJ10e
-dln03     - DEC LN03
-depson    - Epson-maÿ in tẳang thêch ma
trầun âiăøm (9- hoàuc 24-kim)
-deps9high - Epson-tẳang thich 9-kim, cheìn
doìng
-dgif8     - daũng tấũ GIF 8-bit màu
-dpcx16    - daũng tấũ PCX (EGA/VGA, 16
màu)
-dpcx256   - daũng tấũ PCX (256 màu)

```

Caïc lắũa choũn khaïc lại:

```

-append    - Thăm, khăng àeì, hợnh aính
vào tấũ PostScript

```

```
printopt      caïc  giaỉ  trẽ  màuc  âênh  cuía  cáuu  hñnh
maĩy  in  cuïc  bầu
```

Hæåĭng giáuý cuía baín sao cæĭng

ORIENT PORTRAIT traí vẫ hẵing màuc âenh PORTRAIT

ORIENT TALL toàin bǎu giǎúy theo hǎǎing portrait

ORIENT, trái vảo maut chuau chæia hæing giáu in:
 PORTRAIT, LANDSCAPE hoatc TALL

ORIENT	lại	mäüt	M-file	âãø	âàût	caïc	âàüc	tên
PaperOrientation				vàì			PaperPosition	

CAPTURE Láuý main hẹnh củia hẹnh ánh hiấu thầi

CAPTURE(FIG) taôu hợnh ánh mảii chặi baín sao cuía
hợnh ánh chề aềnh, kặo cái mỗii uimenu vài mỗii
uicontrol

CAPTURE, kháng ảnh sáu nháp, lấuy hẹnh ảnh hiếu
thời

[X,map] = CAPTURE trái vảo maut ma trauu henh ainh
vai baing mau mai khang hiaon the noi

Caic phim vai hcnh ánh ảnh

MOVIEIN

MOVIEIN Khái áăöü bâu nhái áăöü lêu caic khung phim

M = MOVIEIN(N) taôu măt ma tráuñ áúi lăñ áăöü giăi N khung phim đău vào caic trưc hiăñ thăi. Ma tráuñ că áúi đöng áăöü lêu N báñ sao cuía caic giăi trê xuăt tăi GETFRAME, măt báñ sao măt căt

M = MOVIEIN(N,H) vai M = MOVIEIN(N,H,RECT) truyăñ caic thei áăöü hoă tuý choñ, H, vai caic vê trê hcnh chăi nhăt tuý choñ, RECT, vào GETFRAME(H,RECT) qua caic áăüi tăăng áăöü hoă chē āēñ khăc văi caic trưc măt āēñ hiăñ thăi. Vê đū: áăöü phăi sinh măt phim văi n khung,

```
M = moviein(n);
for j=1:n
    Lăññ PLOT
    M(:,j) = getframe
end
movie(M)
```

Áăöü phăi sinh măt coi caic khung chăi toăñ bâu hcnh ánh, khăng āūñg caic trưc hiăñ thăi,
M = moviein(n,gcf)

GETFRAME

GETFRAME Lăý khung phim

GETFRAME trái văo măt vectă căt văi măt khung phim. Khung lài măt ánh chuúp nhanh (pixmap) cuía caic trưc hiăñ thăi. GETFRAME thăăng đūng trong vöng lăp FOR áăöü taôu ma tráuñ phim M răi quay ngăăc lăi đūng MOVIE. Vê đū:

```
M = moviein(n);
for j=1:n
    Lăññ PLOT
    M(:,j) = getframe
end
movie(M)
```

Quan tröng trong viăc phăñ bâu trăăc ma tráuñ phim M trăăc khi xáy đăng phim nhăm trăññ săi đūng quăi nhiăöü bâu nhái. Hàm MOVIEIN coi thăö đūng cho căñg đūng

GETFRAME(H) lăý khung cuía áăüi tăăng H, H lài măt thei chē āăñ măt găc, măt hcnh ánh, hoăc măt trưc

GETFRAME(H,RECT) chē āēñ hcnh chăi nhăt áăöü chēi hcnh ánh, tăăng áăüi so văi goic trăn băñ trái cuía áăüi tăăng H vai theo āññ vê cuía āăc tēñ Units cuía năi

[X,MAP] = GETFRAME(H) hoăc [X,MAP] = GETFRAME(H,RECT) trái văo khung nhă ma tráuñ chē său hcnh ánh X vai báñg măt MAP. H coi thăö lài thei cuía măt hcnh ánh hoăc măt trưc

MOVIE

RBBOX

Häüp bàng cao su

RBBOX(RECTI, XY) lăön văút häüp bàng cao su trăn hệnh ánh , kháii áăöu văii häüp RECTI, vai băt áăöu văút tai áiăøm XY. Nuít chuăút pháii giăi khi RBBOX áăăüc goüi. RBBOX trái văö văii biăún cẩu nuít chuăút. Nhă măt kăút quái, RBBOX coi thăö áăăüc dùng trong măt M-file, cùng văii WAITFORBUTTONPRESS, áăö áiăöu khiăön xăi lyi áăüng. RECTI vai XY theo áân vë cuía hệnh ánh hiăün thăii

ROTATE

Quay măt áăúi tăăüng măt goïc alpha theo hăăng chề ênh

ROTATE(H,AZEL,ALPHA,ORIGIN) quay áăúi tăăüng H goïc ALPHA quanh măt truíc mã tai bắi AZEL, lai vectă 2 pháön tăi găöm goïc phăng vë vai goïc năng hoăüc măt vectă 3 chiăöu. Áăúi său tuyi choün ORIGIN lai vectă 3 pháön tăi dùng lai găúc quay

TERMINAL

TERMINAL Áăút kiăöu áăöu cuăúi áăö hoă

TERMINAL áăút ra măt menu áăö hoă caic kiăöu áăöu cuăúi vai nhăöc áăö choün. Răöi năi áăút cáúu hệnh cho MATLAB. Cúng coi thăö chề ênh áăö cuăúi nhă tham săú thăi nhăút. Coi thăö thăm caic cai áăút áăö cuăúi cáön thiăút cho chề ênh mã phóing hay áăö cuăúi vaio tăúp năy

Caic áăö cuăúi áăö hoă áang áăăüc cung cáúp lai:

tek401x - Tektronix 4010/4014, tek4100 - Tektronix 4100 , tek4105 - Tektronix 4105

retro - retrographics card , sg100 - Selanar graphics 100, sg200 - Selanar graphics 200

vt240tek - VT240 & VT340 Tek mode , ergo - ergo terminal , graphon - graphon terminal

citoh - C.Itoh terminal , xtermtek - xterm, Tektronix graphics

wyse - Wyse WY-99GT , kermi - MS-DOS Kermit 2.23 , hp2647 - Hewlett-Packard 2647

versa - Macintosh with VersaTerm (Tektronics 4010/4014)

versa4100 - Macintosh with VersaTerm (Tektronics 4100)

versa4105 - Color/Grayscale Macintosh with VersaTerm (Tektronics 4105)

hds - Human Designed Systems

UIPUTFILE

UIPUTFILE Láúy tăn tăúp bàng caich hiăön thề măt häüp häüi thoaüi

[FILENAME, PATHNAME] = UIPUTFILE('initFile', 'dialogTitle') hiăön thề măt häüp häüi thoaüi vai trái văö caic chuăúi tăn tăúp vai áăăüng dăün. Tham său initFile xăic áênh hiăön thề ban áăö cuía caic tăúp trong häüp häüi thoaüi. Cho pheïp chề ênh tăn tăúp áăöy áúi cúng nhă kyi tăü áăüi diăün. Vë duü,

```
Vê duû: [newmatfile, newpath] = iputfile('*.mat',
'Save As');
```

```
[FILENAME, PATHNAME] = UIGETFILE('filterSpec',
'dialogTitle', X, Y) hiãøn thẽ mãüt häüp häüi thoaüi
ääø ngæãli ðùng äæa vaìo (fill in), vaì traí vãø caïc
chuäüi tãn tãúp vaì äæãng dàùn. Traí vãø thaình cãng
chè nãúu tãúp ääi coì. Nãúu ngæãli ðùng choün mãüt
tãúp chæa coì thç hiãøn thẽ mãüt thãng baìo lãui vaì
traí vãø äiãöu khiãøn cho häüp häüi thoaüi. Sau äoì
ngæãli ðùng coì thãø vaìo tãn tãúp, hoàüc áún nũit
Cancel. Táút cá caïc tham sãú laì tuý choün, nhæng
nãúu ðùng mãüt tham sãú thç táút cá caïc tham sãú
træãic cüng pháí ðùng hãút. Tham sãú filterSpec
xaic äènh tãn caïc tãúp ban ääöu trong häüp häüi
thoaüi. Vê duü '*.*m' liãút kã táút cá caïc tãúp M-
files cuía MATLAB. Tham sãú 'dialogTitle' laì chuäüi
chæia tiãu äãö cuía häüp häüi thoaüi. Caïc tham sãú X
và Y xaic äènh vê trê ban ääöu cuía häüp häüi thoaüi
theo äãn vê äiãøm. Mãüt sãú hãu thãúng khãng cung cáúp
tuý choün naìy. Biãún xuát FILENAME laì chuäüi chæia
tãn tãúp äæãüc choün trong häüp häüi thoaüi. Nãúu
ngæãli ðùng áún nũit Cancel hoàüc nãúu coì lãui, thç
äæãüc ääüt giaì trẽ 0. Biãún xuát PATH laì chuäüi
chæia tãn äæãng dàùn choün trong häüp häüi thoaüi.
Nãúu ngæãli ðùng áún nũit Cancel hoàüc nãúu coì lãui,
thç äæãüc ääüt giaì trẽ 0
```

WHITEBG látut maiu năon cuia hênh ánh hiăun thăi giăi âen vai trăông, vai thay âăoi caic âăuc tênh khaic âăo hênh ánh âăuc nhên roi hân. Hân năia, caic âăuc tênh ngăom âênh cuia găuc âăuc âăut âăo dăy lăunh

PLOT tiếu sấu trong hệnh ánh hiân thài vai caic hệnh ánh mải dùng màu năn mải

WHITEBG(FIG) thay âõ caic hệnh ánh trong vecta cút FIG. Kõ cái thei cĩa sã gũc (0) ánh hã ánh caic âũc tên màu âẽnh cho caic cĩa sã mải hoũc cho CLF RESET

WHITEBG(FIG,C) hoũc WHITEBG(C) âũt màu năn màu âẽnh vào C vai thay âõ caic âũc tên khaic âõ hệnh ánh âũc nhẽn roĩ hãn. C coĩ thã lại mũt màu 1x3 rgb hoũc mũt chuũ màu nhẽ 'white' hoũc 'w'

ZOOM

Bung to vai thu nhĩ mũt hệnh veĩ 2 chiũ

ZOOM ON bũt trũng thũi bung to/thu nhĩ hệnh ánh hiân thũi. Ấn nũt trũi âõ thu nhĩ. Ấn nũt phũi âõ bung to. Mũi khi ấn, caic giũi hũn trũc seĩ thay âõ theo tẽ lũ gũp âĩ (bung to hoũc thu nhĩ). Cũg coĩ thã ấn vai lĩ âõ thay âõ trong mũt vũng

ZOOM OFF tũt trũng thũi bung to/thu nhĩ

ZOOM khũg âũi sũ âõ lũt trũng thũi zoom. ZOOM OUT trũ vũ hệnh ánh ban âũ (âũy âũ)

waitforbuttonpress Chũ ấn phẽ/nũt chuũ trũ hệnh ánh

Caic ma trũn âũc biũt

COMPAN

Ma trũn COMPANION

A=COMPAN(P) lại ma trũn COMPANION cũa âa thũc coĩ caic hũ sũ P, ma trũn coĩ ãng thũ nhũt lại - u(2:n)/u(1) vũi u lại veĩtũ hũ sũ cũa âa thũc P. Caic giũ trẽ riũg cũa A lại caic nghiũm cũa P.

GALLERY

GALLERY Cũp ma trũn kiũm tra nhĩ

GALLERY(3) lại ma trũn vũg âĩũ kiũn xũu cũp 3

GALLERY(5) lại mũt bài toũn giũ trẽ riũg âũg quan tám. Thũ tẽm caic giũ trẽ riũg vai caic vecta riũg chẽnh xũ cũa nũ

HADAMARD

HADAMARD(N) lại ma trũn Hadamard cũp N, âũ lại ma trũn H coĩ caic phũn tũ 1 hoũc -1 mũ H'*H = N*EYE(N). Mũt ma trũn Hadamard cũp N vũi > 2 chẽ tũn tũi nũu REM(N,4)=0 (N lại bũi cũa 4). Hũm nũy chẽ xũ lý caic trũg hũp N, N/12 hoũc N/20 lại lũy thũ cũa 2

HANKEL

Ma trũn Hankel

HANKEL(C) lại ma trũn vũg Hankel coĩ cũt âũũ tiũn lại C vai caic phũn tũ đũi âũg chũ phũ bũng 0

HANKEL(C,R) là ma trận Hankel có cắt đầu tiên là C và dòng cuối cùng là R. Ma trận Hankel đầu tiên, các hàng chéo ngược không rỗng, và có các phần tử $H(i,j) = R(i+j-1)$

HILB

Ma trận Hilbert

HILB(N) là ma trận vuông cấp N có các phần tử $1/(i+j-1)$, đây là về đầu nội tiêu chuẩn về ma trận đầu tiên kiêu xú. Xem thêm INVHILB

Đây cũng là một về đầu tắt về hiều quá của phong cách lập trình MATLAB với qui các vòng lặp FOR hay DO được thay bởi các lệnh vectơ hóa. Cách này nhanh hơn, nhưng ít tốn vùng nhớ

INVHILB

INVHILB Ma trận đảo Hilbert

INVHILB(N) là nghịch đảo của ma trận vuông cấp N có các phần tử $1/(i+j-1)$, đây là về đầu nội tiêu chuẩn về ma trận đầu tiên kiêu xú. Kết quả chênh lệch nếu $N < 15$

KRON

Tích ten-xơ Kronecker

KRON(X,Y) là tích ten-xơ Kronecker của X và Y. Kết quả là một ma trận lớn tạo nên bằng cách lấy tích của các tích có thể giữa các phần tử của X và các phần tử của Y. Về đầu, nếu X là ma trận cỡ 2×3 , thì KRON(X,Y) là

$$\begin{bmatrix} X(1,1)*Y & X(1,2)*Y & X(1,3)*Y \\ X(2,1)*Y & X(2,2)*Y & X(2,3)*Y \end{bmatrix}$$

Nếu X hoặc Y thưa, thì chế nhán các phần tử khác 0 và kết quả là ma trận thưa

MAGIC

Hình vuông ảo thuật (ma phương)

MAGIC(N) là một ma trận vuông cấp N tạo nên tại các số nguyên từ 1 đến N^2 sao cho tổng các hàng, cột và đường chéo bằng nhau

PASCAL

PASCAL Ma trận Pascal

PASCAL(N) là ma trận Pascal cấp N: một ma trận các ảnh đẳng cấp đầu tiên có các phần tử nguyên, tạo nên tại tam giác Pascal. Nghịch đảo của nó có các phần tử nguyên

PASCAL(N,1) là thừa số tam giác dưới Cholesky của ma trận Pascal. Là nghịch đảo của nó

PASCAL(N,2) là phiên bản chuyển về và hoán vị của PASCAL(N,1), là căn bậc ba của ma trận ảnh về

ROSSER

Bài toán kiểm tra giả trẻ riêng đầu tiên càng đầu tiên

Ma trận này là một thách thức đầu tiên với nhiều thuật toán giả trẻ riêng ma trận. Nhưng thuật

toaın Francis QR, nhæ âaı hoαιν thiăın bảıı Wilkinson vai cail âaút trong EISPACK vai MATLAB khăng coi vầu âăö gç. Ma trầu nây lai ma trầu vuăng cấp 8 gầm caıc phầon táı nguyăın. Noı coi:

- * Măút giai trê nguyăın keıp
- * 3 giai trê riăng gần bầong nhau
- * Caıc giai trê riăng lăın ngắăuc đáuú
- * Măút giai trê riăng bầong 0
- * Măút giai trê riăng nhoı khaıc 0

TOEPLITZ

TOEPLITZ Ma trầu Toeplitz

TOEPLITZ(C,R) lai ma trầu Toeplitz khăng âăúi xăıng coi C lai cầt cuăúi cùıg vai R lai đòıg cuăúi cùıg cuía

TOEPLITZ(C) lai ma trầu Toeplitz âăúi xăıng (hay lai ma trầu Hermitian)

VANDER

Ma trầu Vandermonde

VANDER(C) traı vắö ma trầu Vandermonde coi cầt thăı hai vai cầt cuăúi cùıg lai C. Cầt thăı j cuía ma trầu Vandermonde âắăuc cho bảıı $A(:,j) = C^{(n-j)}$

WILKINSON

WILKINSON Ma trầu kiăøm tra giai trê riăng cuía Wilkinson

WILKINSON(n) lai J. H. Ma trầu kiăøm tra giai trê riăng cuía Wilkinson, $Wn+$. Ááy lai ma trầu âăúi xăıng gầm 3 âắăıng cheıö vắıı caıc cầp gần bầong nhăıg khăng âuııg bầong caıc giai trê riăng. Trắăıng hắıp đừıg thắăıng xuyăın nhắút lai WILKINSON(21). Vê đừı, WILKINSON(7) lai

3	1	0	0	0	0	0
1	2	1	0	0	0	0
0	1	1	1	0	0	0
0	0	1	0	1	0	0
0	0	0	1	1	1	0
0	0	0	0	1	2	1
0	0	0	0	0	1	3

Caic haım ăm thanh tăøng quait

SOUND

Chuyăøn vectă sang ăm thanh

SOUND(Y) găi tên hiău trong vectă Y sang loa trăn caic loaui maıy SPARC, HP, SGI, PC, vaı Macintosh. Vectă âăüc chia thang ăm tău âăüng âăø cung cúp biăñ âău lăın nhăút. Ăm thanh âăüc chăi văi tăuc âău mầu mầu âênh. Trăn maıy SPARC, tăuc âău mầu cău âênh laı 8192 Hz. Trăn maıy Macintosh, tăuc âău mầu cău âênh laı 22.255K Hz

SOUND(Y,FS), trăn maıy Macintosh, PC, vaı SGI, chăi ăm thanh văi táøn său mầu laı FS Hz

Vectă Y âăüc tău âăüng taøu thang ăm âăø caic giaı trê lăın nhăút vaı nhoı nhăút trong Y ăıng văi miăøn giaı trê vaıo nhoı nhăút vaı lăın nhăút cho pheıp băi phăøn căıng. Trăn maıy Macintosh vaı SGI, bău âiăøu khiăøn volume trăn Control Panel xăc âênh báuc ăm thanh cuăui cıng

SAXIS

Chia truêc ăm thanh. Haım SOUND chia thang ăm thăng thăıng vectă nhăp Y cuıa noı âăø caic giaı trê lăın nhăút vaı nhoı nhăút trong Y ăıng văi miăøn giaı trê vaıo nhoı nhăút vaı lăın nhăút cho pheıp băi phăøn căıng SAXIS(V), văi V laı vectă 2 chiăøu $V = [SMIN \ SMAX]$, huıy viăüc chia thang ăm tău âăüng cuıa haım SOUND vaı âăut thang ăm âăø SMIN vaı SMAX ăıng văi miăøn giaı trê vaıo nhoı nhăút vaı lăın nhăút cho pheıp băi phăøn căıng. Caic giaı trê ngoăi miăøn naıy bê xeın âi

SAXIS('auto') âăut viăüc chia truêc ăm thanh ngăüc văo caıch chia tău âăüng

SAXIS, tráı văo vectă ðoıng 2 phăøn táı chăıa [SMIN SMAX] laı hiău ăıng hiăun taui. SAXIS hoăn toăın giăıng nhê CAXIS vaı AXIS, chê khăc laı viăüc chia truêc ăıp ðuıng cho ăm thanh chăi khăng phăi cho caic giăi haın văo âăø thê

Caic haım ăm thanh chi tiăút

AUWRITE

Ghi tăp ăm thanh maı hoıa theo luăut mu-law (chê tên căın trăn)

AUWRITE(Y,'filename') chuyăøn Y sang caic byte maı hoıa theo luăut mu-law vaı ghi vaıo tăp ăm thanh chê âênh. Nău khăng cho tăn tăp, maı ðuıng /dev/audio thê hoăc laı trăn maıy cıc bău, hoăc laı trăn maıy nhoı chê âênh băi biăun maı trăıng DISPLAY

AUREAD

Ăuê tăp ăm thanh maı hoıa theo luăut mu-law (chê tên căın trăn)

Y = AUREAD('filename') àoüc tấủ ám thanh 'filename' vài chuyãỏn đặi liãủu tặi cặc byte mặi hoặi theo luấủ mu-law sang tên hiãủu trong miãỏn giáỉ trặ -1 <= Y <= 1.

WAVWRITE

Lặ cặc tấủ ám thanh đặủg WAV củi Windows 3.1

WAVWRITE(y,Fs,wavefile) lặ tấủ đặủg ".WAV" ặặủ chặ ặềnh bặi "wavefile". Cặc ặặủ sặủ nhặủp cho WAVWRITE nhặ sặ:

Y Đặ liãủu mặủ ặặỏ gặ (8 bit max)

Fs Tặc ặặủ mặủ

wavefile Chặủi ký tặ lặ tặ tấủ ".WAV" ặặỏ tặỏ rặ

Lặ yặ: WAVWRITE sặ tặỏ rặ mặủ tấủ kặh sặng ặặ 8-bit. Đặ liãủu mặủ kặh phặi 8-bit bặ cặỏ

WAVREAD

WAVREAD Nặ cặc tấủ ám thanh đặủg WAV củi Windows 3.1

[y]=WAVREAD(wavefile) nặ tấủ đặủg ".WAV" ặặủ chặ ặềnh bặi "wavefile", tặi vặ đặ liãủu mặủ trong biặủn "y". Phặỏn mặi rặủg ".WAV" củi tặ tấủ lặ tặủ chặủn

[y,Fs]=WAVREAD(wavefile) nặ tấủ đặủg ".WAV" ặặủ chặ ặềnh bặi "wavefile", tặi vặ đặ liãủu mặủ trong biặủn "y" vài tặc ặặủ mặủ trong biặủn "Fs"

[y,Fs,Format]=WAVREAD(wavefile) nặ tấủ đặủg ".WAV" ặặủ chặ ặềnh bặi "wavefile", tặi vặ đặ liãủu mặủ trong biặủn "y" , tặc ặặủ mặủ trong biặủn "Fs" , vài thặg tặ tấủ đặủg ".WAV" trong biặủn "Format". Thặg tặ ặặủ tặi vặ lặ mặủ vectặ 6 phặỏn tặi theo thặ tặ nhặ sặ:

Format(1) Đặủ đặ liãủu (lặủ chặ PCM)

Format(2) Sặ kặh

Format(3) Tặc ặặủ mặủ (Fs)

Format(4) Trung bặủ byte/s(mặủ)

Format(5) Kặủi xặủ đặ liãủu

Format(6) Cặc bit/mặủ

Lặ yặ: WAVREAD chặ cặủ cặủ đặ liãủu kặh ặặ 8-bit

MU2LIN

Chuyãỏn mu-law sang tặủủn tên

Y = MU2LIN(MU) chuyãỏn cặc tên hiãủu ám thanh mặi hoặi 8-bit theo luấủ mu-law, lặ "cặc giáỉ trặ rặỏn" trong miãỏn giáỉ trặ 0 <= MU <= 255, sang tên hiãủu biặ ặặ trong miãỏn -s < Y < s vặi s = 32124/32768 ~ = .9803. Thặ sặủ nhặủp MU thặủng nhặủn ặặủ bặủg cặch đặủg fread(...,'uchar') ặặỏ àoüc cặc tấủ ám thanh cặi byte ặặủ mặi hoặi

"Cặc giáỉ trặ rặỏn" lặ cặc sặủ nguyặ củi MATLAB - cặc sặủ thặủ cặi giáỉ trặ nguyặ

LIN2MU

Chuyãỏn tặủủn tên sang mu-law

MU = LIN2MU(Y) chuyển các biến đầu tên hiều âm thanh tuyến tính trong miền $-1 \leq Y \leq 1$ sang "các giá trị rỗng" mà lại theo luật mu-law trong miền $0 \leq MU \leq 255$

Các hàm âm học biến

BESSELJ

BESSELJ Các hàm Bessel loại 1

$J = \text{BESSELJ}(\text{ALPHA}, X)$ tên hàm Bessel loại 1, $J_\alpha(X)$ với giá trị thực, bước khang âm ALPHA và đầu số X. Kết quả có $\text{size}(J) = \text{size}(Z)$ nếu ALPHA và hằng, hoặc $\text{size}(J) = [\text{prod}(\text{size}(Z)), \text{length}(\text{ALPHA})]$ nếu ALPHA là vectơ. Các phần tử của X có thể là các giá trị thực khang âm bước bất kỳ. Tuy nhiên, với ALPHA, có 2 hàm chuẩn quan trọng: ALPHA phải tăng dần về, nghĩa là $\text{ALPHA} = \text{alpha}:1:\text{alpha}+n-1$, và các giá trị phải thoả mãn $0 \leq \text{alpha}(k) \leq 1000$.

Về đầu: `besselj(3:9, (10:.2:20)')` phát sinh bảng 51x7 ở trang 400 trong sách "Handbook of Mathematical Functions" của Abramowitz và Stegun, `besselj(2/3:1:98/3, r)` phát sinh các hàm Bessel bước phân nhỏ dùng bởi MathWorks Logo, mình dùng L. $J_{2/3}(r)$ phù hợp âm kỳ để tải góc trong với góc bằng $\pi/(2/3)$

BESSELY

BESSELY Các hàm Bessel loại 2

$Y = \text{BESSELY}(\text{ALPHA}, X)$ tên các hàm Bessel loại 2, $Y_\alpha(X)$ với giá trị thực, bước khang âm ALPHA và đầu số X. Kết quả có $\text{size}(Y) = \text{size}(Z)$ nếu ALPHA và hằng, hoặc $\text{size}(Y) = [\text{prod}(\text{size}(Z)), \text{length}(\text{ALPHA})]$ nếu ALPHA là vectơ. Các phần tử của X có thể là giá trị thực khang âm bước bất kỳ. Tuy nhiên, với ALPHA, có 2 hàm chuẩn quan trọng: ALPHA phải tăng dần về, nghĩa là $\text{ALPHA} = \text{alpha}:1:\text{alpha}+n-1$, và các giá trị phải thoả mãn $0 \leq \text{alpha}(k) \leq 1000$. Về đầu:

`bessely(3:9, (10:.2:20)')` phát sinh bảng 51x7 ở trang 401 trong sách "Handbook of Mathematical Functions" của Abramowitz và Stegun

BESSELI

BESSELI Sửa âm các hàm Bessel loại 1

$I = \text{BESSELI}(\text{ALPHA}, X)$ tên hàm sửa âm Bessel loại 1, $I_\alpha(X)$ với giá trị thực, bước khang âm ALPHA và đầu số X. Kết quả có $\text{size}(I) = \text{size}(Z)$ nếu ALPHA và hằng, hoặc $\text{size}(I) = [\text{prod}(\text{size}(Z)), \text{length}(\text{ALPHA})]$ nếu ALPHA là vectơ. Các phần tử của X có thể là các giá trị thực khang âm bước bất kỳ. Tuy nhiên, với ALPHA, có 2 hàm chuẩn quan trọng: ALPHA phải tăng

tầng âm vẽ, ghéa lại ALPHA = alpha:1:alpha+n-1, vài các giá trị phải thoả mãn $0 \leq \alpha(j) \leq 1000$
 E = BESSELI(ALPHA,X,1) tên $I_\alpha(X) * e^{-X}$, $I_\alpha(X) * \exp(-X)$. Quan hầu giá các hàm Bessel loại 1 lại $I_\alpha(X) = i^{-\alpha} * J_\alpha(X)$, $I_\alpha(X) = i^{(-\alpha)} * J_\alpha(X)$

BESSELK

BESSELK Sắp xếp các hàm Bessel loại 2

K = BESSELK(ALPHA,X) tên các hàm Bessel loại 2, $K_\alpha(X)$ với giá trị thực, bước khang ám ALPHA vài số X. Kết quả có size(K) = size(Z) nếu ALPHA và hằng, hoặc size(K) = [prod(size(Z)), length(ALPHA)] nếu ALPHA là vectơ. Các phần tử của X có thể là giá trị thực khang ám bước bắt kỳ. Tuy nhiên, với ALPHA, có 2 hàm chú quan trọng: ALPHA phải tầng tầng âm vẽ, ghéa lại ALPHA = alpha:1:alpha+n-1, vài các giá trị phải thoả mãn $0 \leq \alpha(j) \leq 1000$

E = BESSELK(ALPHA,X,1) tên $K_\alpha(X) * \exp(X)$. Quan hầu với các hàm Bessel nguyên bán có số số a: $K_\alpha(x) = \pi/2 * i^{-\alpha} * (J_\alpha(i*x) + Y_\alpha(i*x))$

Về du:

besselk(3:9,[0:.2:9.8 10:.5:20],1) phát sinh bảng 71x7 ở trang 424 trong sách "Handbook of Mathematical Functions" của Abramowitz và Stegun

BETA

Hàm Beta

y = beta(z,w)

y = tích phân từ 0 đến 1 của $t^{(z-1)} * (1-t)^{(w-1)} dt$. z và w phải phù hợp

Hàm beta chưa hoàn thành

y = beta(x,a,b). Các phần tử của x phải trong khoảng [0,1]. a và b phải là các số dương và hằng

BETAINC

BETAINC Hàm beta chưa hoàn thành.

Y = BETAINC(X,A,B). Các phần tử của X phải trong khoảng [0,1]. Các số X, A và B rút cái phải cùng kích thước, ngoài trừ hàm có các số và hằng như các ma trận hàng của kích thước chung của các số khác

BETALN

Là-ga-rit của hàm beta

y = betaln(z,w). z và w phải phù hợp

ELLIPJ

Các hàm elliptic Jacobi SN, CN và DN

[Sn,Cn,Dn] = ELLIPJ(U,M) trả về các giá trị của các hàm elliptic Jacobi SN, CN và DN, các số U và tham số M. Như bộ song hiều thì, M giải từ $0 < M < 1$

ELLIPJ(U,M) là tích phân chẵn xác định EPS. U và M phải là các ma trận cùng cỡ hoặc một trong hai là số thực và hằng. Không cho các tham số K và M - chú ý quan hệ theo cách sau: $M = K^2$

ELLIPKE

ELLIPKE Tính phân elliptic đầy đủ

$[K,E] = \text{ELLIPKE}(M)$ trả về giá trị của tích phân elliptic đầy đủ loại 1 và 2 trên M. Nếu bỏ sung hai tham số, M giải ra $0 < M < 1$

Đầy chẵn xác định của ELLIPKE(M) là EPS. Không cho các tham số K và M - chú ý quan hệ theo cách sau: $M = K^2$

ERF

Hàm sai số

$$y = \text{erf}(x)$$

$y = 2/\sqrt{\pi}$ nhân với tích phân từ 0 đến x của $\exp(-t^2)$ dt

ERFC

Hàm sai số bù

$$y = \text{erfc}(x)$$

$y = 2/\sqrt{\pi}$ nhân với tích phân từ 0 đến x của $\exp(-t^2)$ dt
 $= 1 - \text{erf}(x)$

ERFCX

Hàm sai số bù có chia từ lâu

$$y = \text{erfcx}(x)$$

$y = \exp(x^2) * \text{erfc}(x) \sim (1/\sqrt{\pi}) * 1/x$ với x lớn

ERFINV

Hàm ngược của hàm sai số

$x = \text{erfinv}(y)$ thỏa mãn $y = \text{erf}(x)$, $-1 \leq y < 1$, $-\infty \leq x \leq \infty$

EXPINT

Hàm tích phân mũ, $E_1(x)$

$Y = \text{EXPINT}(X)$ là tích phân từ X đến ∞ của $(\exp(-t)/t)$ dt

GAMMA

Hàm gamma

$Y = \text{GAMMA}(X)$ xác định hàm gamma tại các giá trị của X. X phải thực

$\text{gamma}(x) = \int_0^\infty \exp(-t) t^{x-1} dt$

$$\text{gamma}(n+1) = n! = n \text{ giai thừa} = \text{prod}(1:n)$$

GCD

Ước chung lớn nhất

$G = \text{GCD}(A,B)$ là ước chung lớn nhất của các số nguyên A và B

$G = \text{GCD}(0,0) = 0$ theo quy ước; tất cả các GCD khác đều nguyên dương

$[G,C,D] = \text{GCD}(A,B)$ cũng trả về C và D với $G = A*C + B*D$

GAMMAINC

GAMMAINC Hàm gamma chẵn hoàn thành

$Y = \text{GAMMAINC}(X,A)$ ước lượng hàm gamma chẵn hoàn thành tại tất cả các phần tử của X . X phải thực. X và A phải cùng kích thước, ngoài trừ hàm coi các ảnh số và hằng như các ma trận hàng cùng kích thước với các ảnh số khác

$\text{gammainc}(x,a) = (\text{tích phân tại } 0 \text{ đến } x \text{ của } t^{(a-1)} \exp(-t) dt) / \text{gamma}(a)$

Lưu ý: $\text{gammainc}(x,a)$ tiến về 1 khi x tiến về vô cùng

LCM

Bước chung nhỏ nhất

$\text{LCM}(A,B)$ là bước chung nhỏ nhất của các số nguyên dương A và B

LEGENDRE

LEGENDRE Các hàm Legendre liên tiếp

$P = \text{LEGENDRE}(N,X)$ tên các hàm Legendre liên tiếp bậc N và theo tham số $M = 0, 1, \dots, N$, tên trên X . N là một số nguyên nhỏ hơn 257. X là vectơ coi các phần tử $X(j)$ thực thỏa mãn $\text{abs}(X(j)) \leq 1$. Giao trẻ xuất P là ma trận cỡ $(N+1) \times L$, với $L = \text{length}(X)$. $P(i,j)$ ứng với hàm Legendre liên tiếp bậc N và về trục $(i-1)$, tên tại $X(j)$

Ảnh nhúng toạ độ của hàm P là $P(n,m;x) = (-1)^m * (1-x^2)^{(m/2)} * (d/dx)^m P(n,x)$, với $P(n,x)$ là ảnh thực Legendre bậc n

Lưu ý: dòng đầu của P là ảnh thực Legendre ước lượng tại X . (trường hợp $M = 0$)

$SP = \text{LEGENDRE}(N,X,'sch')$ tên các hàm Schmidt quan hệ biến chuẩn tắc với các hàm Legendre là $SP(n,m;x)$. Các hàm này quan hệ với các hàm Legendre liên tiếp bằng chuẩn $P(n,m;x)$ bởi: $SP(n,m;x) = \sqrt{2*(n-m)!/(n+m)!} * P(n,m;x)$

Về đầu:

`legendre(2, 0:0.1:0.2)` trả về ma trận dưới

$P(2,0;0)$	$P(2,0;0.1)$	$P(2,0;0.2)$
$P(2,1;0)$	$P(2,1;0.1)$	$P(2,1;0.2)$
$P(2,2;0)$	$P(2,2;0.1)$	$P(2,2;0.2)$

GAMMALN

GAMMALN Hàm lô-ga-rit gamma

$Y = \text{GAMMALN}(X)$ là $\text{LOG}(\text{GAMMA}(X))$, trình bày dưới dạng logarit tự nhiên

LOG2

Taích các số thực. Theo chuẩn IEEE lài logb()

$y = \text{LOG2}(x)$ tênh là-ga-rit cả số 2 của x

$[f,e] = \text{LOG2}(x)$ vãi ma trấu thực x , trái vãi ma trấu thực, thằing trong miãon $0.5 \leq \text{abs}(f) < 1$, vãi ma trấu e các số nguyên, ãõ $x = f \cdot 2.^e$. Mũi số 0 trong x cho ra $f = 0$ vãi $e = 0$. Cách này phù hẫp vãi hàm ANSI C lài frexp() vãi hàm thực chuẩn IEEE lài logb()

POW2

Mũi số thực. Theo chuẩn IEEE lài scalbn()

$x = \text{pow2}(y)$ tênh 2 y

$x = \text{pow2}(f,e)$ vãi ma trấu thực f vãi ma trấu nguyên e thç tênh $x = f \cdot (2.^e)$. Kĩt quai tênh toain nhanh bằing cách ãn giáin lài cũng e vãi phẫon mũi của f . Cách này phù hẫp vãi hàm ANSI C lài ldexp() vãi hàm thực chuẩn IEEE lài scalbn()

RAT

Xáúp xè phán số

$[N,D] = \text{RAT}(X,\text{tol})$ trái vãi 2 ma trấu nguyên ãõ $N./D$ gẫon bằing X , trong trằing hẫp này $\text{abs}(N./D - X) \leq \text{tol} \cdot \text{abs}(X)$. Các giái trẽ xáúp xè phán số ãõ phẫt sinh bằing cách cẫt liẫ tuic các khai triẫon phẫn thẫp phán. Ngẫm ãẽnh thç dung sai $\text{tol} = 1.e-6 \cdot \text{norm}(X(:),1)$. $\text{RAT}(X)$ hoẫc $\text{RAT}(X,\text{tol})$ hiẫon thẽ liẫ tuic các bẫic biẫu diẫon. Cũing thuẫt toain, vãi giái trẽ ngẫm ãẽnh của dung sai tol , ãõ ãẽng bẫn trong MATLAB cho lẫnh FORMAT RAT

RATS

Xuất các phán số

$\text{RATS}(X,\text{LENS})$ dùng RAT ãõ hiẫon thẽ các giái trẽ xáúp xè phán số của các phẫn tặ của X . ãõ dài của mẫi phẫn tặ lài LENS. Ngẫm ãẽnh thç LENS = 13, cho pheĩ 6 phẫn tặ trong 78 vẽ trẽ.

Dấu sao (*) ãõ dùng cho các phẫn tặ khẫg thẫ in ra trong khẫg gian phán bẫ, nhẽng so sẫnh khẫ ãẽng kẫ vãi các phẫn tặ trong X . Cũing thuẫt toain, vãi giái trẽ ngẫm ãẽnh của LENS, ãõ ãẽng bẫn trong MATLAB cho lẫnh FORMAT RAT

CART2SPH

CART2SPH Biẫon ãõ hẫ toĩa ãõ ãõ-các sang hẫ toĩa ãõ cáõ

$[AZ,EL,R] = \text{CART2SPH}(X,Y,Z)$ Biẫon ãõ dẫ liẫ lẫ trong hẫ toĩa ãõ ãõ-các sang hẫ toĩa ãõ cáõ.

Nẫ $[M,N] = \text{SIZE}(X)$, thç Y vãi Z cũng phẫi cũng kễ thẫic. AZ vãi EL tênh theo ãn vẽ radian

CART2POL

CART2POL Biẫon ãõ hẫ toĩa ãõ ãõ-các sang hẫ toĩa ãõ cẫc

$[TH,R] = \text{CART2POL}(X,Y)$ Biẫon ãõ dẫ liẫ lẫ trong hẫ toĩa ãõ ãõ-các sang hẫ toĩa ãõ cẫc

[M,N] = SIZE(X), thç Y cũng phải cùng kích thước. TH tên theo ẩn về radian

[TH,R,Z] = CART2POL(X,Y,Z) Biếu ãi ãi liâu læ trong hâu toà ãi ãi sang hâu toà ãi tru.

Nếu [M,N] = SIZE(X), thç Y và Z phải cùng kích thước

POL2CART

POL2CART Biếu ãi hâu toà ãi cæ sang hâu toà ãi ãi ãi

[X,Y] = POL2CART(TH,R) Biếu ãi ãi liâu læ trong hâu toà ãi cæ sang hâu toà ãi ãi ãi. Nếu [M,N] = SIZE(TH), thç R cũng phải cùng kích thước. TH phải theo ẩn về radian

. [X,Y,Z] = POL2CART(TH,R,Z) Biếu ãi ãi liâu læ trong hâu toà ãi tru sang hâu toà ãi ãi ãi. Nếu [M,N] = SIZE(TH), thç R và Z phải cùng kích thước

SPH2CART

SPH2CART Biếu ãi hâu toà ãi cæ sang hâu toà ãi ãi ãi

[X,Y,Z] = SPH2CART(AZ,EL,R) Biếu ãi ãi liâu læ trong hâu toà ãi cæ sang hâu toà ãi ãi ãi. Nếu [M,N] = SIZE(AZ), thç EL và R cũng phải cùng kích thước. AZ và EL phải theo ẩn về radian

Máí vai ãiing tãp

FOPEN

Máí tãp

FID = FOPEN('filename',permission) máí tãp coi tãn chë ãi lai filename vãi chã ãi cho phei chë ãi lai permission. Permission lai mãt trong cã chuã:

'r' _	ãi
'w' _	ghi (tao mãi nếu cã)
'a' _	nãi thãm vào cuã tãp (tao mãi nếu cã)
'r+' _	ãi và ghi (khã tao mãi)
'w+' _	cã tãp hoã tao mãi ãi ãi ãi và ghi
'a+' _	ãi và ghi thãm vào cuã tãp (tao mãi nếu cã)
'W'	ghi mãi khã tã ãi xõ vùng ãi
'A'	nãi thãm mãi khã tã ãi xõ vùng ãi

Ngã ãi cã tãp ãi mãi trong chã ãi nhë phán. ãi mãi tãp và bán thç thãm 't' vào chuã permission, nhã 'rt' và 'wt+'

FID = FOPEN(' filename') giái sã permission lai 'r'. Nếu mãi thã cã thç ãi hiã FID lã mãt ãi læ và hã ãi ãi MATLAB, ãi hiã tãp ãi ãi nhã ãi sã ãi tiã cho cã phu vù vào ra tãp. Nếu viã mã tãp khã thã cã thç

traí vảo -1 cho FID. Ba danh hiếuu tấp âæüc tæu
 âæung dùng mai khăng cãon mải. Chuĩng lài fid=0
 (nhấp chuáon), fid=1 (xuát chuáon), vai fid=2
 (thiấut bẻ lầui chuáon)

[FID, MESSAGE] = FOPEN(' filename ',permission) traí
 vảo mẩt lầui hẩu thẩng nẩu mải thẩt baui.

FOPEN('all') traí vảo mẩt vectả ðoĩng, cãic danh hiếuu
 cho tẩt cãi cãic tấp âang mải bảii ngæĩi dùng.
 (Nhæng khæng phẩi 0, 1, vai 2)

[FILENAME, PERMISSION] = FOPEN(FID) traí vảo tãn tấp
 vai chẩu âẩu vẩi danh hiếuu tấp âaĩ cho. Nẩu tấp
 âæüc mải trong chẩu âẩu 'r' vai khæng tặm thẩy tấp
 trong thæ muĩc lầim viẩc thç FOPEN tặm xuẩng âæĩng
 ðầun tặm kiẩm cuĩa MATLAB

[FID, MESSAGE] = FOPEN(' filename ',permission,
 machineformat) mải tấp chẻ âẻnh vẩi chẩu âẩu chẻ âẻnh
 vai âoĩc ðẩi liẩu bầong cãich dùng FREAD hoặc ghi
 ðẩi liẩu bầong cãich dùng FWRITE vẩi FORMAT âaĩ cho
 bảii machineformat. machineformat lài mẩt trong cãic
 chuầui sau:

'native' hoặc 'n' - FORMAT mẩy cuĩc bẩ- ngặm
 âẻnh

'ieee-le' hoặc 'l' - chuáon IEEE

'ieee-be' hoặc 'b' - chuáon IEEE

'vaxd' hoặc 'd' - chuáon VAX D

'vaxg' hoặc 'g' - chuáon VAX G

'cray' hoặc 'c' - chuáon Cray

'ieee-le.164' hoặc 'a' - chuáon IEEE vai âẩu ðẩi
 sẩu liẩu 64 bit

[FILENAME, PERMISSION, MACHINEFORMAT] = FOPEN(FID)
 traí vảo tãn tấp, chẩu âẩu mải, vai ðầung mẩy vẩi
 ðầnh hiếuu tấp. Chẩu âẩu 'W' vai 'A' âæüc thiấut
 kẩ âẩ ðùng vẩi cãic âẻa tẻi vai khæng tæu âæung
 xoĩa vùng âẩm hiẩu thẩi sau cãic thao tẩc xuẩt.
 Vẻ ðũ, mải 1/4" bầng tẻi trẩn traũm SPARC âẩ ðĩ ghi mải
 khæng tæu âæung xoĩa vùng âẩm: fid =
 fopen('/dev/rst0', 'W')

FCLOSE

Âoĩng tấp

FCLOSE(FID) âoĩng tấp vẩi danh hiếuu FID, lài mẩt
 sẩu nguyãn nhẩn âæüc tẻi FOPEN() trẩic âỷ

FCLOSE('all') âoĩng tẩt cãi cãi tấp âang mải trẻi 0,
 1 vai 2

FCLOSE() traí vảo s 0 nẩu thầnh cầng, -1 nẩu ngặc
 lầui

Vào/Ra tệp khăng daung thæic**FREAD**

Âoüc tấup nhê phán

[A, COUNT] = FREAD(FID,SIZE,PRECISION) âoüc đăi liău nhê phán tãi tấup chề âênh vai ghi vào ma trấun A. Âăúi său xuăút tuiy choùn COUNT trái văo său phăon tãi âoüc thănh căng. FID lài danh hiău nguyên của tấup âăüc măi tãi FOPEN. Âăúi său tuiy choùn SIZE; nău khăng coi thç âoüc toăn bău tấup; nău coi thç căic âăo mưc hăup lyi lài:

N âoüc N phăon tãi vào măt vectă căt.

inf âoüc vào cuăi tấup.

[M,N] âoüc căic phăon tãi âăo lăup âăy ma trấun câi MxN, theo thăi tăt căt

N coi thăo lài inf, nhăng khăng lài M
Âăúi său PRECISION âiăo khiăon său bit âoüc cho măt giai trê vai thăng dề căic bit năy nhă căic giai trê kyi tăt, său nguyên hoăc său thăc. Căic chuăi sau coi thăo duiy, hoăc căic phiăn băn MATLAB, hoăc C hoăc Fortran tăng âăng. Nău khăng chề âênh thç âăüc ngăom âênh lài 'uchar'. Căic phăon tãi của ma trấun kăt quăi lăn lău trong đăung său thăc đăi của

	MATLAB	C hoăc Fortran	Mă tãi
	'char'	'char'	kyi tăt ,
8 bit			
	'schar'	'signed char'	kyi tăt coi
đău, 8 bit			
	'short'	'short'	său
nguyên, 16 bit			
	'int'	'int'	său
nguyên, 16 hoăc 32 bit			
	'long'	'long'	său
nguyên, 32 hoăc 64 bit*			
	'float'	'float'	său thăc
, 32 bit			
	'double'	'double'	său thăc
đăi , 64 bit			
	'uchar'	'unsigned char'	kyi tăt
khăng đău, 8 bit			
	'ushort'	'unsigned short'	său nguyên
khăng đău, 16 bit			
	'uint'	'unsigned int'	său nguyên
khăng đău, 16 hoăc 32 bit			
	'ulong'	'unsigned long'	său nguyên
khăng đău, 32 bit			
	'char'	'char*1'	kyi tăt , 8
bit			
	'float32'	'real*4'	său thăc
32 bit			
	'float64'	'real*8'	său
thăc 64 bit			
	'int8'	'integer*1'	său nguyên, 8
bit			

'int16'	'integer*2'	sáu	nguyên,
16 bit			
'int32'	'integer*4'	sáu	nguyên,
32 bit			
	'integer*8'	sáu	nguyên, 64
bits**			
'intN'		sáu	
nguyên coi dấu, N bit rỗng			
'uintN'		sáu	
nguyên không dấu, N bit rỗng			

N biến lưu giá trị từ bất kỳ giá trị 1 vài 32

* 64 bits trên DEC alpha

** chế độ trên DEC alpha

Về đầu,

```
fid = fopen('FREAD.m','r');
```

```
F = FREAD(fid);
```

```
s = setstr(F')
```

mỗi tập chứa số mục HELP này, rồi cho vài giá trị
thể toàn bộ tập, gồm nên dùng size = inf vài
precision = 'uchar'. Kết quả length(F) là số ký
tên trong tập

[A, COUNT] = FREAD(FID,SIZE,PRECISION,SKIP) đưa vào
đầu số tùy chọn SKIP cho nên số byte bỏ qua
sau mỗi lần đọc. Cách này thường dùng để trích
đặt liệu trong các trường không liên tục tại các
bản ghi coi đầu dài câu nên

FWRITE

Ghi đặt liệu vào một tập nhớ phân

COUNT = FWRITE(FID,A,PRECISION) ghi các phần tử của
ma trận A vào tập chế nên, dịch các giá trị
MATLAB sang đầu nên xác chế nên lại precision. Đặt
liệu nên ghi theo thứ tự cuối. COUNT là số phần
tử nên ghi thành công. FID là danh liệu số
nguyên của tập nhớ tại FOPEN, hoặc 1 chỉ xuất
chuẩn hoặc 2 cho lần chuẩn. PRECISION là đầu khi
đang vài kích thước của kết quả. Xem danh sách
PRECISION cho phép trong FREAD

COUNT = FWRITE(FID,A,PRECISION,SKIP) đưa vào đầu
số tùy chọn SKIP cho nên số byte bỏ qua
trước mỗi lần ghi. Cách này thường dùng để
chèn đặt liệu trong các trường không liên tục tại
các bản ghi coi đầu dài câu nên

Về đầu

```
fid = fopen('magic5.bin','wb')
```

```
FWRITE(fid,magic(5),'integer *4')
```

Tạo mỗi một tập nhớ phân 100-byte, chứa 25 phần
tử của ma phương bậc 5, lưu các số nguyên 4-
byte

Nhập/xuất tập coi đang thực

FSCANF

Cho đặt liệu coi đang thực trong một tập

[A,COUNT] = FSCANF(FID,FORMAT,SIZE) àoüc đêi liău tậi tấu chề àềnh bắi danh hiău tấu FID, chuyăon sang chuăi chề àềnh FORMAT, vậi trậi vắo mậ trậu A. COUNT lậi ắu sắu xuắut tậi choũn ắo trậi vắo sắu phắon tậi àoüc thậnh cắng

FID lậi mắut sắu nguyăn danh hiău tấu nhắu ắoüc tậi FOPEN. SIZE lậi tậi choũn; nối ắu mắut giắi hậu vậo sắu phắon tậi cắi thắo ắoüc àoüc tậi tấu; nắu khắng chề àềnh thắ xem nhắ toậi bắ tấu; nắu cắ chề àềnh thắ cắ ắo mắ hắ lyi lậ:

N àoüc hắu hắ N phắon tậi vậo mắ vectắ cắ

inf àoüc ắu hắ tấu
[M,N] àoüc M * N phắon tậi lắp ắo ắ nhắ mắ mậ trậu cắ MxN, theo thắ tắ cắ. N cắ thắ lậ inf, nhắ khắng lậ M. Nắu mậ trậu trậi vắ A tậ cắ chề đượ chuyăon ắo kyi tắ vậ SIZE khắng thuắ đượ [M,N] thắ trậi vắ mắ vectắ cắ. FORMAT lậ mắ chuăi chắ cắ chề àềnh chuyăon ắo cúi ngắ nắ C. Cắ chề àềnh chuyăon ắo gắ kyi tắ %, lắ choũn "*" bắ qua vắ gắ vậ trắ ắ rắ, vậ cắ kyi tắ chuyăon kiắ d, i, o, u, x, e, f, g, s, c, vậ [. . .]. Xem gắ trắ C vắ cắ kyi tắ chuyăon kiắ. Nắu mắ kyi tắ chuyăon ắo ắoüc đượ ắo àoüc mắ phắon tậi thắ cắ thắ lậ cho mắ sắ phắon tậi mậ trậu cúi MATLAB ắoüc đượ, mắ phắon tậi gắ mắ kyi tắ. Vắ trắ kyi tắ vậ cắ chề àềnh chuyăon kiắ sắ sei lậ cho mậ trậu kắ quắ lậ mậ trậu sắ vậ cắ kyi tắ àoüc vậ nhắ gắ trắ ASCII cúi chắ, mắ kyi tắ trắ mắ phắon tậi cúi mậ trậu

FSCANF kắ vắ hắ trượ tắ cúi nố trong C, ắo quan trượ tắ ắng lậ nố ắoüc "vectắ hắ" nhằo trắi vắ mắ ắu sắ mậ trậu. Chuăi FORMAT ắoüc đượ lắ qua tấu cho ắu khi hắ tấu hắ tắ sắ đắ liău chề àềnh bắi SIZE ắoüc àoüc vậ. Vế đượ:

S = fscanf(fid,'%s') àoüc (vậ trậi vắ) mắ chuăi kyi tắ

A = fscanf(fid,'%5d') àoüc sắ nguyăn thắ phắ 5-chắ sắ

FPRINTF

FPRINTF Ghi đắ liău cắ đượ thắ vậ tấu
COUNT = FPRINTF(FID,FORMAT,A,...) àềnh đượ đắ liău trong mậ trậu A (vậ trong mắ mậ trậu ắu sắ thắ vậ), đắi ắo khiắo cúi chuăi chề àềnh FORMAT, vậ ghi nố vậ tấu cắ đượ hiău FID. COUNT lậ ắu sắ xuắut tậi choũn ắo trậi vắ sắ byte ghi thậnh cắ. FID lậ mắ sắ nguyăn danh hiău tấu nhắ ắoüc FOPEN. Nố cắ cắ thắ lậ 1 cho xuắut chắo (mắ hắ) hắo 2 cho lắ chắo. FORMAT lậ chuăi chắ cắ chề àềnh chuyăon kiắ cúi C. Cắ chề àềnh chuyăon ắo gắ kyi tắ %, , cắ cắ tậ choũn, cắ trắ ắo rắ vậ sắ cắ sắ thắ phắ, chề àềnh kiắ con, vậ cắ kyi tắ chuyăon

kiểu d, i, o, u, x, X, f, e, E, g, G, c, và s. Xem
giáo trình C để biết thêm chi tiết
FPRINTF như ANSI C với các loại trừ và các mã
ràng. Bao gồm:

1. Nếu số thực kép của MATLAB không chuyển
chênh lệch sang kiểu để lưu trữ ở trong ký tự
chuyển kiểu thực FORMAT "e" để dùng. Phải
chuyển rồi rằng các giá trị không nguyên của
MATLAB sang các giá trị nguyên nếu có ý để dùng
để ở chuyển kiểu nguyên như "d" và lấy cách
xử lý của ANSI C

2. Các chế độ kiểu con không chuẩn sau để
cung cấp cho các ký tự chuyển kiểu o, u, x, và X

t - Kiểu để lưu trữ cả số của C là float
khi với số nguyên không dấu

b - Kiểu để lưu trữ cả số của C là double
khi với số nguyên không dấu

Về đây, để in ra một giá trị double cả số hex thực
dùng FORMAT như '%bx'

FPRINTF cũng khi với hàm cùng tên của nó trong
C - quan trọng là " vectơ hoặ " khi A không phải
là một mảng và hằng. Chuỗi FORMAT để dùng
lưu qua các phần tử của A (từng cột) cho đến
khi tất cả các phần tử để dùng hết. Rồi nó
để dùng lưu theo cùng cách, không khi nào
lưu, qua một vài số mà trừn thêm vào. Về đây,
các lệnh

```
x = 0:1:1; y = [x; exp(x)];
fid = fopen('exp.txt','w');
fprintf(fid,'%6.2f %12.8f\n',y);
```

tao một một tập và bán chéo bán ngược các
giá trị của hàm mũ cả số tự nhiên:

```
0.00    1.00000000
0.10    1.10517092
...
1.00    2.71828183
```

FGETL

Trái với dòng tiếp theo của tập như một chuỗi
FGETL(FID) trái với dòng tiếp theo của một tập
với danh hiệu tập FID như một chuỗi MATLAB. Trái
với mà không để vào ký tự sang dòng. Dùng
FGETS() để lấy dòng tiếp theo có để vào ký tự
sang dòng. Nếu đang kết thúc tập thực trái với -1

QUAN TRỌNG: Hay lưu ý rằng hàm này có ý
để dùng với tập và bán. Nếu bán nhóm lùn,
để một tập "nhỏ phần" mà không có ký tự sang
dòng thực phức vụ này có thể tốn nhiều thời
gian với tập lùn. Về đây: (tạo ảnh lùn 'type
fgetl.m')

```
fid=fopen('fgetl.m');
while 1
    line = fgetl(fid);
    if ~isstr(line), break, end
    disp(line)
```

```
end
fclose(fid);
```

FGETS

Traí vảo dòng tiáúp theo cuía tấúp nhæ mắút chuầúi

FGETS(FID) traí vảo dòng tiáúp theo cuía mắút tấúp vắii danh hiấu tấúp fid lài mắút chuầúi MATLAB. Coi áæa vaio kyỉ tæu sang dòng. Dùng FGETL() áãø láúy dòng tiáúp theo mai khăng áæa vaio kyỉ tæu sang dòng. Nắu áuỉng kắút thuíc tấúp thç traí vảo -1

QUAN TROÚNG: Háỷ læu yỉ ràòng hàim nắy coi yỉ chè dùng vắii tấúp vàn báín. Nắu báíi nhắóm láùn, áoúc mắút tấúp "nhẽ phán" mai khăng coi kyỉ tæu sang dòng thç phuíc vuú nắy coi thắø táún nhiấu thắi gian vắii tấúp láín

Về trê tấúp

FERROR

Hoíi cáic láúi trong nhắúp/xuáút tấúp

MESSAGE = FERROR(FID, 'clear') traí vảo thắng báío láúi vaio/ra gắon nhắút vắii tấúp chè áềnh. Chuầúi 'clear' lài tuỷ choún. Nắu coi, thç xoía láúi biấu hiấu cho tấúp chè áềnh

[MESSAGE, ERRNUM] = FERROR(FID) cuỉng traí vảo sắ láúi. FID lài mắút sắ nguyắ danh hiấu tấúp nhắu áæúc tắi FOPEN, hoàúc 0 cho nhắúp chuáon, 1 cho xuáút chuáon hoàúc 2 cho láúi chuáon

FEOF

Kiắøm tra háút tấúp

FEOF(FID) kiắøm tra biấu hiấu háút tấúp cho tấúp vắii danh hiấu tấúp lài FID

FEOF(FID) traí vảo 1 nắu biấu hiấu háút tấúp áæúc áầút, hoàúc 0 nắu khăng áầút

FSEEK

Áãøi biấu hiấu về trê trong mắút tấúp

STATUS = FSEEK(FID, offset, origin) áầút láúi về trê tấúp trong tấúp chè áềnh vaio sắ byte offset táng áểng vắii origin chè áềnh. FID lài mắút sắ nguyắ danh hiấu tấúp nhắu áæúc tắi FOPEN. Cáic giái trê offset áæúc thắng dềch nhæ sau:

> 0 : Di chuyắon vảo phêa cuắúi tấúp

= 0 : Khăng thay áãøi về trê

< 0 : Di chuyắon vảo phêa áầu tấúp

Cáic giái trê origin áæúc thắng dềch nhæ sau:

'bof' hoàúc -1 : Áầu tấúp

'cof' hoàúc 0 : Về trê hiấu thắi trong tấúp

'eof' hoàúc 1 : Cuắúi tấúp

STATUS = 0 nắu thắnh cắng vắi -1 nắu bẻ láúi

Về duú:

fseek(fid,0,-1) "tua láúi" tấúp (áầút về trê láúi ái áầu tấúp)

FTELL

Tìm về trê trong măt tấu

POSITION = FTELL(FID) traí vấ về trê cuía trong tấu chè aãnh. Position biấu hiấu theo aân về byte tênh tài aấu tấu. Nấu -1 aấu traí vấ, thç biấu hiấu khăng thành cắg; ðùng FERROR aấu xấc aãnh trấu thấi lầi. FID lài măt sấu nguyă danh hiấu tấu nhấu aấu tài FOPEN, hoàu 0 cho nhấu chuáon, 1 cho xuấu chuáon hoàu 2 cho lầi chuáon

FREWIND

FREWIND tua laùi mǎüt tǎüp âang mǎi

FREWIND(FID) âăüt con troí tǎüp vaio âăöu tǎüp vǎii danh hiǎüu tǎüp lai fid

Khuyǎün caío: Tua laùi mǎüt danh hiǎüu tǎüp fid vǎii mǎüt thiǎüt bē bàng tǎi coí thǎø khǎng khǎng laim viǎüc, mǎüc ðui khǎng phaít sinh lǎüi!

Chuyǎøn âăöi chuǎüi**SPRINTF**

SPRINTF Ghi ðăi liǎüu coí ðaúng thǎic vaio mǎüt chuǎüi

[S,ERRMSG] = SPRINTF(FORMAT,A,...) âēnh ðaúng ðăi liǎüu trong ma tráün A (vài trong caíc âăúi sǎu ma tráün thǎm vaio), ðăăi âiǎöu khiǎøn cuía chuǎüi chè âēnh FORMAT, vài traí vǎo ñoi trong biǎün S vǎii ðaúng chuǎüi MATLAB. ERRMSG lai âăúi sǎu xuát tuý choün âăø traí vǎo mǎüt chuǎüi thǎng baío lǎüi nǎüu xǎy ra lǎüi hoăüc mǎüt ma tráün rǎüng nǎüu khǎng lǎüi. SPRINTF giǎúng nhê FPRINTF ngoăüi trǎi ñoi traí vǎo ðăi liǎüu trong mǎüt biǎün chuǎüi MATLAB khaiç vǎii ghi vaio tǎüp. FORMAT lai mǎüt chuǎüi chǎia caíc chè âēnh chuyǎøn kiǎöu cuía C. Caíc chè âēnh chuyǎøn kiǎöu gǎöm kyí tǎü %, caíc cǎi, caíc trǎëng âăü rǎüng vài sǎü chǎi sǎü thǎüp phǎn, chè âēnh kiǎöu con, vài kyí tǎü chuyǎøn kiǎöu d, i, o, u, x, X, f, e, E, g, G, c, vài s. Xem giaío trçnh C âăø biǎüt thǎm chi tiǎüt. SPRINTF nhê trong ANSI C vǎii moüt sǎü ngoăüi lǎüi vài mǎi rǎüng. Bao gǎöm:

1. Nǎüu sǎü thǎüc keíp cuía MATLAB khǎng chuyǎøn chēnh xǎic sang kiǎöu ðăi liǎüu chè âēnh trong kyí tǎü chuyǎøn kiǎöu thç FORMAT "e" âăăüc ðùng. Baün phaíi chuyǎøn roí rǎüng caíc giai trē khǎng nguyǎn cuía MATLAB sang caíc giai trē nguyǎn nǎüu baün coí yí âēnh ðùng chè âēnh chuyǎøn kiǎöu nguyǎn nhê "d" vài lǎüy caích xǎi lý cuía ANSI C

2. Caíc chè âēnh kiǎöu con khǎng chuǎøn sau âăăüc cung cáúp cho caíc kyí tǎü chuyǎøn kiǎöu o, u, x, vài X

t - Kiǎöu ðăi liǎüu cả sǎi cuía C lai float khaiç vǎii sǎü nguyǎn khǎng ðǎü

b - Kiǎöu ðăi liǎüu cả sǎi cuía C lai double khaiç vǎii sǎü nguyǎn khǎng ðǎü

Vē ðuü, âăø in ra mǎüt giai trē double cả sǎü hex thç ðùng FORMAT nhê '%bx'

SPRINTF cuíng khaiç vǎii hàm cuíng tǎn cuía ñoi trong C - quan troúng lai " vectǎ hoĩa " khi A khǎng phaíi lai âăüi lǎëng vǎ hǎëng. Chuǎüi FORMAT âăăüc ðùng lǎüi qua caíc phǎön tǎi cuía A (tǎng cǎüt) cho âăün khi táüt cái caíc phǎön tǎi âăăüc ðùng hǎüt. Rǎöi ñoi âăăüc ðùng lǎüi theo cuíng caích, khǎng khǎi taüo lǎüi, qua moüi âăúi sǎü ma tráün thǎm vaio. Vē ðuü, caíc lǎünh

S = sprintf('rho is %5.3f',(1+sqrt(5))/2) cho ra chuǎüi S = 'rho is 1.618'

SSCANF

Âoüc chuăùi dæãii âiăöu khiăøn coĩ daùng thăic

[A,COUNT,ERRMSG,NEXTINDEX] = SSCANF(S,FORMAT,SIZE)
 âoüc dæi liăuu tæi biăún chuăùi MATLAB lài S, chuyăøn sang chuăùi chè âênh FORMAT, vài trái văö trong ma tráun A. COUNT lài măt âăúi său xuát tuỳ choün âăö trái văö său pháön tæ âoüc thành cäng
 ERRMSG lài măt âăúi său xuát tuỳ choün âăö trái văö măt chuăùi thăng baïo lăui năuu coĩ lăui hoăuc măt ma tráun răung năuu khăng lăui. NEXTINDEX lài măt âăúi său xuát tuỳ choün chè âênh măt hay nhiăöu său kyĩ tæu âaĩ queĩt trong S. SSCANF giăúng FSCANF ngoăui trăi noĩ âoüc dæi liăuu tæi măt biăún chuăùi MATLAB khaĩc văĩi âoüc tæi măt tăúp. SIZE lài tuỳ choün; noĩ âăut măt giăĩi haun vaio pháön tæ coĩ thăö queĩt tæi chuăùi; năuu khăng chè âênh, thç xem nhæ toain bău chuăùi; năuu coĩ chè âênh, thç caĩc âăö muúc hăúp lyĩ lài:

N âoüc N pháön tæi vaio măt vectă căut
 inf âoüc âăún cuăúi chuăùi
 [M,N] âoüc M * N pháön tæi lăúp âăöy êt nhăt măt ma tráun căi MxN, theo căut. N coĩ thăö lài inf, nhăng khăng lài M. Năuu ma tráun kăut quai A chè tæi viăúc ðuĩng kyĩ tæu chuyăøn kiăöu vài SIZE khăng coĩ daùng [M,N] thç măt vectă ðoĩng âæăuc trái văö. FORMAT lài măt chuăùi chăia caĩc chè âênh chuyăøn âăöi cuia ngăn ngăi C. Caĩc chè âênh chuyăøn âăöi găöm kyĩ tæu %, lăua choün " * " boĩ qua viăúc gaĩn vài trăăng âău răung, vài caĩc kyĩ tæu chuyăøn kiăöu d, i, o, u, x, e, f, g, s, c, vài [. . .]. Xem giaĩo trçnh C văö caĩc kyĩ tæu chuyăøn kiăöu. Năuu măt kyĩ tæu chuyăøn âăöi âæăuc ðuĩng âăö âoüc măt pháön tæi thç coĩ thăö làm cho măt său pháön tæi ma tráun cuia MATLAB âæăuc ðuĩng, măt pháön tæi giăi măt kyĩ tæu . Viăúc trăun kyĩ tæu vài caĩc chè âênh chuyăøn kiăöu său seĩ làm cho ma tráun kăut quai lài ma tráun său vài caĩc kyĩ tæu âoüc vaio nhæ giaĩ trẽ ASCII cuia chuĩng, măt kyĩ tæu trăn măt pháön tæi cuia ma tráun
 Viăúc queĩt âăún cuăúi chuăùi xaiy ra khi NEXTINDEX lăin hăn kch thăic cuia S

SSCANF khaĩc văĩi hàm trũng tăn cuia noĩ trong C, âiăöu quan troũng tăng æĩng lài noĩ âæăuc "vectă hoĩa" nhăom trăĩ văö măt âăúi său ma tráun. Chuăùi FORMAT âæăuc ðuĩng lăui qua tăúp cho âăún khi hăut tăúp hoăuc tăong său dæi liăuu chè âênh băĩ SIZE âæăuc chuyăøn sang. Vê ðu, caĩc lăũnh:

```
S = '2.7183 3.1416';
```

```
A = sscanf(S,'%f')
```

taũo măt măt vectă hai pháön tæi chăia giaĩ trẽ xăúp xè cuia său e vài pi

Caĩc ma tráun thæa să căúp**SPEYE**

Ma tráun thæa âăn vë

SPEYE(M,N) tạo ma trận thưa cỡ MxN với các
số 1 trên đường chéo chính
SPEYE(N) là cách gọi của SPEYE(N,N)
SPEYE(SIZE(A)) là ma trận giá trị bằng 1
SPARSE(EYE(SIZE(A)))

SPRANDN

Caic ma trầun thæa ngầu nhiãn

$R = \text{SPRANDN}(S)$ coi cùng cáúu trưíc thæa nhæ S , nhæng caic phầon tấi âæåüc phán bầu ngầu nhiãn chuáon

$R = \text{SPRANDN}(m,n,\text{density})$ lài ma trầun thæa câi $m \times n$, vắi caic phầon tấi khaic khăng âæåüc phán bầu chuáon khoaíng $\text{density} \cdot m \cdot n$

$R = \text{SPRANDN}(m,n,\text{density},rc)$ cùng coi sắu âiắou kiắun tắng hầu gắon bàong rc . R âæåüc xáy đắng tắi tắong caic ma trầun hầng bàong 1. Nắu rc lài mắut vectắ áú dài $lr \leq \min(m,n)$, thç R coi lr giaỉ trề kỳi đễ cuía noỉ lài rc , táút cái caic phầon tấi khaic bàong 0. Trong trắng hắp nắy, R âæåüc phaỉt sinh bàong caích quay phầong ngầu nhiãn áip đắng cho mắut ma trầun cheío vắi caic giaỉ trề kỳi đễ cho trắic. Noỉ coi caích phán phắi lắn cáúu trưíc topo vắi áú sắ

SPRANDSYM

SPRANDSYM Caic ma trầun thæa ngầu nhiãn áú xắng

$R = \text{SPRANDSYM}(S)$ lài mắut ma trầun thæa ngầu nhiãn áú xắng coi tam giaic đắi vắ áúng cheío cùng cáoi trưíc nhæ S . Caic phầon tấi âæåüc phán bầu chuáon, vắi trung bçnh 0 vắ phắng sai 1

$R = \text{SPRANDSYM}(n,\text{density})$ lài mắut ma trầun vắng thæa áú xắng ngầu nhiãn cáúp n vắi caic phầon tấi khaic khăng khoaíng $\text{density} \cdot n \cdot n$; mắi phầon tấi lài tắong cuía mắut hoằc nhiắou caic mắu phán bầu ngầu nhiãn chuáon

$R = \text{SPRANDSYM}(n,\text{density},rcond)$ cùng coi sắu âiắou kiắun tắng hầu bàong $rcond$. Viắc phán bầu caic phầon tấi lài khăng áúng bắ; noỉ lài áú xắng thắ qua 0; táút cái trong áoắn $[-1,1]$. Nắu $rcond$ lài 1 vectắ áú dài n , thç R coi caic giaỉ trề riắng lài $rcond$. Do áoi, nắu $rcond$ lài mắut vectắ đắng (khăng á) thç R seỉ xắc áềnh đắng (khăng á). Trong trắng hắp nắy, R âæåüc phaỉt sinh bàong caích quay Jacobi áip đắng voía ma trầun cheío vắi caic giaỉ trề riắng hoằc sắ áú kiắun cho trắic. Noỉ coi caích phán phắi lắn cáúu trưíc topo vắi áú sắ

$R = \text{SPRANDSYM}(n, \text{density}, rcond, kind)$ lài mắut xắc áềnh đắng. Nắu $kind = 1$ thç R âæåüc phaỉt sinh bàong pheíp quay mắut ma trầun cheío xắc áềnh đắng. R coi áúng sắ áú kiắun mong muắ. Nắu $kind = 2$ thç R lài tắong đềch chuyắon caic tềch ngoắi lài. R chề coi sắ áú kiắun mong muắ nhæng cáúu trưíc noỉ hắ. Nắu $R = \text{SPRANDSYM}(S, \text{density}, rcond, 3)$, thç R coi cùng cáúu trưíc nhæ ma trầun S vắ sắ áú kiắun xắp xề $1/rcond$, áú sắ density âæåüc boỉ qua

SPDIAGS

SPDIAGS Trềch ra vắ táu mắi caic ma trầun dài bàng vắ ma trầun cheío thæa SPDIAGS, lài tắong quắt hoía hằg gắon liắon "diag", xắ lý 3 loắi ma trầun, caic tắ hắp khaic nhau, cùng nhæ nhắp vắ xắút. A lài ma trầun câi $m \times n$, thắng thæa (nhæng khăng cáon thiắút), vắi caic phầon

tại khai kháng, hoặc chèn thêm, các phần tại đầu
trên p hàng chéo

B là ma trận $m(n)$ hàng và p cột, mảng d (những hàng cần thêm), coi các cột là các
hàng chéo của A. d là vectơ đầu dài p coi các
thành phần nguyên chèn thêm các hàng chéo trong A

Một cách khác, A, B và d quan hệ bởi

```
for k = 1:p
    B(:,k) = diag(A,d(k))
end
```

Bán thao tác khác nhau, phân biệt theo sáu ảnh
số như sau, coi thao dùng với SPDIAGS là:

Lấy ra tất cả các hàng chéo khác nhau:

```
[B,d] = spdiags(A);
```

Lấy ra hàng chéo chèn thêm:

```
B = spdiags(A,d);
```

Thay các hàng chéo chèn thêm:

```
A = spdiags(B,d,A);
```

Tạo một ma trận theo các hàng chéo của
nó:

```
A = spdiags(B,d,m,n);
```

Quan hệ chèn chéo trong A, B và d là:

```
if m >= n
    for k = 1:p
        for j = max(1,1+d(k)):min(n,m+d(k))
            B(j,k) = A(j-d(k),j);
        end
    end
else
    for k = 1:p
        for i = max(1,1-d(k)):min(m,n-d(k))
            B(i,k) = A(i,i+d(k));
        end
    end
end
```

Với phần tại của B, cũng với các vẽ trên "bên
ngoài" A, hàng chéo thêm ảnh hưởng bằng các vòng lặp
này. Chúng khác ảnh hưởng tham chiếu khi B là ảnh
số như vậy ảnh hưởng đầu giá trị 0 khi B là ảnh số
xuất.

Về đầu, các lệnh này phải sinh một biểu thức 3
hàng chéo theo của phép vi phân cấp 2 cỡ ảnh
tới n ảnh

```
e = ones(n,1);
```

```
A = spdiags([e -2*e e], -1:1, n, n)
```

Lệnh này ảnh ma trận A thành ma trận kiểm tra
Wilkinson (xem WILKINSON(n))

```
A = spdiags(abs(-(n-1)/2:(n-1)/2)', 0, A)
```

Cuối cùng, lệnh này sẽ là ảnh 3 hàng chéo

```
B = spdiags(A)
```

Về đầu hai hàng vuông

```
A = [ 11    0    13    0
      0    22    0    24
      0     0    33    0
     41     0     0    44
```

Ngãüc laûi, vãi B và d nhæ trăn thç biãøu thăic
 $\text{spdiag}(B, d, 7, 4)$ cho laûi A vãi giaĩ trẽ găuc

Chuyãøn ma tráun ááöy áuí thàình ma tráun thæa

SPARSE

Xáy dæung ma tráuun thæa tæi caic sǎu khaic 0 vai caic chè sǎu

S = SPARSE(...) là hàm gồn liãön phaït sinh caïc ma trãün thuäüc lãip læu trãêi thëa cuía MATLAB. Hàm naøy coï thãø goüi vãi 1, 2, 3, 5 hoäuc 6 ääúi säu

$S = \text{SPARSE}(X)$ chuyển ma trận thưa hoàn toàn về ma trận thưa dạng đường chéo với các phần tử 0

S = SPARSE(i,j,s,m,n,nzmax) dùng các dòng của [i,j,s] để sinh ma trận thưa cỡ mxn với hàng gian phân bố cho nzmax phần tử khác 0. Hai vectơ chèn sáu nguyên, i và j, và vectơ thuộc hoặc khác, s, tất cả có cùng chiều dài, nnz, là sáu phần tử khác 0 trong ma trận thưa kết quả. Có một số cách gọi ẩn giá trị 6 nếu cần

$S = \text{SPARSE}(i, j, s, m, n)$ dùng $\text{nzmax} = \text{length}(s)$.

$S = \text{SPARSE}(i, j, s)$ dùng $m = \max(i)$ và $n = \max(j)$.

S = SPARSE(m,n) caïch tàõt cuía
 SPARSE([],[],[],m,n,0). Lăûnh này phait sinh ma trấûn
 thăa cả baïn câi mxn găôm táút cái cáic sấu 0. Ấûi
 sấu s vai mấût trong hai ấûi sấu i vai j coĩ thăo lại
 ấûi lăăung vă hăăing, trong trăăing hăûp này chấing
 ấăăuc bung ra ấăø cái 3 ấăúi sấu ấăou coĩ cườg ấău
 dài. Vê duu, trêch ra vai sau ăoĩ răip lăui mấût ma
 trấûn thăa:

```
[i,j,s] = find(S);
```

```
[m,n] = size(S);
```

```
S = sparse(i,j,s,m,n);
```

Bảii váuỷ, nhæ caỉc lấunh nàii, nấuu đoiing vại cầut
cuầui cườg coỉ caỉc phầon tắi khaỉc 0:

```
[i,j,s] = find(S);
```

```
S = sparse(i,j,s);
```

Táút cái caic thao taic gǎon liǎon cuía MATLAB vǎo sǎu hoüc, logic vai vǎo chè sǎu ǎǎou coi thǎo ǎip ðuùng cho caic ma tráun thǎa, hoàuc cho caic ma tráun vǎia thǎa vǎia ǎǎoy ǎuí. Caic thao taic trǎn ma tráun thǎa traí vǎo ma tráun thǎa, ǎǎoy ǎuí traí vǎo ǎǎoy ǎuí. Trong háou hǎut cái trǎǎng hǎup thç cái thao taic trǎn cái ma tráun vǎia thǎa vǎia ǎǎoy ǎuí traí vǎo ǎǎoy ǎuí. Ngoaúi trǎi trǎǎng hǎup coi ǎǎa vaio, kǎut

quái của các thao tác trên thao với array của coi cấu trúc thao, nhẽ lại A.* S ết nhất thao nhẽ S. Mũt sũ thao tác, nhẽ S >= 0, phải sinh các ma trận thao lần ("Big Sparse", hay "BS") -- ma trận thuậ tậ chặc lậ trậ thao nhẽ ết phậ tậ 0

FULL

Chuyậ ma trận thao sang lậ lậ trậ array của

A = FULL(X) lài hàm gậ liậ chuyậ mũt ma trận thao sang cả cấu lậ trậ array của vài cho ra mũt ma trận array của khẽ thay ậ

FIND

Tậ các chẽ sũ của các phậ tậ khặ 0

I = FIND(X) trậ vậ chẽ sũ của các phậ tậ khặ 0 của vectơ X

Vê du, I = FIND(A>100) , trậ vậ các chẽ sũ của các phậ tậ lần hữ 100 của A. Xem RELOP

[I,J] = FIND(X) trậ vậ các chẽ sũ đòg vài cậ của các phậ tậ khặ 0 trong ma trận X. Lậ nhậ ñậ thậ ñùng với các ma trận thao

[I,J,V] = FIND(X) cữg trậ vậ mũt vectơ cậ gậ các phậ tậ khặ 0 của X. Lậ yậ ñòg find(X) vài find(X~=0) sẽ cho ra cữg I vài J, nhẽ hàm sau cho V với tậ cái các sũ 1

SPCONVERT

SPCONVERT Chuyậ ñậg thao mậ ñậg

S = SPCONVERT(D) chuyậ mũt mậg (nnz hoặ nnz+1)x (3 hoặ 4) với các đòg chặ [i,j,s] hoặ [i,j,real(s(i,j)),imag(s(i,j))] sang ma trận thao tậg ẽg. Mũt đòg bậ kậ trong D coi ñậg [m n 0] hoặ [m n 0 0] coi thậ ñùng ậ chẽ ậ ñậ kậ thậ của S. Nậ D ậ thậ ñậ thậ khẽ chuyậ, vậ vậ SPCONVERT coi thậ ñùng sau khi D ậ ñậ ñậ tậ tậ MAT hoặ tậ ASCII

Làm viãnũ vẫi cũc phẫn tặi khaĩc 0 cũa ma trấũn thũa

NNZ

Sẫũ phẫn tặi khaĩc 0

`nz = NNZ(S)` lài sẫũ phẫn tặi khaĩc 0 trong S

NONZEROS

NONZEROS Cũc phẫn tặi khaĩc 0

NONZEROS(S) lài mẫũ vectỏ cũũt ấẫỹ ẫũĩ cũc phẫn tặi khaĩc 0 trong S. Hẫm nẫỹ cho s, nhẽng khẽng cho i vẫi j nhẽ `[i,j,s] = find(S)`

NZMAX

Tẫng sẫũ khẽng gian lẫũ trấi cũc phẫn tặi khaĩc 0

Ầẫũ vẫi ma trấũn thũa, NZMAX(S) lài sẫũ vẽ trẽ lẫũ trấi phẫn bẫũ cho cũc phẫn tặi khaĩc 0 trong S. Vẫi ma trấũn ẫẫỹ ẫũĩ, NZMAX(S) lài `prod(size(S))`. Trong cũ hai trẫĩng hẫũp, `nnz(S) <= nzmax(S) <= prod(size(S))`

SPONES

Thay thẫũ cũc phẫn tặi khaĩc 0 vẫi cũc sẫũ 1

`R = SPONES(S)` phẫũ sinh mẫũ ma trấũn vẫi cũũ trũĩ thũa gẫĩũg S, nhẽng vẫi cũc sẫũ 1 tẫũ vẽ trẽ khaĩc 0

SPALLOC

SPALLOC Phẫn bẫũ bẫũ nhẫĩ cho cũc phẫn tặi khaĩc 0

`s = SPALLOC(m,n,nzmax)` tẫũ mẫũ ma trấũn thũa cũĩ m_{xn} tẫũ cũ cũc phẫn tặi 0 vẫi cũũ trẫũg ẫẫũ lẫũ nzmax phẫn tặi khaĩc 0. Vẽ đũũ:

```
s = spalloc(n,n,3*n);
    for j = 1:n
        s(:,j) = (mẫũ vectỏ cũũt thũa gẫĩũ 3 phẫn
tặi khaĩc 0);
    end
```

ISSPARSE

ISSPARSE Ầẫĩng nẫũũ ma trấũn lài ma trấũn thũa

ISSPARSE(S) bầũg 1 nẫũũ lẫũp lẫũ trấi cũĩ S lài thũa vẫi ngẫũc lẫũĩ thũc bầũg 0

SPFUN

Ầẫũ đũũg mẫũ hẫm cũũ cho cũc phẫn tặi khaĩc 0

`F = SPFUN('fun',S)` ẫẫĩ lẫũũg hẫm `fun(s)` trẫũ cũc phẫn tặi khaĩc 0 cũĩ S. Nghĩa lài

`F = SPFUN('exp',S)` cũũ cũũg mẫũũ thũa nhẽ S (trẫĩ trẫĩng hẫũp trẫũ đẫĩĩ), ngẫũc lẫũĩ EXP(S) cũũ gẫĩ trẽ 1 tẫũ nẫĩ S cũũ gẫĩ trẽ 0

Xem cũc ma trấũn thũa

SPY

Trẫũ quan hũa cũũ trũĩ thũa

SPY(S) vẽ màu thà cuía ma tráuñ S báút kyì
 SPY(S,color) dùng màu à ãnh dáu chề à ãnh thay cho màu vaing
 SPY(S,marksize) dùng kèch thã ãic à ãnh dáu chề à ãnh thay cho kèch thã ãic phuû thuãüc vaio kèch thã ãic hñh à ãnh vai kèch thã ãic ma tráuñ
 SPY(S,color,marksize) vai SPY(S,marksize,color) lai cho pheip

GPLOT

Veì à ã ò thề theo "lyì thuyãút à ã ò thề"

GPLOT(A,xy) veì à ã ò thề chề à ãnh báíi A vai xy. Mãüt à ã ò thề, G, gãom mãütì táúp caic nuít (à ãnh) à ã ãüc à ãnh sãú tãi 1 à ã ãn n, vai mãüt táúp caic cung, hoãuc caũnh, nãúi caic à ãnh laúi. À ã ò veì à ã ò thề G, cáon 2 ma tráuñ. Ma tráuñ kã ò, A, coì a(i,j) khaic 0 nãúi vai chề nãúi à ãnh i à ã ãüc nãúi vã ãi à ãnh j. Mãing toĩa à ã ã, xy, lai mãüt ma tráuñ cá ã nx2 vã ãi về trề cho à ãnh i taúi doing thã ã i xy(i,:) = [x(i) y(i)]

GPLOT(A,xy,lc) dùng kiã òu à ã ãing vai màu thay cho ngãom à ãnh lai 'r-'. Vê duũ, lc = 'g:'. Xem PLOT

[X,Y] = GPLOT(A,xy) traí vã ò caic vectã à ãnh dáu NaN X vai Y mai khãng veì. Caic vectã nãy coì thã ò dùng à ã ò phãit sinh à ã ò thề sau nãy nãúi muãun

Caic thuãút toã ãn sã òp xãúp laúi

COLMMD

Bãuc táúi thiã òu vã ò cáüt

p = COLMMD(S). Bãuc táúi thiã òu vã ò cáüt cuía S. À ã ãi vã ãi mãüt ma tráuñ khãng à ã ãi xãing S, haím nãy traí vã ò mãüt cáüt hoã ãn về p à ã ò S(:,p) coì caic nhãn tãí LU thã hãñ S

SYMMMD

Bãuc à ã ãi xãing táúi thiã òu

p = SYMMMD(S), à ã ãi vã ãi mãüt ma tráuñ xãic à ãnh đã ãng à ã ãi xãing S thç traí vã ò mãüt hoã ãn về à ã ò S(p,p) coì nhãn tãí Cholesky thã hãñ S. À ãi khi SYMMMD cuing laim viãüc tãút à ã ãi vã ãi caic ma tráuñ à ã ãi xãing khãng xãic à ãnh

SYMRCM

À ãio thã ã tãu Cuthill-McKee

p = SYMRCM(S) traí vã ò mãüt hoã ãn về p à ã ò S(p,p) coì profile nhoí hãñ S. Haím nãy sã òp trã ãic tãút cho viãüc taich nhãn tãí LU hoãuc Cholesky à ã ãi vã ãi caic ma tráuñ nhãñn à ã ãüc tãí caic bài toã ãn dài vai gã òy ("long, skinny"). Noí laim viãüc cho cá ã hai loaúi S à ã ãi xãing vai khãng à ã ãi xãing

COLPERM

COLPERM Bãc caic cáüt đẽu vaio caich à ã ãm caic phãon tãí khaic 0

`rank(A)`, vai trong sáu hoặc `sprank(A) == rank(sprandn(A))` với sáu hoặc xaíc suát

Caic thao taic trăn cáy

TREELAYOUT

`TREELAYOUT` Trộn bày mặt cây hoặc rặng
`[x,y,h,s] = treelayout(parent,post)` với `parent` là vectơ các nút cha, với 0 cho gốc. `post` là hoán về hâu tậ trăn các nút của cây (nếu bỏ qua `post` tên nội á ááy). `x` và `y` là các vectơ gồm các toạ độ trong hình vuông ảnh về mặt á áôi trộn bày các nút của cây á áø ta ra mặt hình ảnh á áêp mặt. Các á áúi sáu tùy cho, `h` là á áúi cao của cây và `s` là sáu nút con của nút gốc

TREEPLOT

`TREEPLOT` Vẽ hình ảnh của mặt cây
`TREEPLOT(p,c,d)` `p` là vectơ các á áøm cha, với `p(i) == 0` cho gốc. `c` là màu và ký tậ cho các nút, hoặc `c=''` á áø không vẽ nút. `d` là màu và ký tậ cho các cành, hoặc `d=''` á áø không vẽ cành. `c` hoặc `d` có thể bỏ qua, và các giá trị trẻ gồm á ánh hữp lý á áúi sáu đư

ETREE

Cây khẩ của mặt ma trậ
`p = etree(A)` trái vẽ mặt cây khẩ á áúi với mặt ma trậ vuông á áúi xẽ co tam giá trăn là tam giá trăn của `A`. `p(j)` là cha của cút `j` trăn cây, hoặc 0 nếu `j` là gốc
`p = etree(A,'col')` trái vẽ cây khẩ của `A'*A`
`p = etree(A,'sym')` giá `p = etree(A)`
`[p,q] = etree(...)` cũng trái vẽ hoán về hâu tậ
`q` trăn `cáy`

ETREEPLOT

`ETREEPLOT` Vẽ mặt cây khẩ
`etreeplot(A):` Vẽ cây khẩ của `A` (hoặc `A+A'`, nếu `A` không á áúi xẽ)
`etreeplot(A,c,d):` Xem `treelayout` cho các tham sáu tùy cho `c` và `d`

Caic thao taic linh tinh

SYMBFACT

SYMBFACT Phán têch nhán tãi Symbol

count = SYMBFACT(A) trái vảo vectả gầom sấu âầum ðoing cuía nhán tãi tam giaic trăn nhán tãi Cholesky cuía mầut ma trấun âầui xẩng coỉ tam giaic trăn lại tam giaic trăn cuía A, giai sải boỉ qua trong tiẩun trỡnh phán têch thầi sấu. Phuỷc vuỷ nầy nhanh hẵn chol(A)

count = SYMBFACT(A,'col') phán têch A'*A (mài khẩng taồ taồ noỉ mầut caỉch roỉ rang)

count = SYMBFACT(A,'sym') giẩng nhặ p = symbfact(A). Coỉ mầut sấu giai trẽ trái vảo tuỷ choỷn:

[count,h,parent,post,R] = symbfact(...) cẩng trái vảo âầu cao cuía cắy khẩi, chểnh cắy khẩi, mầut hoẩn vể cuía cắy khẩi, vại ma trấun 0-1 R coỉ cắu truỷc cuía chol(A)

SPPARMS

SPPARMS Âầut caic tham sấu cho caic phuỷc vuỷ ma trấun thầ

SPPARMS('key',value) âầut mầut hoầc nhiẩu tham sấu "tunable" ðuỷng cho caic toẩn tãi phẩng trỡnh tuyẩn tểnh thầ, lại \ vại /, vại caic sắp xẩp báuc âầui xẩng tẩi thiẩu, COLMMD vại SYMMMD

SPPARMS, in ra mã tãi caic cài âầut hiẩn thẩi.

Nẩu khẩng coỉ âầui sấu nhẩp thặ values = SPPARMS trái vảo mầut vectả coỉ caic thẩnh phẩn cho trong caic cài âầut hiẩn thẩi

[keys,values] = SPPARMS trái vảo vectả âoỉ, vại cẩng trái vảo mầut ma trấun kyỉ tẩ coỉ caic ðoing lại caic tãi khoỉa cho caic tham sấu

SPPARMS(values), khẩng coỉ âầui sấu xuẩt, âầut tẩt cái caic tham sấu vại caic giai trẽ chề âểnh báỉ vectả âầui sấu

value = SPPARMS('key') trái vảo cài âầut hiẩn thẩi cuía mầut tham sấu

SPPARMS('default') âầut tẩt cái caic tham sấu vảo caic cài âầut mầc âểnh cuía noỉ

SPPARMS('tight') âầut caic tham sấu báuc tẩi thiẩu vảo caic cài âầut chàut cheỉ cuía chuỷng, chuỷng coỉ thẩo âầun âầun caic báuc êt hẵn báuc thay thẩ, nhểng làm cho caic hẩm thẩc hiẩn nhiẩu láon hẵn. Caic tham sấu vẩi caic giai trẽ ngắom âểnh vại caic giai trẽ chàut cheỉ lại:

	tãi khoỉa	
ngắom âểnh	chầut	
values(1)	'spumoni'	0
values(2)	'thr_rel'	1.1
1.0		
values(3)	'thr_abs'	1.0
0.0		
values(4)	'exact_d'	0
1		

values(5)	'supernd'	3
1		
values(6)	'rreduce'	3
1		
values(7)	'wh_frac'	0.5
0.5		
values(8)	'autommd'	1
values(9)	'aug_rel'	0.001
values(10)	'aug_abs'	0

Yĩ nghĩa các tham số lài:

spumoni: Xuát chấon àoain các âiãöu khiãon Sparse Monitor Flag; 0 cho khăng coi, 1 cho mấut vài, 2 cho quãi nhiãöu

thr_rel, thr_abs: Ngẽãĩng báuc táúi thiãöu lài thr_rel*mindegree + thr_abs

exact_d: Khaĩc 0 ããø dùng các báuc chẽnh xaĩc trong báuc táúi thiãöu, 0 dùng cho các báuc xáúp xẽ

supernd: If > 0, MMD các siãu nuĩt hãun hãúp mầui giãĩ trẽ supernd

rreduce: If > 0, MMD thu goũn doĩng mầui giãĩ trẽ rreduce

wh_frac: Các doĩng vãiĩ density > wh_frac ããũc boĩ qua trong COLMMD

autommd: Khaĩc 0 ããø dùng báuc táúi thiãöu vãiĩ \\ vài /

aug_rel, aug_abs: Tham số tẽ lãũ thầũng dẽ cho các phãĩng trẽnh gia sốũ lài aug_rel*max(max(abs(A))) + aug_abs. Vẽ duũ, aug_rel = 0, aug_abs = 1 ãũt mầut ma trãũn ãĩn vẽ khăng chia tẽ lãũ vaĩo khãũĩ (1,1) củĩa ma trãũn gia sốũ

SPAUGMENT

SPAUGMENT Trẽnh bàiĩ bàiĩ toain bẽnh phãĩng táúi thiãöu nhẽ mầut hãũ tuyãũn tẽnh lãĩn

S = SPAUGMENT(A,c) tãũ ma trãũn báút ãẽnh ãũĩ xẽĩng, thẽa, vưỡĩng S = [c*I A; A' 0].Ma trãũn nãĩy liãĩn quan vãiĩ bàiĩ toain bẽnh phãĩng táúi thiãöu min norm(b - A*x) bãĩĩ

r = b - A*x

S * [r/c; x] = [b; 0].

Giaĩ trẽ táúi ẽu củĩa nhãn tãĩ tẽ lãũ thầũng dẽ c, bao gũĩm min(svd(A)) vãi norm(r), thẽĩĩng quãi tãũn nhiãöu pẽĩp tẽnh

S = SPAUGMENT(A), khăng chẽ ãẽnh giaĩ trẽ củĩa c, dùng giaĩ trẽ mầuc ãẽnh củĩa SPPARMS, thẽĩĩng lài max(max(abs(A)))/1000. Ma trãũn gia sốũ ããũc dùng tãũ ãũũĩng bãĩĩ vĩaũc giaĩĩ phãĩng trẽnh tuyãũn tẽnh, \\ vài /, cho các bàiĩ toain khăng vưỡĩng

SYMBOLIC TOOLBOX**ALLVALUES**

Tìm tất cả các giá trị của biểu thức RootOf
 ALLVALUES(S), với S là một biểu thức symbolic
 hoặc một vectơ cột chứa biểu thức con
 'RootOf(EXPR)', tìm nghiệm của EXPR rồi tên S. Kết
 quả là một vectơ chứa tất cả các giá trị có
 ảnh của S
 Ví dụ: $p = 'x^5 + x^4 + 2';$ $s =$
`solve(p); allvalues(s)`

AR2SM

Chuyển ma trận Maple sang ma trận symbolic
 A = AR2SM(M) ảnh các dòng của MATRIX([[...],[...]])
 hoặc VECTOR([...]) cho ra bảng các hàm ảnh số
 tương ứng tên của Maple sang ma trận symbolic

CHARPOLY

Ảnh các ảnh trong symbolic
 CHARPOLY(A) tên ảnh các ảnh trong của ma trận A.
 Kết quả là một ảnh các
 symbolic theo biến 'x'
 CHARPOLY(A,'v') dùng 'v' thay cho 'x'
 Trật tự số làm tròn, charpoly(A) như
`poly2sym(poly(A))` và `poly(A)` như
`sym2poly(charpoly(A))`
 Ví dụ: `charpoly(gallery(3))`

COLLECT

Tập hợp các hằng số
 COLLECT(S) xem ảnh phần tử của S như một ảnh các
 biến symbol của S. Nếu biến symbol của S là 'x',
 thì COLLECT(S) tập hợp tất cả các hằng số với
 cùng mũ của 'x'
 COLLECT(S,'v') lấy 'v' làm biến symbolic trong
 ảnh phần tử của S

COMPOSE

Hàm tếp
 COMPOSE(f,g), với f và g là các biểu thức
 symbolic hoặc hàm, gọi là f(y) và g(x),
 thì trái vế một biểu thức symbolic hoặc hàm
 f(g(x))
 COMPOSE(f,g,'u') dùng biến 'u' cho cả f và g.
 f và g có dạng sau:
 $f = f(u, a_1, a_2, \dots); g = g(u, b_1, b_2, \dots)$
 COMPOSE(f,g,'u','v') dùng biến 'u' cho f và
 'v' cho g. f và g có dạng sau:
 $f = f(u, a_1, a_2, \dots); g = g(v, b_1, b_2, \dots)$
 Ví dụ: `compose 1/(1+x^2) sin(x) là`
`1/(1+sin(x)^2)`

COMSTACK

Sắp xếp các dấu phẩy trong mặt nạ toán symbolic
 $A = \text{comstack}(A)$ chèn các khoảng trống trong ma trận symbolic A để rút gọn các dấu phẩy của nó
 tách dòng

COSINT

Hàm tích phân cosin

$\text{COSINT}(x) = \gamma + \ln(i \cdot x) - i \cdot \pi/2 + \int_0^x (\cos(t) - 1)/t, dt$

DETERM

Định thức ma trận symbolic

$\text{DETERM}(A)$ tính định thức symbolic của ma trận A, với A là mặt nạ toán symbolic hoặc ma trận số

$\text{DETERM}(\text{VPA}(A))$ dùng chính xác số học
 Ví dụ: $\text{determ}(\text{sym}(5,5, '1/(i+j-t)'))$

DIGITS

Đầu số chữ số của Maple

Đầu chính xác các phép tính số của Maple
 xác định bài các chữ số

Chức năng DIGITS hiển thị số chữ số hiển thị

$\text{DIGITS}(D)$ Đầu số chữ số của D cho các phép tính tiếp sau

$D = \text{DIGITS}$ trả về số chữ số hiển thị

DSOLVE

Giải hệ phương trình vi phân thường symbolic

$\text{DSOLVE}('eqn1', 'eqn2', \dots)$ cho phép nhập đầu vào 12 đầu số nhập, là các phương trình symbolic biểu diễn các phương trình vi phân thường và các điều kiện đầu. Các phương trình hoặc các điều kiện đầu có thể nhúng nhau dấu phẩy trong mặt nạ nhập. Biểu thức lặp thường dùng 'x'. Có thể đặt biến thành 't' bằng cách dùng 'x' là biểu thức phụ thuộc, hay dùng 't' thay cho 'x' là biểu thức độc lập trong mặt nạ phương trình. Biểu thức lặp có thể thay đổi tại 'x' sang ký tự thường nào đó bằng cách đặt vào ký tự thường đó như mặt nạ đầu số cuối cùng. Ký tự 'D' biểu diễn vi phân từng phần bằng biểu thức lặp, nghĩa là thường dùng d/dx . Mặt chữ D có mặt chữ số kèm theo biểu diễn vi phân lặp, nghĩa là D^2 là d^2/dx^2 . Các ký tự bắt đầu theo sau ngay các phép toán vi phân này để biểu diễn phụ thuộc, nghĩa là D^3y biểu diễn đạo hàm bậc ba của $y(x)$ hoặc $y(t)$. Các điều kiện đầu hoặc điều kiện biên các phương trình như ' $y(a)=b$ ' hoặc ' $Dy(a) = b$ ', ở đây y là mặt biểu thức phụ thuộc vào a, b là các hằng. Nếu số các điều kiện đầu hoặc điều kiện biên phụ thuộc các hằng tùy ý C_1, C_2, \dots Nếu đầu số xuất hiện mặt biểu thức phụ thuộc theo DSOLVE trả về giá trị trái vế mặt danh sách các lời giải

$[Y1, Y2, \dots] = \text{DSOLVE}(\dots)$ trả về các nghiệm, theo thứ tự alphabet, trong các đầu số xuất.

Vài hầu phi tuyến, nếu nghiêm ngặt duy nhất thì các giải trẻ xuất coi thảo lại các vectơ symbolic

Về đầu:

```

Phải trình vi phân bậc 1
dsolve('Dy = a*y')
dsolve('Df = f + sin(t)')
y = dsolve('(Dy)^2 + y^2 = 1','s')
Phải trình vi phân bậc 1 với 1 điều kiện
đầu
dsolve('Dy = a*y', 'y(0) = b')
dsolve('Df = f + sin(t)', 'f(pi/2) = 0')
y = dsolve('(Dy)^2 + y^2 = 1', 'y(0) = 0',
's')
Phải trình vi phân bậc 2
dsolve('D2y = -a^2*y')
Phải trình vi phân bậc 2 với các điều kiện
biên
dsolve('D2y = -a^2*y', 'y(0) = 1, Dy(pi/a)
= 0')
Hai phải trình vi phân bậc 1
[x,y] = dsolve('Dx = y', 'Dy = -x')
[f,g] = dsolve('Df = 3*f+4*g', 'Dg = -
4*f+3*g')
Hai phải trình vi phân bậc 1 với các điều
kiện đầu
[x,y] = dsolve('Dx = y', 'Dy = -x',
'x(0)=0', 'y(0)=1')
[f,g] = dsolve('Df = 3*f+4*g, Dg = -
4*f+3*g', 'f(0)=0, g(0)=1')
Ba phải trình vi phân bậc 1 với các điều
kiện đầu
[u,v,w] = dsolve('Du=v, Dv=w, Dw=-
u', 'u(0)=0, v(0)=0, w(0)=1')
Cùng bài toán lại một phải trình vi phân
bậc 3
w = dsolve('D3w = -w', 'w(0)=1, Dw(0)=0,
D2w(0)=0')

```

EIGENSYS

Ma trận riêng và vectơ riêng symbolic

EIGENSYS(A) tên các giải trẻ riêng symbolic của ma trận A

EIGENSYS(VPA(A)) tên các giải trẻ riêng sáu bằng cách dùng sáu hoặc

[V,E] = EIGENSYS(A) tên một vectơ symbolic E chứa các giải trẻ riêng và một ma trận symbolic V chứa các vectơ riêng của một ma trận sáu hoặc symbolic A. Các vectơ riêng coi thảo ảnh hưởng điều trong các phần tử của E(n), ở đây n là sáu nguyên dạng ảnh hưởng cho vectơ giải trẻ riêng E

[V,E] = EIGENSYS(VPA(A)) tên các giải trẻ riêng và các vectơ riêng bằng sáu, bằng cách dùng sáu hoặc

Về đầu:

```

eigensys(rosset)
[v,e] = eigensys(sym('[a,b,c; b,c,a; c,a,b]'))

```

EXPAND

Khai triển symbolic

EXPAND (S) khai triển mẫu phân tử của S, S là ma trận symbolic

EXPAND khai triển chuỗi yáo là các âa thâc. Nó cũng khai triển mẫu số các hàm toân hoâc, nhâc hàm læúng giâc, hàm mũ và hàm loga

EZPLOT

Đùng âõ vẽ âõ thê hàm

EZPLOT(f) vẽ âõ thê của f(x), vâi f là mẫu biâu thâc symbolic biâu hiên mẫu biâu thâc toân boac gôm mẫu biâu symbolic, gõi là 'x'. Miôn giãi trê của trục x trong khoâng -2*pi và 2*pi

EZPLOT(f,[xmin xmax]) ðùng âõ chê âên miôn giãi trê của x thay cho ngôm âên là [-2*pi, 2*pi]

EZPLOT(f,[xmin xmax],fig) ðùng hñh vẽ chê âên thay cho hñh ánh hiên thâi

Vê ðu:

```
ezplot('erf(x)')
ezplot erf(x)
ezplot('tan(sin(x))-sin(tan(x))')
ezplot tan(sin(x))-sin(tan(x))
```

FACTOR

Thâi số symbolic

FACTOR(S), nhâc S là mẫu ma trận symbolic matrix âât thâi số mẫu phân tử của S

FACTOR(N), nhâc N là mẫu ma trận nguyên thç thên thâi số nguyên tã của mẫu phân tử của N

FINDCOMMA

Tìm các dấu phâc khâng cõ bñ trong cãp ngoâc ân

FINDCOMMA(S) là vectơ các chê số của các dấu phâc (',') trong chuâi S mà khâng á trong các cãp ngoâc ân phui hũp

k = findcomma('fun1(x), fun2(x,y), fun3(x), fun4(x,y), fun5') traí vã k = [8 19 28 39]

Khâng âm các cãp ngoâc ân trong [16 36]

Các phép toán I f và finv có thể thoát khỏi nếu các biểu thức symbolic tăng cũng không thu được đúng. Dòng thời hai của các nút dịch vai chia trước $f(x)$ theo tham số 'a'

Các phép toán lại:

```
f + a      - Thay f(x) bởi f(x) + a
f - a      - Thay f(x) bởi f(x) - a
f * a      - Thay f(x) bởi f(x) * a
f / a      - Thay f(x) bởi f(x) / a
f ^ a      - Thay f(x) bởi f(x) ^ a
f(x+a)     - Thay f(x) bởi f(x + a)
f(x*a)     - Thay f(x) bởi f(x * a)
```

Dòng thời ba của các nút lại các phép toán nhờ hàm tên trên cái hai $f(x)$ và $g(x)$.

Các phép toán lại:

```
f + g      - Thay f(x) bởi f(x) + g(x)
f - g      - Thay f(x) bởi f(x) - g(x)
f * g      - Thay f(x) bởi f(x) * g(x)
f / g      - Thay f(x) bởi f(x) / g(x)
f(g)       - Thay f(x) bởi f(g(x))
g = f      - Thay g(x) bởi f(x)
swap       - Đổi f(x) và g(x)
```

Ba nút đầu trên dòng thời tạo quản lý một danh sách các hàm. Nút Insert thêm hàm đang kích hoạt vào danh sách. Nút Cycle chuyển qua danh sách hàm. Nút Delete xóa hàm kích hoạt ra khỏi danh sách. Danh sách các hàm có tên fxlist. Ngõm ảnh fxlist chứa một số hàm ảnh quan tâm

Nút Reset thêm f, g, x, a và fllait vào các giá trị đầu. Nút Help in ra và bán trả giúp này

Nút Demo chạy màu

Nút Close ảnh cái ba cửa sổ

HORNER

Biểu thức ảnh tạo Horner

HORNER(P) biểu ảnh đổi ảnh thức symbolic, P, sang biểu thức ảnh Horner của nó

Về đầu:

```
Nếu p = 'x^3-6*x^2+11*x-6'      th
horner(p)  là 'x*(x*(x-6)+11)-6'
```

INT

Tích phân

INT(S) là tích phân bất ảnh của S tăng cũng với biểu thức symbolic của nó

INT(S,'v') là tích phân bất ảnh của S tăng cũng với biểu thức v

INT, không tham số, là tích phân bất ảnh của biểu thức trên ảnh tăng cũng với biểu thức symbolic của nó

INT(S,a,b) là tích phân xác định ảnh của S tăng cũng với biểu thức symbolic của nó tại a ảnh b

INT(S,'v',a,b) là tích phân xác định ảnh của S tăng cũng với biểu thức v tại a ảnh b

Về đầu: int('1/(1+x^2)') là arctan(x) .

INVERSE

Nghệch âaío ma tráũn symbolic

INVERSE(A) tên nghệch âaío symbolic của ma tráũn A, vài A là mãũt ma tráũn symbolic hoàũc ma tráũn sãũ

INVERSE(VPA(A)) dùng âãũ chênħ xăĩc sãũ hoũc thay âãøi

Vê duũ: `inverse(sym(5,5,'1/(i+j-t)'))`

INVFOURIER

Biãũn âãøi tên phán nghĩch âaío Fourier

f = INVFOURIER(F) là biãũn âãøi tên phán nghĩch âaío Fourier của biãøu thăĩc F,

$f(t) = 1/(2\pi) \int_{-\infty}^{\infty} F(w) \exp(iw \cdot t) dw$

f = INVFOURIER(F, 'x') là hàm của 'x' thay cho 't'

f = INVFOURIER(F, 'x', 'v') giáĩ thiăũt F là hàm của 'v' thay cho 'w'

f = INVFOURIER, khăng âãũi sãũ nhăũp, biãũn âãøi kăũt quăĩ trăăĩc

Vê duũ:

```

invfourier                                exp(-w^2)
1/2/pi^(1/2)*exp(-1/4*t^2)
invfourier 1/(w-i)                        i*exp(-
t)*Heaviside(t)
```

INVLAPLACE

Biãũn âãøi nghệch âaío Laplace

f = INVLAPLACE(F) là biãũn âãøi nghệch âaío Laplace của biãøu thăĩc symbolic F,

$f(t) = \int_0^{\infty} F(s) \exp(s \cdot t) ds$

f = INVLAPLACE(F, 'x') là hàm của 'x' thay cho 't'

f = INVLAPLACE(F, 'x', 'v') giáĩ thiăũt F là hàm của 'v' thay cho 's'

f = INVLAPLACE, khăng âãũi sãũ nhăũp, biãũn âãøi kăũt quăĩ trăăĩc

Vê duũ:

```

invlaplace                                1/(s-1)
exp(t)
invlaplace('(2*s^2+2+s^3)/s^3/(s^2+1)')
t^2+sin(t)
invlaplace('t^(-5/2)', 'x')
4/3/pi^(1/2)*x^(3/2)
invlaplace('laplace(f(t))')
f(t)
```

INVZTRANS

Biãũn âãøi nghệch âaío Z

f = INVZTRANS(F) là biãũn âãøi nghệch âaío Z của biãøu thăĩc symbolic F,

$f(n) = 1/(2\pi i) \oint_{\gamma} F(z) z^{n-1} dz$

f = INVZTRANS(F, 'x') là hàm của 'x' thay cho 'n'

f = INVZTRANS(F, 'x', 'v') giáĩ thiăũt F là hàm của 'v' thay cho 'z'

f = INVZTRANS, khăng âãũi sãũ nhăũp, biãũn âãøi kăũt quăĩ trăăĩc

Vê duû:

```
invztrans z/(z-1)          1
invztrans z/(z-a)          a^n
invztrans('exp(x/z)', 'k', 'z') x^k/k!
invztrans(ztrans('f(n)'))  f(n)
```

JACOBIAN

Ma trấun Jacobian

JACOBIAN(f,v) tênh Jacobian của âaui lăăung và hăăing hoàuc vectă f ăing vắi vectă v. Phăon tắi thắi (i,j) của kắut quắi lài $df(i)/dv(j)$. Lău yắ rằong khi f lài âaui lăăung và hăăing thắ Jacobian của f lài f

Vê duû:

```
jacobian(sym('x*y*z; y; x+z'), sym('x,y,z'))
jacobian('u*exp(v)', sym('u,v'))
```

JORDAN

Daung Jordan Canonic

JORDAN(A) tênh Daung Jordan Canonical/Daung chuăon của ma trấun A. Ma trấun phải âăăuc biăút chềnh xắc, vắ vắu cắc phắon tắi phải nguyắn hoàuc phắn sắu của cắc sắu nguyắn nhoắ (hắu tề). Mắut lắui bắút kắi trong ma trấun nhắp coắ thắo làm thay âăoỉ hoắn toắn JCF của noắ

$[V, J] = JORDAN(A)$ cắng tênh pheíp biăún âăui tắng tắu, V, sao cho $V \backslash A * V = J$. Cắc cắut của v lài cắc vectă riắng tắong quắit

Vê duû:

```
[V,J] = jordan(gallery(5))
```

LAMBERTW

Hằm W của Lambert

$w = \text{lambertw}(x)$ âăăuc giáắi thắnh $w * \exp(w) = x$

LAPLACE

Biăún âăoỉ Laplace

$F = \text{LAPLACE}(f)$ lài biăún âăoỉ Laplace của biăou thắc symbolic F,

```
F(s) = int(f(t)*exp(-s*t), 't', 0, inf)
```

$F = \text{LAPLACE}(f, 'v')$ lài hằm của 'x' thay cho 's'

$F = \text{LAPLACE}(f, 'v', 'x')$ giáắ thiăút F lài hằm của 'v' thay cho 't'

$F = \text{LAPLACE}, ,$ khắng âăúi sắu nhắp, biăún âăoỉ kắut quắi trắắc

Vê duû:

```
laplace exp(t)          1/(s-1)
laplace                  t^2+sin(t)
(2*s^2+2+s^3)/s^3/(s^2+1)
laplace('y^(3/2)', 'z')
3/4*pi^(1/2)/z^(5/2)
laplace(diff('F(t)'))
laplace(F(t), t, s)*s-F(0)
```

LATEX

Biăou hiăun LaTeX của giáắ trắ xuăút symbolic

$\text{LATEX}(S)$ in biăou hiăun LaTeX của S

```

LATEX(S,'filename') cũng in nội sang tệp chèn ảnh
Về đầu:
r = '(1+2*x+3*x^2)/(4+5*x+6*x^2)'
latex(r)
{\frac {1+2\,x+3\,x^{2}}{4+5\,x+6\,x^{2}}}
H = hilb(3);
latex(H,'hilb.tex')
\left [ \begin{array}{ccc}
1&1/2&1/3 \\
1/3&1/4&1/5 \end{array} \right ]

```

LINSOLVE

Giaíi hệ phương trình tuyến tính

$X = \text{LINSOLVE}(A,B)$, với ma trận A , giải $A \cdot X = B$.

Mất thăng bằng khiếu cáo ảnh hưởng in ra nếu ma trận A suy biến

$[X,Z] = \text{LINSOLVE}(A,B)$ cũng tên Z , mất cả số cho không gian không của A . Lại giải tổng quát cho hệ tuyến tính là $X + Z \cdot p$, với p là một vectơ (hoặc ma trận) các tham số tự do

MAPLE

Truy cập hàm nhân Maple

$\text{MAPLE}('l\grave{a}nh')$ giải lệnh cho hàm nhân Maple vài trái vế kết quả lại một biểu thức symbolic. Một dấu chấm phẩy với cuối phép Maple ảnh hưởng nếu cần vào các lệnh nếu cần $\text{MAPLE}('function', \text{ARG1}, \text{ARG2}, \dots)$ giúp nhận tất hàm Maple trong nháy ảnh vài lần ảnh 10 ảnh số. Các ảnh số ảnh chuyển sang các biểu thức symbolic nếu cần, rồi hàm chèn ảnh ảnh gửi với các ảnh số ảnh ảnh cho. Kết quả trái vế trong một biểu thức symbolic

$[\text{RESULT}, \text{STATUS}] = \text{MAPLE}(\dots)$ trái vế trả về hai khiếu cáo/lỗi. Khi lệnh ảnh thực hiện thành công thì RESULT là kết quả vài $\text{STATUS} = 0$. Nếu thất bại thì RESULT là một khiếu cáo/lỗi trả về, vài STATUS là một số nguyên dương

$\text{MAPLE}('traceon')$ cho ra dãy lệnh Maple luôn trả về kết quả ảnh ảnh in ra $\text{MAPLE}('traceoff')$ tắt việc này

MAPLEMEX

Tập Mex-file giao diãun vớì Maple

Thăng thằìng, hàm này ằằằ gửi bắì M-file củà "maple". Nớì thằìng khằìng gửi trằìc tiằìp tài ðoìng lằình

[RESULT,STATUS] = MAPLEMEX(STATEMENT) gắì cắì lằình ằằ cho vào hát nhắì OEM củà Maple, nớì cho mắìt kắìt quắì vài mắìt biằìu hiằình trằìng thằìì. Mắìt ằằì sắì nhắìp lằìu choình thằì hai ằằ ằằình dắìu ằằìu kiằình ằằìu hoằìc in ra trằìc tiằìp. Hàm này ằằằ viằìt bằìng C vài biằìn ðẻch sang mắìt tắìp Mex-file. Kắìt quắì làì mắìt tắìp vớì tắìn ðắìng "maplemex.mexx", "mexx" làì tắìn mắì rằìng. Nắìu khằìng cớì tắìp thằì tắìp M-file này sẽ ằằằ thằìc hiằình vài kắìt quắì làì mắìt thăng bắìo lằìì

MAPLEINIT

Khắì taừ MAPLE

MAPLEINIT ằằằ gửi bắì MAPLEMEX ằằ khắì taừ hát nhắì Maple

MAPLEINIT xắì ằằình ằằìng dằình chề thằì muừ chằìì thằì viằình Maple, nhắìp gớì hằìng ằằì sắì tuyằình tằình, khắì taừ cắìc chằì sắì, thiằìt lắìp mắìt sắì phạm vì. Tập M-file này, "symbolic/mapleinit.m", cớì thằì ằằằ sắì ằằì ằằì trưởì cắì Maple V, Release 2, Thằì viằình bắìt kỳì ằằ cớì thằì ằằằ

MFUN

ằằì lằìng sắì củà mắìt hàm Maple

MFUN('fun',p1,p2,p3,p4), 'fun' làì tắì mắìt hàm Maple vài p1, p2, p3 vài p4 gắì trề sắì ằìng vớì cắìc tham sắì củà hàm. Tham sắì củằì cùng cớì thằì làì mắìt ma trắình. Tắìt cắì cắìc tham sắì khắì phắì ằằằ chề ằằình kiằìu bắìì hàm củà Maple. MFUN ằằì lằìng sắì hàm 'fun' vớì cắìc tham sắì chề ằằình vài trắì vắì gắì trề sắì củà MATLAB. Mớì suy biằình trong 'fun' ằằì trắì vắì NaN

Về ðuì:

```
x = 0:0.1:5.0;
y = mfun('FresnelC',x)
```

MFUNLIST

Cắìc hàm ằằì biằìt củà MFUN

Cắìc hàm ằằì biằìt ằằằ liằìt kắì theo thằì tằì alphabet. n biằìu hiằình ằằì sắì nguyắìn, x biằìu hiằình ằằì sắì thằìc, vài z biằìu hiằình ằằì sắì phằìc. ằằ biằìt thằì chỉ tiắìt cắìc mắì táì củà cắìc hàm, kắì cắì cắìc hằình chắì vắì ằằì sắì, thằì xem tài liằìu tham khắì hoằìc ðuìng MHELP

bernoulli	n	Cắìc sắì Bernoulli
bernoulli	n,z	Cắìc ằằ thằìc
Bernoulli		
BesselI	x1,x	Hàm Bessel loaìì
1		
BesselJ	x1,x	Hàm Bessel loaìì1

BesselK	x1,x	Hàm Bessel loaûi 2
BesselY	x1,x	Hàm Bessel loaûi 2
Beta	z1,z2	Hàm Beta
binomial	x1,x2	Caïc hãu sãu nhẽ thãic
LegendreKc aúi loaûi 1	x	Têch phán Elliptic âáöy
LegendreEc aúi	x	Têch phán Elliptic âáöy
LegendrePic aúi loaûi 3	x1,x	Têch phán Elliptic âáöy
LegendreKc1 bùi	x	LegendreKc dùng mãân
LegendreEc1 bùi	x	LegendreEc dùng mãân
LegendrePic1 bùi	x1,x	LegendrePic dùng mãân
erfc	z	Hàm sai sãu bùi
erfc	n,z	Têch phán lãûp
cuía hàm sai sãu bùi		
Ci	z	Têch phán Cosin
dawson	x	Têch phán Dawson
Psi	z	Hàm Digamma
dilog	x	Têch phán
Dilogarithm		
erf	z	Hàm sai sãu
euler	n	Caïc sãu Euler
euler	n,z	Caïc âa thãic
Euler		
Ei	x	Têch phán muí e
Ei	n,z	Têch phán muí e
FresnelC	x	Têch phán Cosin Fresnel
FresnelS	x	Têch phán Sin Fresnel
GAMMA	z	Hàm Gamma
harmonic	n	Hàm Harmonic
Chi	z	Têch phán Cosin
Hyperbol		
Shi	z	Têch phán Sin Hyperbol
hypergeom quaít)	X1,X2	Hàm Hypergeometric (tãong
LegendreF hoàn thành loaûi 1	x,x1	Têch phán Elliptic chãa
LegendreE hoàn thành loaûi 2	x,x1	Têch phán Elliptic chãa
LegendrePi hoàn thành loaûi 3	x,x2,x1	Têch phán Elliptic chãa
GAMMA thành	z1,z2	Hàm Gamma chãa hoàn
W	z	Hàm W cuía
Lambert		
W	n,z	Hàm W cuía
Lambert		
lnGAMMA Gamma	z	Logarit cuía hàm
Li	x	Têch phán Logarit
Psi	n,z	Hàm Polygamma

Ssi	z	Têch	phán	dêch
chuyãøn				
Si	z	Têch	phán	Sin
Zeta	x	Hàim		Zeta
(Riemann)				
Zeta	n,x	Hàim		Zeta
(Riemann)				
Zeta	n,x,x1	Hàim	Zeta	(Riemann)
Caïc	âa	thăic	trăuc	giao (chè cho
Math Toolbox	mải	răüng)		Symbolic
T	n,x	Chebyshev	loaúi	1
U	n,x	Chebyshev	loaúi	2
G	n,x1,x	Gegenbauer		
H	n,x	Hermite		
P	n,x1,x2,x	Jacobi		
L	n,x	Laguerre		
L	n,x1,x	Laguerre	tăøng	quaít
P	n,x	Legendre		

MHELP

Trăü giuíp cuía Maple

MHELP topic in ra vãn baín trăü giuíp cuía Maple vảo
váuñ âăö topic

MHELP('topic') giăúng lăüh trăn

MPA

Lăüh gaín cuía Maple

MPA('v','expr') gaín expr cho biăún symbolic v trong
vùng làm viăüc cuía Maple. expr coỉ thăø lài măt
biăún symbolic, măt biăøu thăic symbolic, hoàüc măt
giaỉ trê săú. Daùng lăüh thăàng coỉ êch. Trong
trăàng hăüp năy, coỉ 3 daùng lăüh kháic nhau:

mpa v = expr

mpa v := expr

mpa v expr

Ba daùng năy chè hăüp lăüh khi daùng lăüh tăøng
quaít hăüp lăüh, văii ngoaúi lăüh cuía lăüh gaín. Âăö
lăúy năúi dung cuía v tặi vùng làm viăüc cuía Maple,
duòng caïc lăüh sau:

v = maple('v')

v = maple('print(v)')

Vê du:

mpa a = 1

mpa b = sqrt(1/2)

mpa s = (a+b)/2

mpa('P',pascal(3))

mpa R = evalm(inverse(P-s*eye))

maple print(R)

PROCREAD(FILENAME) àoüc tấủp chè àềnh chặỉa vàn báỉn nguẫủn của mặủt thuỉ tuủc Maple. Nổỉ xỏỉa cặỉc lặỉi chủỉ thềch vặỉ cặỉc kyỉ tặủ sang đỏỉng, rặỉi gặỉi chuặủi kặủt quặỉ sang Maple. Symbolic Toolbox mặỉ rặỉng yặủ cặủu

Về đứ: Gặỉ sặỉ tấủp "check.src" chặỉa nặủi đung nhặ sặ

```

check := proc(A)
#   check(A) computes A*inverse(A)
    local X;
    X := inverse(A):
    evalm(A &* X);
end;
Thợ lặủnh
procread('check.src')

```

cặỉi àầủt thuỉ tuủc. Nổỉ cỏỉ thặủ àầủc truy cặủp vặỉi

```

maple('check',magic(3))          hoặủc
maple('check',vpa(magic(3)))

```

RSUMMER

Àặỉc lặủng vặỉ hiặủn thề tặủng Riemann

RSUMMER('expr',n) hiặủn mặủt àầủ thề của tặủng Riemann của 'expr' đừủng n àỉặủm trặỉn [0,1]

RSUMS

Àặỉc lặủng cỏỉ tặủng tặỉc của cặỉc tặủng Riemann

RSUMS(f) xặủp xề tềch phặỉn của f(x) bặỉi cặỉc tặủng Riemann

RSUMS thặủng àầủc gỏỉi vặỉi đặủng đỏỉng lặủnh, nhặ rsums exp(-5*x^2)

SHIFTEPT

Đềch chuyểủn đặủu chặủm àầủng trong cặỉc sặủ đặủng khoa hoặủ

```
SHIFTEPT('1234.0E10') = '1.234e13'
```

SIMPLE

Tặủ đặủng àầủn gặỉn nhặủt của mặủt biặủu thặỉc symbolic

SIMPLE(EXPR) lặủy mặủt sặủ đặủng àầủi sặủ àầủn gặỉn của biặủu thặỉc EXPR, hiặủn thề mỏỉi biặủu hiặủn ruỉt gỏỉn àầủ đặỉ của biặủu thặỉc EXPR vặỉ trặỉ vặủ đặủng ngặủn nhặủt

[R,HOW] = SIMPLE(EXPR) khặủng hiặủn thề cặỉc đặủng àầủn gặỉn trung gian, nhặủg trặỉ vặủ đặủng ngặủn nhặủt tặủ àầủc, củnặỉiiii chặủi mặỉ tặỉ cặỉch àầủn gặỉn hoặỉ

SIMPLE, khặủng àầủi sặủ, đừủng biặủu thặỉc trặỉc

Về đứ:

S	R
How	
cos(x)^2+sin(x)^2	1
simplify	
2*cos(x)^2-sin(x)^2	3*cos(x)^2-1
simplify	

$\cos(x)^2 - \sin(x)^2$	$\cos(2*x)$
<code>combine(trig)</code>	
$\cos(x) + (-\sin(x)^2)^{1/2}$	$\cos(x) + i*\sin(x)$
<code>radsimp</code>	
$\cos(x) + i*\sin(x)$	$\exp(i*x)$
<code>convert(exp)</code>	
$(x+1)*x*(x-1)$	$x^3 - x$
<code>collect(x)</code>	
$x^3 + 3*x^2 + 3*x + 1$	$(x+1)^3$
<code>factor</code>	
$\cos(3*\arccos(x))$	$4*x^3 - 3*x$
<code>expand</code>	

SIMPLER

Rút gọn biểu thức

`SIMPLE(HOW,S,R,H,P,X)` áp dụng phảng phaíp HOW với tham số tùy chọn X cho biểu thức S, in kết quả nếu $P \neq 0$, so sánh đầu dài của kết quả với biểu thức R, nhân nếu phảng phaíp H, và trái vế chuỗi ngắn nhất và phảng phaíp tăng sẽ

SIMPLIFY

Ấn giá trị hoặ symbolic

`SIMPLIFY(S)` ấn giá trị mội phởn tấi của ma trấu symbolic S

Vê du: `simplify('sin(x)^2 + cos(x)^2') = 1`

SINGVALS

Các giá trị vậ vectả kỳ dễ của ma trấu symbolic

`SINGVALS(A)` tên giá trị kỳ dễ symbolic của ma trấu A

`SINGVALS(VPA(A))` tên giá trị kỳ dễ bàng sáu bàng cách dùng đầu chên xấi sáu hoặ thay đầu

`[U,S,V] = SINGVALS(VPA(A))` cho 2 ma trấu trấu giao với đầu chên xấi thay đầu, U và V, và ma trấu cheo vpa, S, đầu symop(U, '*', S, '*', transpose(V)) = A

Các vectả kỳ dễ symbolic khăng nếu dùng trấu tiấu

Vê du:

```
A = sym('[a, b, c; 0, a, b; 0, 0, a]');
s = singvals(A)
A = magic(8);
s = singvals(A)
[U,S,V] = singvals(vpa(A))
```

SININT

Hàm tích phân Sin

`SININT(x) = int(sin(t)/t, t=0..x)`

SM2AR

Chuyển ma trấu symbolic sang máng Maple

`A = SM2AR(M)` chuyển ma trấu sáu hoặ ma trấu symbolic sang máng 'array([[...],[...]])' đầu dùng bấi các hàm đầu sáu tấu tên của Maple

SOLVE

Giaíi hữu phảng trệnh âaui sảu symbolic

Phảng trệnh mảut biảu:

SOLVE(S), S lài phảng trệnh symbolic hoàuc lài mảut biảu thảic symbolic thệ giaíi phảng trệnh âaui cho, hoàuc phảng trệnh $S = 0$, biảu tầu đơ cuía nũi âaui xảic âềnh bảii SYMVAR

SOLVE(S,'v') giaíi theo biảu 'v'

Hữu phảng trệnh nhiảu biảu

SOLVE(S1,S2,...,SN) giaíi hữu N phảng trệnh symbolic N biảu xảic âềnh bảii SYMVAR

SOLVE(S1,S2,...,SN,'v1,v2,...,vn') giaíi hữu N phảng trệnh symbolic N biảu chề âềnh bảii N âaui sảu nhữp cuảui cườg

[X1,X2,...,XN] = SOLVE(S1,S2,...,SN), vài

[X1,X2,...,XN] = SOLVE(S1,S2,...,SN,'v1,v2,...,vn')

traí vữ N vectả symbolic chảia cảic biảu thảic theo cảic biảu riảng biảut trong lảii giaíi. Trong tẩt cải cảic trầing hữp thệ traí vữ giaí trề sảu nửu khảng tặm thẩy lảii giaíi symbolic

Về đườ:

solve('log(x) = x/pi')

x = solve('a*x^2 + b*x + c')

b = solve('a*x^2 + b*x + c', 'b')

[x,y] = solve('x^2 + 2*x*y + y^2 = 4', 'x^3 + 4*y^3 = 1')

[u,v] = solve('a*u^2 + v^2', 'u - v = 1', 'u,v')

[a,u,v] = solve('a*u^2 + v^2', 'u - v = 1', 'a^2 - 5*a + 6')

SUBS

Thay thẩ kyi hiảu trong mảut biảu thảic hoàuc ma trầu symbolic

SUBS(S,NEW) Thay biảu symbolic trong S bảii NEW

SUBS(S,NEW,OLD) Thay tẩt cải OLD trong S bảii NEW

Về đườ:

subs sin(x) pi/3 = 'sin(1/3*pi)'

subs sin(z) x+i*y = 'sin(x+i*y)'

f = 'F(a*r^2)'

r = 'sqrt(x^2+y^2)'

subs(f,r,'r') = 'F(a*(x^2+y^2))'

SVDVPA

Taích giai trẻ biăún kyì dề
 SINGVALS cuing coì thăø tênh caic giai trẻ kyì dề.
 SVDVPA âăüc thay bảii SINGVALS. Năn dùng:
 $S = \text{SINGVALS}(\text{VPA}(A))$ thay cho $S = \text{SVDVPA}(A)$
 $[U, S, V] = \text{SINGVALS}(\text{VPA}(A))$ thay cho $[U, S, V] = \text{SVDVPA}(A)$

SYM

Taô ra, truy câp hoàuc sảia âăøi măt ma trấún symbolic

Măt ma trấún symbolic lài măt máng vàn báin MATLAB coì mătì òòng bắót áăôu vắi

'[', kắút thuic vắi ']', vài chắia caic chuắuì con caich nhau bắii caic đắu phắy áăø biắôu hiắún caic phắn tái riắng biắút . Coì 3 caich taô ra caic ma trấún symbolic :

SYM(X) chuyắn ma trấún sắu X sang đắung symbolic cuía nắi vắi caic phắn tái áăüc biắôu hiắún bằng phắn sắu (nhắún áăüc tắi SYMRAT)

SYM(m,n,'expr') taô ra ma trấún symbolic cái mxn, caic phắn tái cuía ma trấún symbolic áăüc sắi lắúng áăúi vắi i = 1:m vài j = 1:n. Biắôu thắic expr lài măt biắôu thắic symbolic thắìng chắia caic kyỉ tắu 'i', 'j', vài caic biắún tắu do kắic

SYM(m,n,'r','c','expr') dùng 'r' vài 'c' lài caic biắún òòng vài cắút thay cho 'i' vài 'j'

SYM('[s11,s12,...,s1n; s21,s22,...; ...,smn]') taô ra ma trấún symbolic cái mxn bằng caich dùng caic phắn tái symbolic s11, s12, ..., smn. Đắung nắy cuía symbolic giắúng hắút phắt sinh ma trấún sắu trong MATLAB. Caic đắu chắúm phắy kắút thuic caic òòng. Coì 2 caich áăø truy câp caic phắn tái riắng biắút cuía ma trấún symbolic:

SYM(S,i,j,'expr') lài phiắn báin symbolic cuía S(i,j) = 'expr'

r = SYM(S,i,j) lài phiắn báin symbolic cuía r = S(i,j)

Về du:

M = sym(hilb(3)) lài măt ma trấún vàn báin vắi 3 òòng,

```
[ 1, 1/2, 1/3]
[1/2, 1/3, 1/4]
[1/3, 1/4, 1/5]
```

```
M = sym(3,3,'1/(i+j-t)') phắt sinh
[1/(2-t), 1/(3-t), 1/(4-t)]
[1/(3-t), 1/(4-t), 1/(5-t)]
[1/(4-t), 1/(5-t), 1/(6-t)]
```

M = sym(M,1,3,'1/t') thay áăøi phắn tái (1,3) cuía M thắìng '1/t'

M = sym('a, 2*b, 3*c; 0, 5*b, 6*c; 0, 0, 7*c') phắt sinh ma trấún symbolic tam giai trắn coì áềnh thắic determ(M) = 35*a*b*c. Sau áôi M, sym(M,1,3) = '3*c'

SYM2POLY

Âäøi äa thæic symbolic sang vectå hãu säú cuía äa thæic

SYM2POLY(p) trái vãö vectå hãu säú cuía äa thæic symbolic p

Vê duû:

```
sym2poly('x^3 - 2*x - 5') = [1 0 -2 -5]
```

SYMADD

Cäüng symbolic

SYMADD(A,B) tênh täøng symbolic A + B

Vê duû:

```
symadd('cos(t)','t') = 'cos(t)+t'
```

SYMDIFF

Vi phán symbolic

Hàm naìy thæäüng äæäüc goüi báíi DIFF äãø tênh ääüo hàm

SYMDIFF(S) vi phán S theo biãún tæü do cuía noí

SYMDIFF(S,'v') vi phán S theo biãún 'v'

SYMDIFF(S,n) vai SYMDIFF(S,'v',n) vi phán S n láön

SYMDIFF, khäng tham säú, vi phán biãøu thæic træäic

SYMDIV

Chia symbolic

SYMDIV(A,B), vãi cáic biãøu thæic hoàüc ma tráün symbolic A vai B, tênh A / B

Vê duû:

```
symdiv('2*cos(t)+6',3) trái vãös 2/3*cos(t)+2
```

Nãúu

A =

```
[ 2, a + 3/2]
[ 7/6, a/2 + 1]
```

B =

```
[ 1, 1/2]
[ 1/2, 1/3]
```

thç symdiv(A,B) trái vãö

```
[ -1-6*a, 6+12*a]
[-4/3-3*a, 5+6*a]
```

SYMMUL

Nhán symbolic

SYMMUL(A,B), vãi cáic biãøu thæic hoàüc ma tráün symbolic A vai B, tênh têch ääüi säú tuyãún tênh symbolic A * B

Vê duû:

```
symmul('x','exp(x)') = 'x*exp(x)'
```

SYMOP

Tênh toaìn symbolic

SYMOP(arg1,arg2,arg3,...) láúy äãún 16 ääüi säú. Mäüi däúi säú coí thãø lai mäüt ma tráün symbolic, mäüt ma tráün säú, hoàüc mäüt trong cáic pheïp toaìn sau: '+', '-', '*', '/', '^', '(',')'

SYMOP(...) nối các biểu thức và các biểu thức biểu thị các kết quả

Về đầu:

```
x = 'x'
f = symop(1,'+',x,'+',x,'^',2,'/',2);
symop(f,'-',int(diff(f)))
symop('exp(x)','/',('(',f,'+',x,'^3','/',6,')'))
```

```
G = sym('[c, s; -s, c]')
symop(G,'*',transpose(G))
```

Lưu ý: Việc hoán vị các biểu thức và hằng số các ma trận có thể có các kết quả khác nhau. Về đầu, symop(A,'+',x) cũng x vào ảnh hưởng của A

SYMPOW

Tên lũy thừa của một biểu thức hoặc ma trận symbolic

SYMPOW(S,p) tên S^p. Nếu S là một biểu thức symbolic thì p có thể là một biểu thức symbolic và hằng số hoặc biểu thức số và hằng. Nếu S là ma trận symbolic thì S phải vuông, và p phải là một số nguyên

Về đầu: sympow('exp(t)',2) = 'exp(t)^2'

SYMRAT

Xấp xỉ phân số symbolic

SYMRAT(X), với và hằng X, là một chuỗi biểu thức hiên một số nguyên, phân số, phân số nhân 'pi' hoặc số nguyên mũ 2. Khi chuỗi biểu thức các biểu thức với số chữ số ảnh hưởng của MATLAB thì kết quả cho lỗi ảnh hưởng giữa trẻ X

Về đầu:

```
symrat(22/7) = '22/7'
symrat(2*pi/3) = '2*pi/3'
symrat(1.e12) = '1000000000000'
symrat(eps) = '2^(-52)'
```

SYMSIZE

Kích thước ma trận symbolic

D = SYMSIZE(S), với ma trận S cỡ MxN, trái với hai vectơ dòng gồm 2 phần tử

D = [M, N] chứa số dòng và số cột trong ma trận.

[M,N] = SYMSIZE(S) trái với số dòng và số cột trong 2 biểu thức riêng biệt

M = SYMSIZE(S,1) trái với ảnh số dòng

N = SYMSIZE(S,2) trái với ảnh số cột

SYMSUB

Trừ symbolic

SYMSUB(A,B) với các biểu thức hoặc ma trận symbolic A và B, tên A - B

SYMSUM

Tổng symbolic

SYMSUM(S) lài tặng vặ haùn S theo biấún symbolic cuía noỉ

SYMSUM(S,'v') lài tặng vặ haùn S theo biấún v

SYMSUM, khặng ấúủ sắủ, lài tặng vặ haùn theo biấún symbolic cuía cuía biấủu thặic trặic

SYMSUM(S,a,b) lài tặng vặ haùn S theo biấún symbolic cuía noỉ tặi a ấúủ b

SYMSUM(S,'v',a,b) lài tặng vặ haùn S theo biấún v tặi a ấúủ b

Về dụ:

symsum k^2	1/3*k^3-
1/2*k^2+1/6*k	
symsum k^2 0 n-1	1/3*n^3-
1/2*n^2+1/6*n	
symsum k^2 0 10	385
symsum k^2 11 10	0
symsum 1/k^2	-Psi(1,k)
symsum 1/k^2 1 Inf	1/6*pi^2
symsum x^k/k! k 0 Inf	exp(x)

SYMVAR

Xaíc ấềnh cáic biấún symbolic trong mặit biấủu thặic

SYMVAR(S) tặm trong chuầủi s ấầủ lắủy mặit kyỉ tặủ chặi thặểng riặng biấủt, kặic 'i' hoặủc 'j', ấỏỉ lài mặit phắủn cuía 1 tặi tặủ thặinh tặi mặit sắủ kyỉ tặủ. Nắủu coỉ kyỉ tặủ ấỏỉ vặi lài dụy nhắủt thặ tráỉ vặủ kyỉ tặủ ấỏỉ. Nắủu khặng coỉ thặ tráỉ vặủ 'x'

Nắủu kyỉ tặủ khặng dụy nhắủt thặ tráỉ vặủ mặit kyỉ tặủ gắủn 'x'

Nắủu coỉ rặng buặủc thặ mặit kyỉ tặủ la tặnh ấầủc choủn

SYMVAR(S,'t') choủn biấún gắủn 't' thay cho 'x'

SYMVAR(S,N), vắỉ sắủ nguyặn vặ hặểng N, tặm N kyỉ tặủ kặic nhau trong, kặủ cáỉ 'i' vặi 'j' Nắủu coỉ N kyỉ tặủ thặ tráỉ vặủ đặnh sặich chặing. Ngặểủc lắủi thặ kắủt quặi lài mặit lắủi SYMVAR(S,N), vắỉ vectắ nguyặn ểt nhắủt 2 thặinh phắủn thặ tặm mặit sắủ kyỉ tặủ kặic nhau Kặi N lài mặit vectắ thặ SYMVAR(S,N) khặng bao gặi thặng bặủ lắủi. Nắủu sắủ tặm thắủy gặiủa mặn(N) vặi max(N), thặ tráỉ vặủ mặit đặnh sặich. Nắủu sắủ tặm thắủy ểt hặn mặn(N), thặ tráỉ vặủ mặit ma trắủn rặng . Nắủu sắủ tặm thắủy lắủn hặn max(N), thặ tráỉ vặủ NaN

Về dụ:

```

symvar('sin(x)') = 'x'
symvar('sin(pi*t)') = 't'
symvar('a+y') = 'y'
symvar('3*i+4*j') = 'x'
symvar('pi',[1 1]) = rặng
f = '3*x+4*y';
symvar(f) = 'x'
symvar(f,2) = 'x, y'
g = 'Dx = y; Dy = -x + sin(t)';
symvar(g,2:3) = 't,x,y'
symvar(g,[1 1]) = NaN

```



```

vpa pi 1919                                lài măt main hợnh
ääöy caïc säú cuía pi
vpa exp(pi*sqrt(163)) 36                    hiăün măt säú
"gáön nguyăñ"

```

ZETA

Hàm Zeta Riemann

```
ZETA(s) = sum(1/k^s,k=1..infinity)
```

ZTRANS

Biăún ääøi Z

F = ZTRANS(f) lài biăún ääøi Z cuía biăøu thăïc
symbolic f,

```
F(z) = symsum(f(n)/z^n,'n',0,inf)
```

F = ZTRANS(f,'v') lài hàm theo biăún 'v' thay cho 'z'

F = ZTRANS(f,'v','x') giaí thiăút f lài hàm theo biăún
'x' thay cho 'n'

F = ZTRANS, khăng ääúi säú, biăún ääøi kăút quái trăăïc
Vê duê:

```

ztrans 1                                z/(z-1)
ztrans a^n                              z/(z-a)
ztrans sin(n*pi/2)                      z/(1+z^2)
ztrans('x^k/k!','z','k')                exp(1/z*x)
ztrans('f(n+1)')
z*ztrans(f(n),n,z)-f(0)*z

```

TAÌI LIỆU THAM KHẢO

- [1] **USER'S GUIDE** - MATHWORKS
- [2] WWW.MATHWORKS.COM
- [3] **ĂN Ö HOÛA VỚI MATLAB** - ÂÂÛNG MINH HOÀNG
- [4] **CẢ SÁI MATLAB & ĂÏNG DUÊNG** - NGUYỄN HỮU TÇNH

GIỚI THIỆU	1
Hướng dẫn cài đặt MATLAB 7.0	2
Chương 1. CÁC KHÁI NIỆM CƠ BẢN	11
1.1. Nhập ma trận đơn giản	11
1.2. Các phần tử của ma trận	12
1.3. Câu lệnh và biến	13
1.4. Cách lấy thông tin vùng làm việc	14
1.5. Số và biểu thức số	15
1.6. Số phức và ma trận phức	16
1.7. Dạng thức xuất	17
1.8. Công cụ trợ giúp	19
1.9. Thoát và lưu vùng làm việc	19
1.10. Các hàm	20
Chương 2. CÁC PHÉP TOÁN TRÊN MA TRẬN	22
2.1. Chuyển vị ma trận	22
2.2. Cộng và trừ ma trận	23
2.3. Nhân ma trận	23
2.4. Chia ma trận	25
2.5. Lũy thừa ma trận	26
2.6. Các hàm sơ cấp về ma trận	26
Chương 3. CÁC PHÉP TOÁN TRÊN MẢNG	28
3.1. Cộng và trừ trên mảng	28
3.2. Nhân và chia trên mảng	28
3.3. Lũy thừa trên mảng	28
3.4. Phép toán quan hệ	29
3.5. Phép toán logic	31
3.6. Các hàm toán sơ cấp	32
3.7. Các hàm toán học đặc biệt	34
Chương 4. THAO TÁC TRÊN VECTƠ VÀ MA TRẬN	35
4.1. Cách phát sinh vectơ	35
4.2. Mô tả chỉ số	37
4.3. Mô tả chỉ số bằng vectơ 0-1	39
4.4. Ma trận rỗng	40
4.5. Ma trận đặc biệt	40
4.6. Cách tạo ra ma trận lớn	42
4.7. Thực hiện trên ma trận	43
Chương 5. THAO TÁC TRÊN VECTƠ VÀ MA TRẬN	45
5.1. Phân tích theo hướng cột	45
5.2. Các giá trị bỏ qua	48
5.3. Cách xóa các giá trị quá hạn	50
5.4. Hồi quy và đường cong thực nghiệm	50
Chương 6. HÀM MA TRẬN	53
6.1. Thừa số tam giác	53
6.2. Thừa số trực giao	56
6.3. Tách giá trị kỳ dị	58
6.4. Giá trị riêng	58
6.5. Hạng và điều kiện	59
Chương 7. ĐA THỨC VÀ XỬ LÝ TÍN HIỆU	61
7.1. Đa thức	61
7.2. Xử lý tín hiệu	62
7.3. Lọc dữ liệu	63
7.4. FFT(Fast Fourier Transform-Biến đổi Fourier nhanh)	64
Chương 8. HÀM CÓ ĐỐI SỐ LÀ HÀM	67

8.1.	Tích phân số.....	67
8.2.	Phương trình và tối ưu phi tuyến	68
8.3.	Phương trình vi phân.....	69
Chương 9.	ĐỒ THỊ	72
9.1.	Hình vẽ trong mặt phẳng x-y	73
9.2.	Dạng thức cơ bản	73
9.3.	Nhiều đường	75
9.4.	Kiểu đường và kiểu điểm.....	76
9.4.1.	Kiểu.....	76
9.4.2.	Màu	77
9.5.	Dữ liệu ảo và phức	77
9.6.	Hình vẽ loga, cực, và biểu đồ	77
9.7.	Vẽ mặt lưới 3 chiều và đường mức.....	78
9.8.	Điều khiển màn hình.....	80
9.9.	Cách chia đơn vị trục tọa độ	82
9.10.	Bản sao phần cứng.....	82
Chương 10.	ĐIỀU KHIỂN LUỒNG	83
10.1.	Vòng lặp FOR.....	83
10.2.	Vòng lặp WHILE.....	85
10.3.	Các lệnh IF và BREAK.....	87
Chương 11.	SIÊU TẬP M-FILE	89
11.1.	Tập nguyên bản.....	89
11.2.	Tập hàm	91
11.3.	Các lệnh Echo, input, pause, keyboard	93
11.4.	Xâu chữ và macro xâu chữ	94
11.5.	Chương trình bên ngoài	96
11.6.	Vấn đề về tốc độ và bộ nhớ.....	97
Chương 12.	VỀ TẬP TRÊN ĐĨA	99
12.1.	Thao tác về tệp.....	99
12.2.	Chạy chương trình bên ngoài.....	99
12.3.	Nhập và xuất dữ liệu	100
PHỤ LỤC	102
	Quản lý Lệnh vai hàm.....	103
	Quản lý các biến vai vùng làm việc.....	104
	Làm việc với tập vai dữ liệu hành.....	106
	Điều khiển các cửa sổ làm việc.....	106
	Thăng tin chung.....	108
	Các hàm Logic.....	109
	Các hàm đại lượng cả biến.....	110
	Vị phân các ảnh.....	112
	Các thao tác với vectơ.....	112
	Các dữ liệu tương quan.....	113
	Loại vai tích chập.....	113
	Các phép biến đổi ảnh ngược ảnh Fourier.....	121
	Các hàm lượng giác.....	124
	Các hàm mũ vai logarit.....	126
	Các hàm phức.....	126
	Hàm với dữ liệu ngẫu nhiên vai nhiễu.....	127
	Các ma trận cả biến.....	127
	Phân tích ma trận.....	129
	Phân tích tuyến tính.....	131
	Giải trẻ riêng vai giải trẻ kỳ dị.....	133
	Các hàm ma trận.....	135

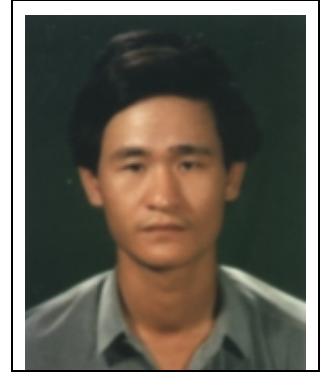
Caïc biăún vai hàòng ààúc biăút.....	136
Thăi gian vai nhăút kyì.....	139
Thao taïc trăn ma trăún.....	139
Caïc haìm coĩ àăúi săú lai haìm.....	141
MATLAB lai măút ngăn ngăi lăúp trệnh.....	146
Âiăöu khiăøn luăöng.....	147
Haìm văö âa thăic.....	151
Năúi suy săú liăûu.....	154
Năúi suy Spline.....	156
Haìm văö xăú chăi.....	156
Ăăö hoăa X-Y cả baín.....	162
Caïc lăûnh âăö thê X-Y ààúc biăút.....	164
Chuĩ giaíi trăn âăö thê.....	168
Caïc lăûnh veĩ âăêng vai tă vuòng.....	169
Veĩ âăêng măic vai caïc hệnh veĩ khaïc 2 chiăöu cuía dăi liăûu 3 chiăöu.....	171
Caïc lăûnh veĩ băö măút vai lăăi.....	173
Caích thăø hiăûn hệnh aính.....	177
Caïc àăúi tăăng 3 chiăöu.....	179
Âiăöu khiăøn mău.....	179
Caïc baíng mău.....	181
Caïc haìm baíng mău liăn quan.....	182
Caïc mă hệnh saíng.....	184
Tău căi săø hệnh aính vai caïc âiăöu khiăøn.....	185
Tău caïc truệc vai caïc âiăöu khiăøn.....	186
Caïc àăúi tăăng theĩ âăö thê.....	189
Caïc thao taïc văö theĩ âăö hoăa.....	193
Baín sao căíng vai lău trăi.....	195
Caïc phim vai hệnh aính âăng.....	198
Caïc haìm lĩnh tĩnh.....	199
Caïc ma trăún ààúc biăút.....	202
Caïc haìm ám thanh tăøng quaĩt.....	205
Caïc haìm ám thanh chỉ tiăút.....	205
Caïc haìm ààúc biăút.....	207
Măi vai âoĩng tăúp.....	212
Văo/Ra tăúp kăng đăng thăic.....	214
Năúp/xuăút tăúp coĩ đăng thăic.....	215
Vê trê tăúp.....	218
Chuyăøn âăoi chuăùi.....	220
Caïc ma trăún thăa să cúp.....	221
Chuyăøn ma trăún âăöy âúi thănh ma trăún thăa.....	225
Lăim viăúc văĩ caïc pháön tăi khaïc 0 cuía ma trăún thăa.....	227
Xem caïc ma trăún thăa.....	227
Caïc thuăút toăĩn săøp xăúp lăui.....	228
Chuăøn, săú âiăöu kiăûn, vai haúng.....	229
Caïc thao taïc trăn cáy.....	230
Caïc thao taïc lĩnh tĩnh.....	231
SYMBOLIC TOOLBOX.....	233
TÀI LIỆU THAM KHẢO.....	255

THÔNG TIN VỀ TÁC GIẢ

GIÁO TRÌNH “MATLAB”

1 Thông tin về tác giả :

- + Họ và tên : **PHAN THANH TAO**
- + Quê quán :
- + Cơ quan công tác :
KHOA CÔNG NGHỆ THÔNG TIN
Trường Đại học Bách khoa, Đại học Đà Nẵng
- + Email:



2 Phạm vi và đối tượng sử dụng :

- + Giáo trình dùng tham khảo cho các ngành
- + Có thể dùng ở các trường có đào tạo các chuyên ngành
- + Từ khóa :.
- + Yêu cầu kiến thức trước khi học môn này :