



Hướng dẫn sử dụng Scilab dành cho người mới bắt đầu

Tài liệu này là sản phẩm của Scilab Enterprises và Christine Gomez, giáo viên toán học tại Lycée Descartes (Trường Trung Học Descartes) tại Antony, Hauts-de-Seine (Pháp).  
© 2013 Scilab Enterprises. Bộ sưu tập này.

## M c l c

---

### Gi i thi u

---

V tài li u này	4
Cài t Scilab	4
Danh sách a ch	4
Các ngu n b sung	4

### Ch ng1–Làm quen v i Scilab

---

Môi tr ng chung và giao di n i u khi n	5
Các phép tính b ng s n gi n	6
Thanh trình n	7
Trình so n th o	8
C a s h a	9
Quy n lý các c a s và tùy bi n không gian làm vi c	11

### Ch ng2-L p trình

---

Các bi n s , gán giá tr và hi n th	12
Các vòng l p	16
Ki m tra	17
th 2 và 3D	18
B sung vào ma tr n và vecto	23
chính xác c a phép tính	29
Gi i các ph ng trình vi phân	30

### Chapter3– Các hàm Scilab h u d ng

---

Trong phân tích	32
Trong xác su t và th ng kê	32
Hi n th và v th	33
Ti n ích	33

## **Giới thiệu**

---

### **Về tài liệu này**

Mục đích của tài liệu này là nhằm hướng dẫn người dùng bước đầu trong quá trình khám phá các tính năng cơ bản của Scilab dành cho người chưa từng sử dụng qua phần mềm tính toán số. Những gì trình bày trong tài liệu này chỉ giúp ích cho người dùng mới giúp người dùng Scilab dễ dàng hơn.

Các phép tính, biểu đồ và ví dụ minh họa được thể hiện bằng Scilab 5.4.0. Quý vị có thể sao chép tất cả các nội dung phiên bản này.

### **Cài đặt Scilab**

Scilab là một phần mềm tính toán số miễn phí. Các hệ điều hành thích hợp cho phần mềm này là Windows, Linux và Mac OS X, quý vị có thể tải Scilab về máy tính của mình sau:

<http://www.scilab.org/>

Nếu quý vị muốn nhận thông báo về các phiên bản mới của phần mềm Scilab, quý vị có thể đăng ký nhận thông báo từ kênh thông tin của chúng tôi theo cách:

<http://lists.scilab.org/mailman/listinfo/release>

### **Danh sách thành viên**

Nhằm tôn vinh những người đóng góp cho việc phát triển Scilab, chúng tôi lập các danh sách thành viên dành riêng cho từng nhóm (danh sách thành viên người Pháp, danh sách thành viên ngành giáo dục, danh sách người dùng tiếng Anh). Quy tắc rất đơn giản: những người đăng ký có thể liên lạc với nhau thông qua email (tên câu hỏi, trả lời, chia sẻ tài liệu, phần mềm...).

xem các danh sách hiện có và đăng ký theo dõi, quý vị có thể truy cập vào các danh sách sau:

[http://www.scilab.org/communities/user\\_zone/mailling\\_list](http://www.scilab.org/communities/user_zone/mailling_list)

### **Các nguồn bổ sung**

Nếu quý vị có kết nối Internet, mời quý vị ghé thăm trang web Scilab, tại đây quý vị có thể truy cập vào chuyên mục sử dụng Scilab (<http://www.scilab.org/support/documentation>) cùng các hướng dẫn và tài liệu có liên quan cho phép quý vị tải về và in miễn phí.

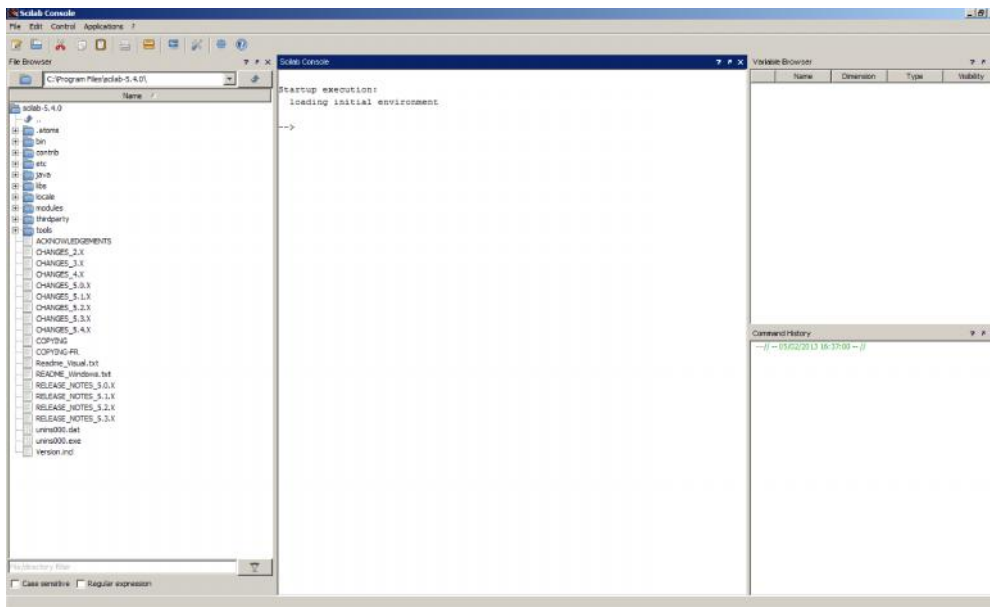
## Chương 1 – Làm quen với Scilab

Không gian làm việc chủ định trong Scilab bao gồm các chức năng sau:

- Giao diện người dùng khi cần thiết chỉ ra các phép tính,
- Trình soạn thảo lập trình,
- Các chức năng hỗ trợ hiển thị,
- Trợ giúp tích hợp

### Môi trường chung và giao diện người dùng

Sau khi nhấp chuột vào biểu tượng khởi động Scilab, môi trường Scilab mở ra bao gồm các chức năng tích hợp thành dock như sau – giao diện người dùng, các trình duyệt file và biểu thức, lịch sử lệnh (xem phần “Quy định các chức năng và tùy biến không gian làm việc”, trang 11):



Trong giao diện người dùng sau dấu ch “-->”, nhấn dòng lệnh và nhấn phím Enter (Windows và Linux) hoặc phím Return (Mac OS X) trên bàn phím nhấn nút quét ngôn ngữ.

```
--> 57/4
```

```
ans =
```

```
14.25
```

```
--> (2+9)^5
```

```
ans =
```

```
161051.
```

**M**o

Trên nút quét, **ans** có nghĩa là “ấn”.

Ngay dùng có thể quay lại bất kỳ lúc nào bằng các phím mũi tên trên bàn phím hoặc bằng chuột. Các phím trái và phải của dòng thay thế phím chức năng còn các phím lên và xuống của dòng quay về một dòng lệnh để chỉ ra trình duyệt.

## Các phép tính số nguyên

Tất cả các phép tính thể hiện bởi Scilab sử dụng số nguyên. Scilab thể hiện các phép tính bằng ma trận (xem chương 2, trang 23).

Các thao tác cơ bản trình bày ví dụ "+" cho phép cộng, "-" cho phép trừ, "\*" cho phép nhân, "/" cho phép chia, "^" cho số mũ. Ví dụ:

```
-->2+3.4
ans =
    5.4
```

Phần này có phân biệt viết hoa và viết thường. Do đó cần chú ý phân biệt viết hoa và viết thường các phép tính để thể hiện chính xác. Ví dụ, ví dụ như **sqrt** (tính căn bậc hai):

```
-->sqrt(9)           trong khi đó:           -->SQRT(9)
ans =                --error 4
    3.                Bị số không xác định: SQRT
```

## Các hằng số

**%e** và **%pi** là hằng số cho  $e$  và  $\pi$ :

```
--> %e                --> %pi
%e =                  %pi =
    2.7182818         3.1415927
```

**%i** là hằng số cho  $i$  trong các phép tính khi nhập và thể hiện thì là  $i$  trong kết quả:

```
--> 2+3*%i
ans =
    2. + 3.i
```

## Viết các kết quả không hiển thị

Khi thêm dấu chấm phẩy ";" vào cuối dòng lệnh, phép tính thể hiện nhưng kết quả không hiển thị.

```
-->(1+sqrt(5))/2;           --> (1+sqrt(5))/2
ans =
    1.618034
```

### nhắc tên các hàm

Tên các hàm thường có dạng `ctngktti` Chương 3 của tài liệu (trang 32).

Ví dụ :

```
--> exp(10)/factorial(10)
ans =
    0.0060699
```

**M o**  
Tất cả các hàm kh  
dùng cú pháp  
trong phần trợ  
giúp tích hợp  
nhập chuỗi vào  
thành trình  
trên ?>  
**Scilab Help.**

Có thể dùng phím tab trên bàn phím hoàn tất tên các hàm hay mã lệnh bằng cách nhập một số cái đầu tiên:

Ví dụ , sau khi nhập vào giao diện khi như:

```
-->fact
```

Và nhấn phím tab, mã lệnh sẽ hiện tất cả các tên hàm và mã lệnh bắt đầu bằng **fact**, chúng hiện như **factorial** và **factor** sẽ hiện ra. Chọn nhập chuỗi vào hàm cần dùng hoặc chọn hàm có bảng chuỗi hoặc bảng các phím và nhấn Enter (Windows và Linux) hoặc Return (Mac OS X) chèn hàm vào dòng lệnh.

### Thanh trình

Các trình liệt kê dưới đây có thể dùng:

#### Applications

- Lệnh `tl` cho phép tìm kiếm các lệnh các phiên trình có sẵn hiện tại.
- Trình duy trì mã lệnh cho phép tìm kiếm các mã lệnh có sẵn trình có sẵn trong phiên hiện tại.

#### Edit

**Preferences** (trong trình **Scilab** dành cho Mac OS X) cho phép chỉnh sửa và tùy chỉnh màu sắc, phong cách và các chức năng trong giao diện khi trình soạn thảo, ví dụ như rút xuống dòng khi phích lên màn hình.

Nhập chuỗi vào **Clear Console** sẽ xóa toàn bộ nội dung trong giao diện khi trình. Trong trình này, lệnh `tl` hiện còn và các phép tính có thể hiển thị trong phiên văn bản trong bảng. Các lệnh xóa và có thể liệt kê các phím mã lệnh trên bàn phím.

#### Control

làm gián đoạn mã lệnh trình đang chạy, bạn có thể làm như sau:

- Nhấn phím **pause** vào chế độ trình hoặc nhập chuỗi vào **Control>Interrupt** trong thanh trình (Ctrl X với Windows và Linux hoặc Command X với Mac OS X), nếu chế độ trình đang chạy. Trong môi trường nhập, dấu nháy "-->" sẽ chuyển thành "-1->", sau đó thành "-2->" ..., nếu thao tác tiếp theo.
- quay về vị trí mã lệnh khi làm gián đoạn trình, nhấn phím **resume** vào giao diện khi trình hoặc nhập chuỗi vào **Control>Resume**.
- thoát ra hoàn toàn mã lệnh tính và không quay lại, nhấn phím **abort** trong giao diện khi trình hoặc nhập chuỗi vào **Control>Abort** trên thanh trình.

## Trình soạn thảo

Vì cách trình bày vào giao diện khi có hai lựa chọn: thứ nhất là không thể gõ các lệnh và thứ hai là vì các soạn thảo dòng lệnh là không hiệu quả. Trình soạn thảo là công cụ thích hợp để thực hiện cùng lúc.

### Mở trình soạn thảo

Mở trình soạn thảo tại giao diện khi nhấp chuột vào biểu tượng ưu tiên trên thanh công cụ hoặc vào menu **Applications > SciNotes** trên thanh trình đơn.

Trình soạn thảo sẽ đặt tên file mặc định là **"Untitled 1"**.

### Soạn thảo trong trình soạn thảo

Soạn thảo trong trình soạn thảo có thể gõ lệnh và nhập công thức lý văn bản khác.

Trong trình soạn thảo văn bản, chúng ta nhập và gõ các dấu ngoặc, kết thúc các lệnh vòng lặp, hàm và kiểm tra kết quả thêm vào. Tuy nhiên, có thể tắt các tính năng này trong menu **Options > Auto completion on** trình đơn, nhấp chuột vào hai menu kích hoạt menu để tắt:

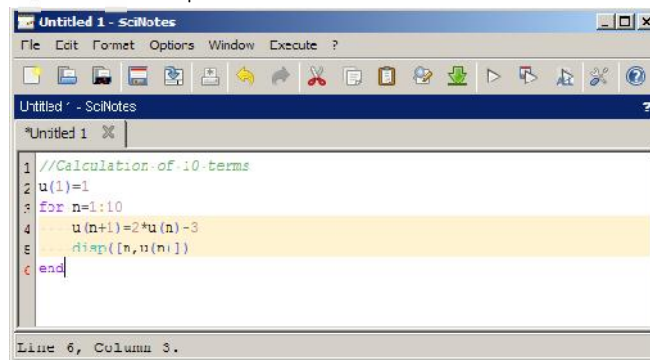
- **([,...**
- **if,function,...**

Vấn đề nguyên tắc, mỗi lệnh nhập sẽ nhập vào một dòng riêng biệt, tuy nhiên, vẫn có thể nhập nhiều câu trên cùng một dòng lệnh, tách biệt nhau bằng dấu chấm phẩy ";".

Kiểm tra ngữ pháp gõ lệnh là dấu là tắt khi một vòng lặp hay phép kiểm tra khi kết thúc.

Trong ví dụ sau đây, chúng tôi tính 10 số hạng của chuỗi  $(u_n)$  xác định bởi:

$$\begin{cases} u_1 = 1 \\ u_{n+1} = 2u_n - 3 \end{cases}$$



```
Untitled 1 - SciNotes
File Edit Format Options Window Execute ?
Untitled - SciNotes
*Untitled 1 X
1 //Calculation of 10 terms
2 u(1)=1
3 for n=1:10
4     u(n+1)=2*u(n)-3
5     disp([n,u(n)])
6 end
Line 6, Column 3.
```

### Mẹo

- Các dòng chú thích bắt đầu bằng "//" sẽ không được tính trong các phép tính;
- Thay đổi phong cách, nhấp chuột vào **Options > Preferences**.
- Khi lập trình, dòng lệnh có thể gõ thẳng vào. Nếu dòng lệnh không thể gõ thẳng vào, nhấp chuột vào **Format > Correct indentation** để khôi phục (Ctrl I trong Windows và Linux hoặc Command I trong Mac OS X).



## L u

B n có th l u b t k file nào b ng cách nh p chu t vào **File>Save as**.

uôi “.sce” cu i tên file s t ng kh i ng Scilab khi c m (tr h i u hành Linux và Mac OS X).

Chép vào giao di n i u khi n, ch y m t ch ng trình.

Khi nh p chu t vào m c Execute trên thanh trình n, s có ba l a ch n nh sau:

- Th c hi n “...file with no echo” (Ctrl Shift E v i Windows và Linux, Cmd Shift E v i Mac OS X): file s c ch y mà không c n l p trình trong giao di n i u khi n (b t bu c ph i l u file tr c).
- Th c hi n “... file with echo” (Ctrl L v i Windows và Linux, Cmd L v i Mac OS X): chép l i file vào giao di n i u khi n và ch y file.
- Th c hi n “...until the caret, with echo” (Ctrl E v i Windows và Linux, Cmd E v i Mac OS X): chép l i l a ch n c ch n b ng chu t vào giao di n i u khi n và ch y l a ch n ho c ch y d li u file cho n khi ng i d ùng xác nh v trí d u nháy.

Có th s d ng ch c n ng copy/paste thông th ng.

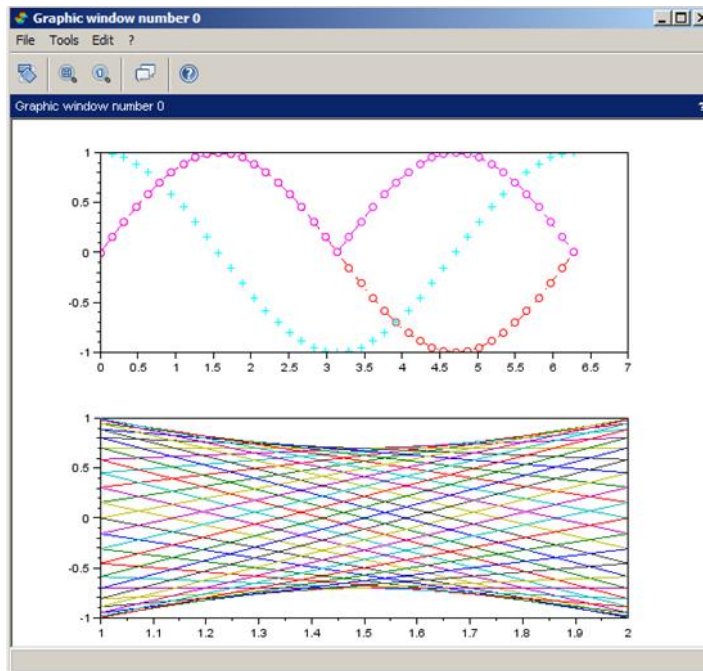
## C a s h a

### M m t c a s h a

C a s h a s t ng m khi v m t bi u b t k . Có th v các ng cong, m t ph ng, các chu i i m (xem ch ng 2, trang 18).

có ví d v ng cong, nh p vào giao di n i u khi n l nh:

```
-->plot
```





### M o


- xóa bi u tr c ó, nh p l nh **clf** (“xóa s ”).

- m m t c a s h a khác, nh p l nh **scf**; (“l p bi u ”).

N u m cùng lúc m t vài c a s h a, có th l a ch n c a s mu n v bi u b ng cách nh p l nh **scf(n)**; n là s c a s h a (n m phía trên bên trái).

## Chỉ định trục

Chỉ định trục kính lúp  dùng phóng to thu nhỏ. Phóng to ra các trục, nhấp chuột vào công cụ và dùng chuột để chỉnh hình chữ nhật ở góc xem camera. Phóng to ra ba trục, nhấp chuột vào công cụ và để hình chữ nhật ở góc xem camera. Bạn cũng có thể phóng to bằng con lăn của chuột. Trên màn hình bàn phím, nhấp chuột vào chỉ định trục kính lúp khác .

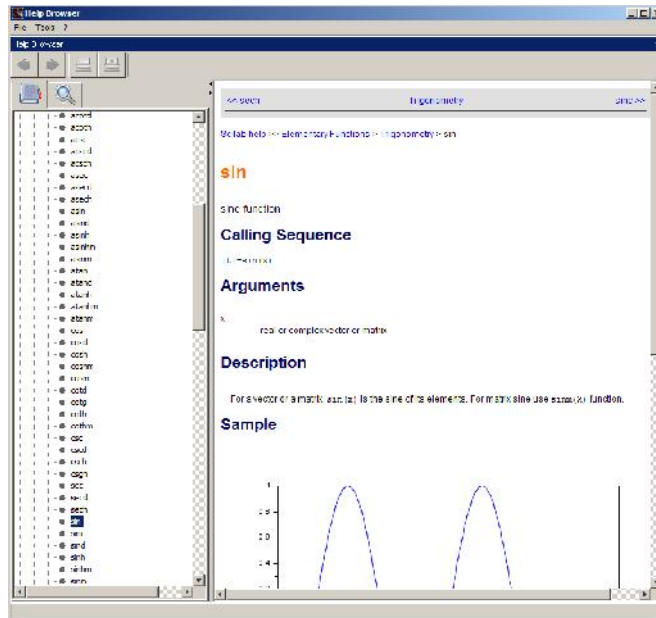
Chỉ định trục  cho phép xoay hình (các biểu thức và hình 3D) bằng thao tác nhấp chuột phía trên hình ảnh trong phần nhìn phía đầu của cửa sổ.

Chỉ định trục chính xác hơn, nhấp chuột vào **Edit > Figure properties** hoặc **Axes properties** và xem hình ảnh (lưu ý rằng cách này chỉ hoạt động trong Mac OS X).

## Hỗ trợ trợ giúp

Để truy cập trợ giúp, nhấp chuột vào **? > Scilab Help** trên thanh trình đơn, hoặc nhập lệnh sau vào giao diện người dùng:

```
-->help
```



**Mô**  
Các ví dụ và sơ đồ có thể truy cập trên Scilab và SciNotes bằng cách sử dụng các phím kèm theo trong khung mô tả.

Để truy cập vào bất kỳ hàm nào, nhập lệnh help vào giao diện người dùng, sau đó nhập tên hàm tương ứng. Ví dụ:

```
-->help sin
```

hiện thông tin trợ giúp cho hàm **sin** (sine).

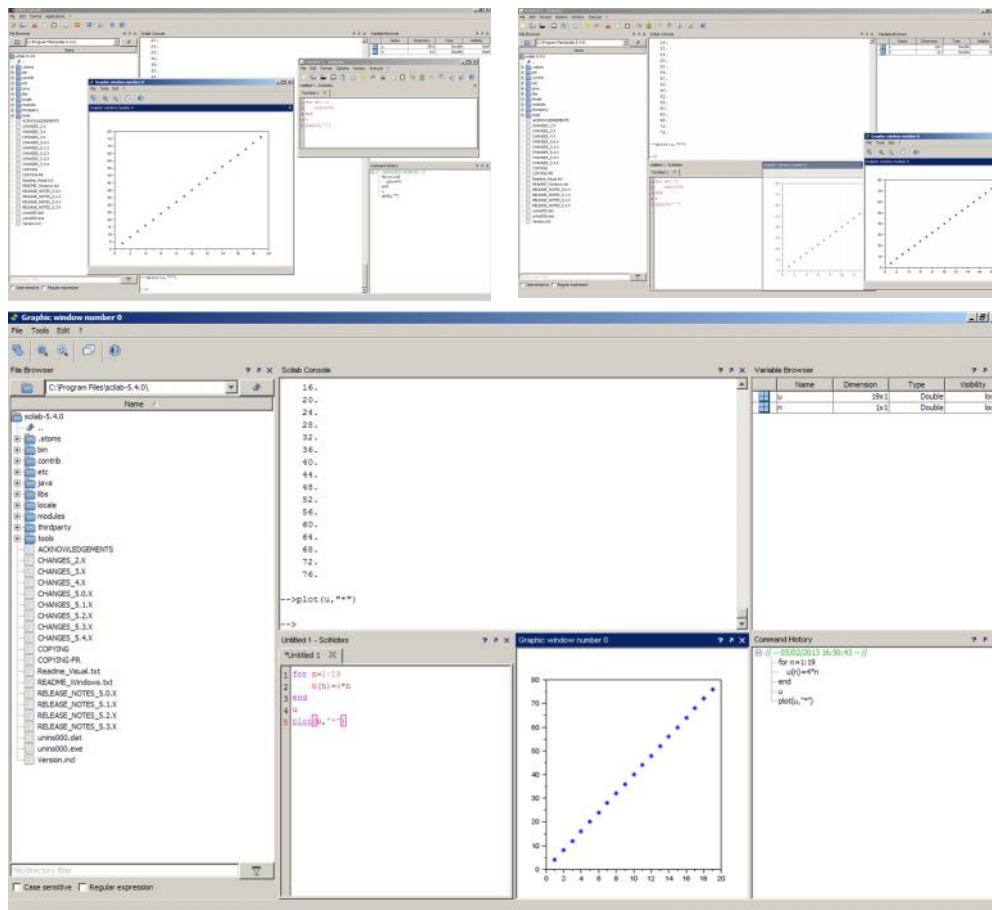
## Quy n lý các c a s và tùy ch nh không gian làm vi c

Trong môi tr ng Scilab m c nh, giao di n i u khi n, các trình duy t file và bi n s và l ch s t l nh u là các c a s đ ng dock, t t c các c a s khác trong Scilab u có th chuy n thành v trí riêng r . Ví d , ng i dùng có th ch n v trí cho trình so n th o trong môi tr ng m c nh c a Scilab.

g n m t c a s vào m t c a s khác, u tiên ph i xác nh thanh ngang màu xanh trong Window hay màu en trong Mac OS X và Linux, trên u c a s trong thanh công c có d u h i góc ph i.

- Trong Windows và Linux, nh p chu t vào thanh này b ng chu t trái và trong khi v n gi chu t trái, di chuy n con tr chu t vào c a s mong mu n.
- Trong Mac OS X, nh p chu t vào thanh này và trong khi v n gi chu t, chuy n thanh này vào c a s mong mu n.

M t hình ch nh t s xu t hi n, cho bi t v trí ti p theo c a c a s . Khi ã di chuy n n v trí mà b n mu n, th ph m b m chu t. h y thao tác và chuy n c a s ra, nh p chu t vào m i tên nh g c ph i c a thanh này.



## Chương 2 Lập trình

---

Trong các ví dụ đưa ra trong tài liệu này, bất cứ dòng nào bắt đầu bằng "-->" là một lệnh, các dòng khác là phần hiển thị các dòng lệnh (kết quả phép tính, tin nhắn báo lỗi...). Không cần ghi "-->" trong trình soạn thảo. Các ví dụ đưa ra đây là nhằm phân biệt giữa các dòng lệnh và kết quả tính toán hiển thị trong giao diện khi nhập sau khi sao chép/dán. Khi hiển thị trong bảng (không có "-->") và kết quả tính toán, các lệnh có một chính xác như khi nhập trong trình soạn thảo.

### Các biến số, gán giá trị và hiển thị

#### Các biến số

Scilab không phải là một thiết bị trên máy tính. Phần mềm này chỉ tính toán bằng các chuỗi. Mọi phép tính được thực hiện bằng ma trận, mặc dù đôi khi có thể không cần chú ý. Ngay cả khi khái niệm ma trận không rõ ràng, thì vectơ và các chuỗi ký tự cũng có thể giải thích khái niệm này, mà trên thực tế, là ma trận có chiều  $1 \times n$  hoặc  $n \times 1$  và một chuỗi là ma trận có chiều  $1 \times 1$ .

Các biến số không cần các biểu thức, nhưng mọi biến số đều phải có giá trị. Ví dụ, tính giá trị của một biến số chưa cho giá trị sẽ hiển thị:

```
-->a
```

```
! --error 4
```

Biến số chưa xác định: a

Nếu một giá trị được gán cho biến số a, lệnh này sẽ không còn nữa:

```
--> a=%pi/4
```

```
a =
```

```
0.7853982
```

```
--> a
```

```
a =
```

```
0.7853982
```

Các biến số có thể có tên bất kỳ không do họ thống nhất:

```
--> Piby2=%pi/2
```

```
Piby2 =
```

```
1.5707963
```

#### M o

Trên tất cả các phiên bản của Scilab, tên một biến số không thể chứa dấu hay ký tự đặc biệt.

Kết quả một phép tính không cần gán biến số thì sẽ gán cho biến số **ans**:

```
-->3*(4-2)
```

```
ans =
```

```
6.
```

```
-->ans
```

```
ans =
```

```
6.
```

### Các hàm

Hàm là một phép toán dạng công thức thông thường thể hiện các phép tính từ các biến và thu được kết quả từ các biến.

Xác định một hàm bất kỳ với **function** và kết thúc bằng **endfunction**. Ví dụ, để tính euro (e) sang đô la (d) theo tỷ giá hối đoái là (t), hàm **dollars** sẽ xác định. Các biến số là **e** và **t** và nh là **d**.

```
-->function d=dollars(e,t); d=e*t; endfunction
```

```
-->dollars(200,1.4)
```

```
ans =
```

```
280.
```

Một số hàm sẵn có là các hàm có một biến số. Ví dụ, hai hàm **f** và **g** sẽ xác định bằng các lệnh sau:

```
-->function y=f(x); y=36/(8+exp(-x)); endfunction
```

```
-->function y=g(x); y=4*x/9+4; endfunction
```

Các hàm xác định có thể sử dụng tính các giá trị:

```
--> f(10)
```

```
ans =
```

```
4.4999745
```

```
--> g(12.5)
```

```
ans =
```

```
9.5555556
```

**M o**  
Các biến số **x** và **y** là các biến số, tên các biến số này có thể tái sử dụng khi xác định các hàm khác trong Scilab.

### Yêu cầu gán giá trị cho biến

Việc gán giá trị cho biến khá đơn giản bằng cách sử dụng toán tử "=".

### Hiển thị

Số nguyên

Việc số nguyên biến sẽ hiển thị giá trị của biến đó, trừ khi có dấu ";" cuối dòng lệnh.

Dùng

Các ma trận xác định bằng dấu ngoặc vuông (xem trang 23). Như đã nêu trên, việc tính toán bằng ma trận là nền tảng cho các phép tính trong Scilab. Ký tự trống hay dấu phẩy để chia dòng và dấu chấm phẩy để dùng để chia các hàng.

xác định vectơ cột và hiển thị theo cột:

```
-->v=[3;-2;5]
```

```
v =
```

```
3.
```

```
-2.
```

```
5.
```

xác định vectơ hàng và hiển thị theo hàng:

```
-->v=[3,-2,5]
```

```
v =
```

```
3. -2. 5.
```

**Mô**

Lệnh này có thể sử dụng để hiển thị:

```
v=[3 -2 5]
```

xác định ma trận và hiển thị theo bảng:

```
-->m=[1 2 3;4 5 6;7 8 9]
```

```
m =
```

```
1. 2. 3.
```

```
4. 5. 6.
```

```
7. 8. 9.
```

**Mô** Lệnh này có thể sử dụng để hiển thị:

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

Hàm disp

**disp** bu c ph i s d ng v i d u ngo c.

V i vecto **v** c xác nh tr c ó:

```
-->v(2)
```

```
ans =
```

```
- 2.
```

```
-->disp(v(2))
```

```
- 2.
```

hi n th m t chu i (th ñ ng là m t m nh ), t chu i ó trong d u ngo c:

```
-->disp("Bob won")
```

```
Bob won
```

hi n th m t t h p t và giá tr , s d ñ ng l nh **string** cho phép chuy n i các giá tr sang các chu i ký t b ñ g d u "+" g i a các ph n khác nhau:

```
-->d=500;
```

```
-->disp("Bob won "+string(d)+" dollars")
```

```
Bob won 500 dollars
```

N u m nh ch có m t d u ngo c, thì c n ph i thêm m t d u ngo c n a trong chu i chu i c hi n th úng.

```
-->disp("It's fair")
```

```
It's fair
```

### Các vòng l p

#### Gia l ñ g

Toán t ":" cho phép xác nh các vecto c a các ch s có t a theo th t s h c. Ta a ra: << beginning value: step: ending value>>. Giá tr cu i "ending value" có th s không t c. N u b c trên không c c p, thì giá tr m c nh s là 1.

Ví d , xác nh m t vecto hàng c a các s nguyên t ñ g theo c p l t 3 ñ n 10:

```
-->3:10
```

```
ans =
```

```
3. 4. 5. 6. 7. 8. 9. 10.
```

Ho c t ñ g theo c p 2 t 1 ñ n 10:

```
-->1:2:10
```

```
ans =
```

```
1. 3. 5. 7. 9.
```

Học gì mới theo công thức 20 n 2:

-->u=20:-4:2

u =

20. 16. 12. 8. 4.

### For

Cấu trúc vòng lặp để tính tích lũy các số nguyên **for ... end**.

Ví dụ: Tính 20 số hạng của một chuỗi các xác định bằng phép truy toán: 
$$\begin{cases} u_1 = 4 \\ u_{n+1} = u_n + 2n + 3 \end{cases}$$

Thuật toán	Trình số nguồn Scilab
Thay 4 vào u(1)	u(1)=4;
Lệnh for n từ 1 đến 20	for n=1:20
u(n+1) mang giá trị u(n)+2n+3	u(n+1)=u(n)+2*n+3;
và u(n)	disp([n,u(n)])
Hết vòng lặp và u(n)	end
Kết thúc lệnh for	

### While

Đang làm một vòng lặp khi đạt đến một tiêu chuẩn, ta sử dụng **while ... end**.

Vào năm 2005 tôi đã tính chiều cao của cây thông Noel là 1,20m. Mỗi năm cây thông này lại cao thêm 30cm. Tôi quyết định chặt cây khi cây cao hơn 7m. Vậy tôi sẽ chặt cây vào năm nào?

Thuật toán	Trình số nguồn Scilab
Thay 1.2 vào h (h = chiều cao của cây)	h=1.2;
Thay 2005 vào y (y = năm)	y=2005;
Lệnh while h<7	while h<7
h mang giá trị h+0.3 (cây cao lên mỗi năm)	h=h+0.3;
y mang giá trị y+1 (mỗi năm qua)	y=y+1;
Kết thúc lệnh while	end
Hết vòng lặp (n năm cuối cùng)	disp('I will cut the.. tree in '+string(y))

### Mở

Khi có một lệnh quá dài không thể ghi trên một dòng, trình số nguồn sẽ chèn các dòng bằng ".." (hai dấu chấm) trước khi bắt đầu dòng tiếp theo.



## Kiểm tra

### Các toán tử so sánh

Các phép kiểm tra hình thức bao gồm các chuỗi ký hiệu hoặc xác định mệnh đề logic là đúng hay sai. Bên dưới là các logic ngữ nghĩa:

Bằng	Khác	Nhỏ hơn	Lớn hơn	Nhỏ hơn hoặc bằng	Lớn hơn hoặc bằng
==	<>	<	>	<=	>=
Đúng	Sai	Và	Hoặc	Phiên	
%T	%F	&		~	

### Mô

Hãy cẩn trọng với chính xác của phép tính. Các phép tính có thể hiển thị và "==" đôi khi sẽ cho kết quả sai (Xem chính xác của phép tính, trang 30).

So sánh hai vectơ (hoặc hai ma trận), phép kiểm tra "==" và "<>" sẽ so sánh từng số hạng.  
Ví dụ:

```
-->X=[1,2,5]; Y=[5,3,5];
```

```
-->X==Y
```

```
ans =
```

```
F F T
```

kiểm tra xem hai vectơ có bằng nhau hay không, sử dụng **isequal** và **~isequal** nếu chúng khác nhau:

```
-->isequal(X,Y)
```

```
ans =
```

```
F
```

```
-->~isequal(X,Y)
```

```
ans =
```

```
T
```

### If...then

Các câu lệnh điều kiện như sau:

- **if ... then ... else ... end,**
- **if ... then ... elseif ... then ... else ... end.**

**if ... then** phải có như trên cùng một dòng và tiếp theo với **elseif ... then**.

Ví dụ : Alice tung ba cái xúc xắc

- Nếu ba số là 1, 2, 3, cô bé sẽ có \$20,
- Nếu ba số là 1, 2, 3, 4, 5, 6, cô bé sẽ có \$10,
- Nếu có hai xúc xắc 1, 2, 3, 4, 5, 6, cô bé sẽ có \$5,
- Nếu không cô bé sẽ không có gì cả.

Mô phỏng một phép thử và tính số tiền Alice có bằng các hàm:

- **grand** (xem trang 22),
- **unique(D)** cho giá trị duy nhất của các giá trị xuất hiện trong **D**,
- **length(unique(D))** cho chiều dài của vectơ thu được, là 1 nếu ba xúc xắc 1, 2, 3, 4, 5, 6, là 2 nếu có 2 xúc xắc giống nhau.

Thuật toán	Trình số nguồn Scilab
Thay ba giá trị vào D	D=grand(1,3, "uin", 1,6);
Nếu lần đầu Alice tung xúc xắc ba là 6, thì Alice có 20 đô la.	if D==[6,6,6] then W=20;
Nếu cô bé tung xúc xắc ba giá trị giống nhau, thì Alice có 10 đô la	elseif length(unique(D))==1 then W=10;
Nếu cô bé tung xúc xắc hai giá trị giống nhau, thì Alice có 5 đô la	elseif length (unique(D))==2 then W=5;
Nếu không Alice không có đô la nào cả	else W=0;
Kết thúc lệnh if	end
Hiển thị số tiền Alice có	disp( Alice wins +. string(W)+ dollars )

### th 2D và 3D

Lệnh **plot** vẽ đồ thị 2D và 3D trên mặt phẳng. Màu sắc và dạng biểu đồ có thể xác định bằng cách chèn các dấu hiệu màu và kiểu chữ vào trong dấu ngoặc:

- Màu sắc

“b” = xanh dương (mặc định), “k” = đen, “r” = đỏ, “g” = xanh lá cây, “c” = màu lục lam, “m” = tím,

“y” = vàng, “w” = trắng.

- Kiểu chữ in

Nhãn trục (mặc định), hoặc “.”, “+”, “o”, “x”, “\*”.

Các nhãn khác: “s”, “d”, “v”, “<”, và “>”.

### Các biểu đồ

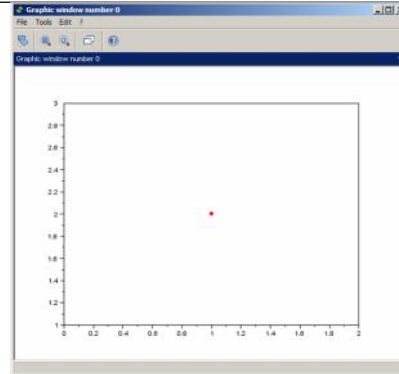
điểm

Vẽ điểm A(1;2) bằng lệnh

Trình soạn thảo Scilab

Cas h a

`plot(1,2, "r")`



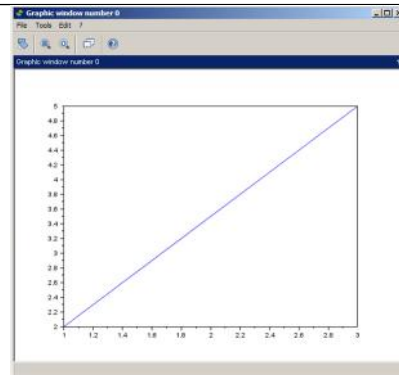
đường

Vẽ đường thẳng [AB] màu xanh dương (màu xanh) với A(1;2) và B(3;5).

Trình soạn thảo Scilab

Cas h a

`plot([1,3],[2,5])`



Biểu đồ các đường cong trên mặt phẳng xác định bằng hàm  $y=f(x)$

Điểm vẽ hàm  $x$ - ( $x$ ) các giá trị của  $x$  xác định bằng lệnh **linspace** bằng cách nhập:  **$x=linspace(a,b,n)$** ; trong đó  $a$  là giá trị nhỏ nhất của biến  $x$ ,  $b$  là giá trị cao nhất của biến  $x$ , và  $n$  là số lượng giá trị tính giữa  $a$  và  $b$ .

Không quên dấu ";" nếu không thể các giá trị của  $x$  sẽ hiển thị.

Ví dụ, xét hai hàm  $f$  và  $g$  được xác định qua khoảng  $[-2; 5]$  bởi hàm:

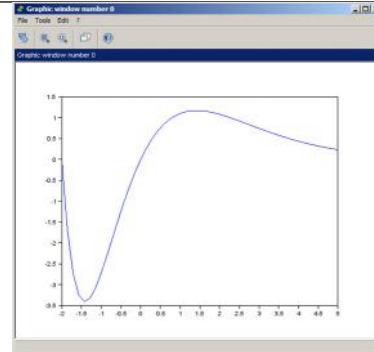
$$f(x) = (x^2 + 2x)e^x, \text{ và } g(x) = \sin\left(\frac{x}{2}\right)$$

Vì chúng ta chưa biết công thức tính  $f$ , mặt khác màu xanh đậm:

Trình soạn thảo Scilab

Cas h a

```
function y=f(x)
    y=(x^2+2*x)*exp(-x)
endfunction
x=linspace(-2,5,50);
plot(x,f)
```

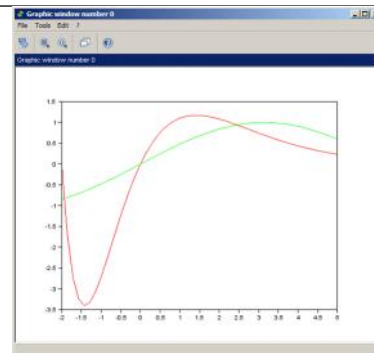


Bằng cách bổ sung thêm chương trình để vẽ hai đường cong, một màu đỏ và một màu xanh lá cây. Bạn có thể xóa bỏ lệnh `clf` (“xóa bảng vẽ”).

Trình soạn thảo Scilab

Cas h a

```
function y=g(x)
    y=sin(x/2)
endfunction
x=linspace(-2,5,50);
clf
plot(x,f, 'r', x, g)
```



### Biểu chú ý

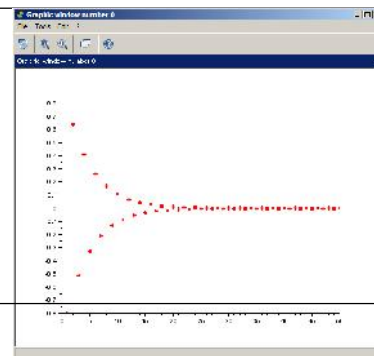
Các hệ thống hàm tuần hoàn

Trong hình vẽ trên, các chú ý là các chú ý  $(n, u(n))$  sau khi tính các thành phần  $u(n)$  của vectơ  $u$ . `plot(u, '*r')` xác định đường và màu sắc của các điểm trong đồ thị: màu đỏ và chỉ định hình sao. Theo mặc định, các điểm vẽ bằng màu xanh đậm và cỡ chữ nhỏ.

Trình soạn thảo Scilab

Cas h a

```
for n=1:50
    u(n)=(-0.8)^n;
end
clf; plot(u, '*r')
```



D li u th ng kê hai bi n s

D li u th ng kê hai bi n s c cho d i d ng hai vecto: g i hai vecto này là X và Y.  
**plot(X,Y, “<”)** s t o bi u phân tán c a i m  $M(X_i; Y_i)$  v i các hình tam giác màu xanh d ùng.

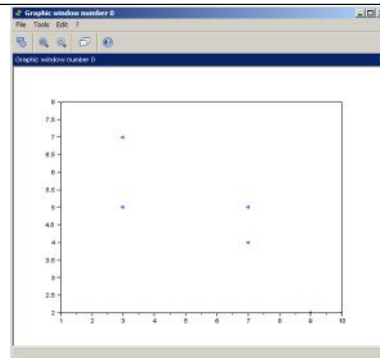
Trình so n th o Scilab

C a s h a

```
X=[1,3,3,7,7,9,10];
```

```
Y=[8,7,5,5,4,2,2];
```

```
clf; plot(X,Y, “<”)
```



### B i u b a chi u

Scilab có th c s d ng v các m t ph ng và ng cong trong không gian, cùng nhi u tùy ch n x lý các m t n, màu s c các m t, các i m quan sát c... Các ví d, sau ây s minh h a cho bi u 3-D

Hàm **surf** có th dùng v i các mặt ph ng. Hàm này có ba bi n s u vào **x**, **y** và **z**. **x** và **y** là các vecto lần l i t có l n **m** và **n** tương ng v i các i m trên các tr c ( $Ox$ ) và ( $Oy$ ). **z** là ma tr n có chi u  $n \times m$  v i thành t z tương ứng v i chi u cao của đi m v i t a l i tr c  $X x_i$  và t a tr c  $Y y_j$ .

v m t m t ph ng xác định b ng hàm có d ng  $z = f(x, y)$ , ta ph i:

- Xác nh hàm  $f$
- Tính  $z = \text{feval}(x, y, f)$   
 $\text{feval}(x, y, f)$  cho ma tr  $n \times m$  có  $ij$  là  $f(x_i, y_j)$  c i b ng cách s d ng bi u t ng ngo c n “ ”
- Ch y hàm **surf(x,y,z)**.

v m t ph ng =  $2x^2 + y^2$  (m t parabol hình elip):

Trình so n th o Scilab

C a s h a

```
function z=f(x,y)
```

```
z=2*x^2+y^2;
```

```
endfunction
```

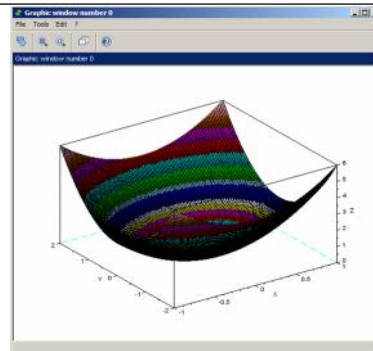
```
x=linspace(-1,1,100);
```

```
y=linspace(-2,2,200);
```

```
z=feval(x,y,f);
```

```
clf
```

```
surf(x,y,z)
```



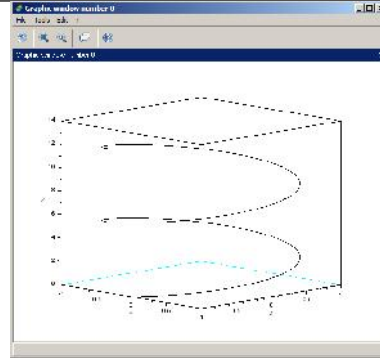
Các đường cong trong không gian có thể biểu diễn bằng hàm **param3d**. **param3d** có ba biến số  $x, y$  và  $z$ , mỗi vectơ đều có cùng chỉ số và t là tham số trong véc-tơ các điểm  $(x_i, y_i, z_i)$  trên đường cong.

Vẽ đường tròn bằng hàm  $(x = \cos(t), y = \sin(t), z = t)$ :

Trình số nguồn Scilab

C a s h a

```
t=linspace(0,4*%pi,100);
param3d(cos(t),sin(t),t)
```



### Mô phỏng và thống kê

Scilab có một số hàm thống kê để thực hiện các phép mô phỏng một cách nhanh chóng và hiệu quả:

Các chuỗi ngẫu nhiên

- **grand(1,p,uin,m,n)** cho vectơ bao gồm các chuỗi số nguyên ngẫu nhiên  $p$  giá trị  $a$  và  $n$  với  $p$  là số nguyên dương,  $m$  và  $n$  là các số nguyên và  $m \leq n$ .

```
-->t= grand(1,4, uin ,1,6)
```

```
t =
```

```
3. 1. 3. 6.
```

- **grand(1,p,unf,a,b)** cho vectơ bao gồm các chuỗi số thực ngẫu nhiên  $p$  giá trị  $a$  và  $b$  với  $p$  là số nguyên dương,  $a$  và  $b$  là số thực và  $a \leq b$ .

```
-->tr= grand(1,2, unf ,-1,1)
```

```
tr =
```

```
- 0.7460264 0.9377355
```

## Thống kê

Tổng các hàm thống kê cơ bản liệt kê tại trang 32.

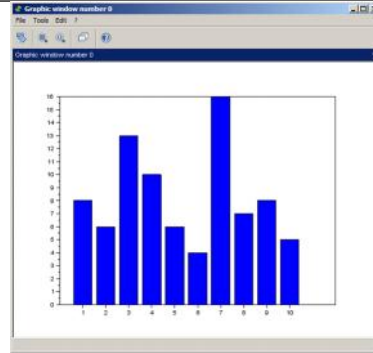
Chú ý:

- Hàm **bar(x,n,color)** vẽ biểu đồ cột:

Trình số nguồn Scilab

Cas h a

```
x=[1:10];  
n=[8,6,13,10,6,4,16,7,8,5];  
clf; bar(x,n)
```

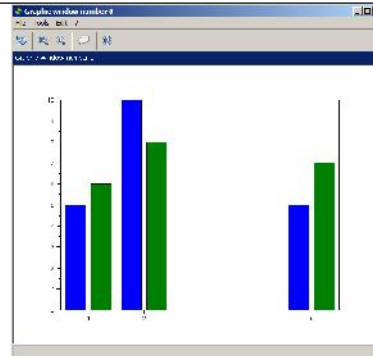


- Ív i bi u đ ng c t th hi n hai kh i c nh nhau: xét chu i các giá tr X và hai chu i các ch s n1 và n2. Khi v bi u , n1 và n2 ph i là các vecto c t và ó là lý do vì sao chuy n v c s đ ng trong ví d d i này:

Trình số nguồn Scilab

Cas h a

```
X=[1,2,5];n1=[5,10,5];n2=[6,8,7];  
bar(X,[n1',n2'])
```



Í s tu ch n **color** xác nh màu s c nh trong hàm **plot**.

### Thông tin bổ sung về ma trận và vecto

#### Nhập các phần tử

Dùng ngoặc vuông để nhập các ma trận. Ký tự ngoặc đơn để nhập các dòng chuyển từ cột này sang cột khác và dấu chấm phẩy để nhập các dòng chuyển từ dòng này sang dòng khác.

```
-->m=[1 2 3;4 5 6]
```

```
m =
```

```
1.    2.    3.  
4.    5.    6.
```

**M o**

Lưu ý này có thể  
còn áp dụng  
dùng:  
**m=[1,2,3;4,5,6]**

Dùng ngoặc đơn để nhập các phần tử hoặc sử dụng chúng.

```
-->m(2,3)
```

```
ans =
```

```
6.
```

```
-->m(2,3)=23
```

```
m =
```

```
1. 2. 3.  
4. 5. 23.
```

Toán tử “:” dùng để chỉ các hàng hoặc các cột của ma trận.

xem hàng thứ hai của ma trận **m**, như sau:

```
-->m(2,:)
```

```
ans =
```

```
4. 5. 23.
```

Và hàng ba:

```
-->m(:,3)
```

```
ans =
```

```
3.  
23.
```

chuyển vị ma trận hay một vectơ số để chỉ định dùng ngoặc đơn “'”:

```
-->m'
```

```
ans =
```

```
1. 4.  
2. 5.  
3. 23.
```

### Các thao tác

Các thao tác “\*” , “/” là các thao tác trong ma trận. Chúng thể hiện các thao tác theo từng phần tử, ta phải chú ý thứ tự của các thao tác: “.\*”, “./”.

---

```
-->A=[1,2,3;4,5,6]
```

```
A =
```

```
1. 2. 3.  
4. 5. 6.
```

---

```
-->B=[1;1;2]
```

---



B =	
1.	
1.	
2.	
-->A*B	
ans =	Phép nhân trong ma tr n
9.	
21.	
-->A*A	
--error 10	Chi u không phù h p
Phép nhân không phù h p.	
-->A.*A	
ans =	Phép nhân theo t ng ph n t
1. 4. 9.	
16. 25. 36.	
-->2*(A+2)	
ans =	Thao tác c th c hi n trên t ng ph n t vì 2 là m t ch s
6. 8. 10.	
12. 14. 16.	
-->A/A	Cho ma tr n X trong ó $X*A = A$ K t qu chính xác là:
ans =	1. 0 0 1.
1. 1.518D-16	tính toán c chính xác, k t qu
3.795D-15 1.	có th s h i khác bi t tu thu c vào phiên b n Scilab và h i u hành (xem chính xác c a phép tính, trang 29).
-->A./A	
ans =	Cho ma tr n chia thành t ng ph n t .
1. 1. 1.	
1. 1. 1.	
Trong tr ng h p vecto:	
-->C=1:4	
C =	
1. 2. 3. 4.	
-->C*C	
--error 10	Chi u không phù h p
Phép nhân không phù h p.	

<pre>--&gt;C.*C ans =     1.    4.    9.   16.</pre>	<p>Có thể nh p <math>C^2</math>, i v i vecto, các i s s c th hi n nh m t thao tác theo t ng ph n t . ây không phi i là tr ã ng h p gi ã ng v i ma tr n.</p>
<pre>--&gt;1/C ans =     0.0333333     0.0666667     0.1     0.1333333</pre>	<p>Trong tr ã ng h p c bi t i v i vecto này, ta tìm vecto X trong ó <math>C*X = 1</math></p>
<pre>--&gt;(1)./C ans =     1.    0.5   0.3333333     0.25</pre>	<p>Vi c ò ng c t ng ph n t Tr c ó, <math>C^{(-1)}</math> là kh d .Đ u ngo c quanh l là c n thi t i m bên c nh không c coi là m t đ u ph y thu c ch s 1. Có th vi t <math>1./C</math> v i m t ký t tr ã ng gi ã 1 và “.”</p>

#### Gi i các h tụy n tính

gi i h tụy n tính  $AX = Y$ , trong ó A là m t ma tr n vuông, s d ã ng đ u g ch chéo ng c “\”  
 $X = A \setminus Y$ .

L u ý, thao tác  $Y / A$  s cho (ch khi úng chỉ u) m t k t qu khác, ó là ma tr n Z trong ó  $Z * A = Y$

gi i h : 
$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \times X = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

-->A=[1 2 3;4 5 6];

-->Y=[1;1];

-->X=A\Y

X =

- 0.5

0.

0.5

-->A\*X

ans =

1.

1.

## Một số hàm hữu dụng

### Sort

Hàm **gsort** có dùng để phân loại các phần tử của một vectơ theo thứ tự tăng dần hoặc giảm dần.

```
-->v=[2,6,9,6,-4,0,2]
```

```
v =
```

```
2.    6.    9.    6. -4.    0.    2.
```

```
--> gsort(v, "g", "i")
```

```
ans =
```

```
-4.    0.    2.    2.    6.    6.    9.
```

```
--> gsort(v)
```

```
ans =
```

```
9.    6.    6.    2.    2.    0. -4.
```

### Length

Hàm **length** cho số các thành phần của một vectơ. Hàm **size** cho các chỉ số (hàng, cột) của một ma trận.

```
-->U=[1:10]
```

```
U =
```

```
1.    2.    3.    4.    5.    6.    7.    8.    9.    10.
```

```
-->length(U)
```

```
ans =
```

```
10.
```

```
-->m=[1 2 3;4 5 6];
```

```
-->size(U)
```

```
ans =
```

```
2.    3.
```

Sum and product

Hàm **sum** and **prod** tính tổng và tích của các phần tử.

```
-->U=[1:10];
```

```
-->sum(U)
```

```
ans =
```

```
55.
```

```
-->prod(U)
```

```
ans =
```

```
3628800.
```

Unique

Hàm **unique** liệt kê các phần tử của một vectơ một lần (ngay cả khi chúng có lặp lại) và phân loại chúng theo thứ tự tăng dần.

```
-->v=[2,6,9,6,-4,0,2]
```

```
v =
```

```
2. 6. 9. 6. -4. 0. 2.
```

```
-->unique(v)
```

```
ans =
```

```
-4. 0. 2. 6. 9.
```

Find

Hàm **find** tìm các phần tử trong một vectơ hoặc một ma trận và cho một vectơ chứa các chỉ số tương ứng:

tìm tất cả các phần tử của vectơ  $w$  nhỏ hơn 5:

```
-->w=[1,5,3,8,14,7,3,2,12,6]; find(w<5)
```

```
ans =
```

```
1. 3. 7. 8.
```

Vectơ  $k$  tương ứng (1,3,7,8) cho thấy các phần tử  $w_1, w_3, w_7$  and  $w_8$  nhỏ hơn 5.

tìm tất cả các phần tử của vectơ  $w$  bằng 3:

```
-->w=[1,5,3,8,14,7,3,2,12,6]; find(w==3)
```

```
ans =
```

```
3. 7.
```

Vectơ  $k$  thuộc  $(3,7)$  cho thấy các phần tử  $w_3$  và  $w_7$  bằng 3.

### chính xác c a phép tính

#### Phép tính

Các con số có giá trị tuy rất nhỏ khoảng  $2.2 \times 10^{-308}$  và  $1.8 \times 10^{+308}$ .

Chỉ số `%eps` trong `ng` và `ngv` là **2.220446049D-16** cho chính xác tính toán như một số có thể có trong các phép tính, và đó cho khoảng 16 chữ số thập phân.

#### Ví dụ 1: Phép tính $\sin(\pi)$

```
-->sin(%pi)
```

```
ans =
```

```
1.225D-16
```

Giá trị của  $\sin \pi$  trên không phải 0, nhưng có vẻ là không. Quyển vở, so sánh với giá trị của `c` của hàm `sin` (i.e. 1), giá trị này bằng 0 vì một giá trị thập phân `%eps`.

#### Ví dụ 2: Kiểm tra xem tam giác với các cạnh $\sqrt{3}, 1$ và $2$ có đúng là một tam giác với các góc phù hợp hay không:

---

```
-->a=sqrt(3)
```

```
a =
```

```
1.7320508
```

---

```
-->b=1
```

```
b =
```

```
1.
```

---

```
-->c=2
```

```
c =
```

```
2.
```

---

```
-->a^2+b^2==c^2
```

```
ans =
```

```
F
```

Chỉ số trình bày này không đúng do giá trị của  $a^2 + b^2$  là xấp xỉ

---

```
-->abs(a^2+b^2-c^2)<%eps
```

```
ans =
```

```
F
```

Ch ̣ng tṛnh ph n h i không ́ng  
do yêu c u giá tr ́nh xác tuy t ́i

---

```
-->abs(a^2+b^2-c^2)/c^2<%eps
```

```
ans =
```

```
T
```

Ch ̣ng tṛnh ph n h i ́ng do yêu  
c u giá tr ́nh xác t ́ng ́i

---

### Display

Các k t qu ́ c h i n th , theo m c ̣nh, v i 10 ký t , bao g m c ̣ d u th p phân. Hàm **format**  
c ̣ dùng ̣ h i n th nh i u s th p phân h n. ̣ h i n th 20 ch s th p phân, nh p  
**format(20)**.

Xét l i  $a=\sqrt{3}$ :

---

```
-->a^2
```

```
ans =
```

```
3.
```

̣ây có 7 ch s th p phân và ta không  
th y l i.

---

```
-->format(20)
```

```
-->a^2
```

```
ans =
```

```
2.99999999999999956
```

̣ây có 17 ch s th p phân và ta  
th y l i.

---

### Gi i các ph ̣ng tṛnh vi phân

t ́m các l i gi i cho m t h ̣ h i n c a các ph ̣ng tṛnh vi phân, ch c n h ph ̣ng tṛnh vi  
phân theo b c 1.

$$\begin{cases} y'(t) = f(y(t)) & \text{trong } \acute{o} t_0 \text{ và } t \text{ th h i n th i gian và } y_0 \text{ và } y(t) \text{ là các vecto,} \\ (t_0) = y_0 \end{cases}$$

Sau ó dùng hàm **ode**:  $y=\text{ode}(y_0,t_0,t,f)$ , v i:

- **y0**: i u ki n ban ́u, vecto có chi u n,
- **t0**: th i i m ban ́u,
- **t**: vecto có h ̣ng T t ̣ng ng v i các th i i m t ́nh áp án. Vecto này ph i b t ́u  
v i **t0**.
- **f**: hàm xác ̣nh h d i d ng:

```
function yprim=f(t,y)
```

```
    yprim(1)=...
```

```
    yprim(2)=...
```

```
    ....
```

```
    yprim(n)=...
```

```
endfunction
```

áp án là ma trận có hàng  $n \times T$ : 
$$\begin{pmatrix} y_1(1) & y_1(2) & \dots & y_1(T) \\ y_2(1) & y_2(2) & \dots & y_2(T) \\ \vdots & \vdots & \dots & \vdots \\ y_n(1) & y_n(2) & \dots & y_n(T) \end{pmatrix}$$

Ví dụ: giải hệ phương trình vi phân 
$$\begin{cases} \dot{y} = -2y \\ y(0) = 3, y(1) = 0 \end{cases}$$

Phương trình bậc 2 này có hai nghiệm riêng tìm được từ phương trình bậc 1 như sau:

$$Y = \begin{pmatrix} Y(1) \\ Y(2) \end{pmatrix} = \begin{pmatrix} y \\ y' \end{pmatrix}, Y_{prim} = \begin{pmatrix} Y_{prim}(1) \\ Y_{prim}(2) \end{pmatrix} = \begin{pmatrix} y' \\ y'' \end{pmatrix} \quad \text{và} \quad \begin{cases} Y_{prim}(1) = (2) \\ Y_{prim}(2) = -4 \times Y(1) \end{cases}$$

Chú thích	Trình số nguồn Scilab
Ta xác định hàm cho ra vectơ $y'$ từ các biến số $u$ vào $t$ và $y$ ( $y$ là một vectơ).	<code>function yprim=f(t,y)</code>
Ta xác định các giá trị các biến số.	<code>yprim(1)=y(2);</code> <code>yprim(2)=-4*y(1);</code>
(nguyên tử chỉ cần các giá trị phù hợp để cho phép tính suy ra bài giải).	<code>endfunction</code>
Ta xác định các điều kiện ban đầu.	<code>t0=0; tmax=5;</code>
Ta chọn hàm ode.	<code>t=t0:0.05:tmax;</code>
Ta viết công thức phân chia về vị trí.	<code>y0=3; yprim0=0;</code>
	<code>y=ode([y0;yprim0],t0,t,f);</code>
	<code>clf; plot(t,y(1,:))</code>

**Phân tích**

- **sqrt(x)** cho căn bậc hai của  $x$  với  $x$  là số thực không, và nghĩa phức của phần thực trong trường hợp phức.
- **log(x)** cho logarit tự nhiên của  $x$  với  $x$  là số thực dương.
- **exp(x)** cho cấp số nhân của  $x$  với  $x$  là số thực hoặc số phức.
- **abs(x)** cho giá trị tuyệt đối của  $x$  thực (hoặc mô-đun nếu  $x$  là số phức).
- **int(x)** biến các số hạng của  $x$  thành số dương trước số thập phân.
- **floor(x)** cho phần thực của  $x$  thực (số nguyên với  $n \leq x < n+1$ ).
- **ceil(x)** biến  $x$  thực thành số nguyên  $n$  với  $n-1 < x \leq n$ .

**Xác suất và thống kê**

- **factorial(n)** cho giai thừa của  $n$  với  $n$  là số nguyên dương không.
- **grand(1,p, "uin", m,n)** cho vectơ của các chuỗi số nguyên ngẫu nhiên  $p$  được lấy trong khoảng giá trị  $a$  và  $b$  với  $p$  là số nguyên dương,  $m$  và  $n$  là số nguyên và  $m \leq n$ .
- **grand(1,p, "unf", a,b)** cho vectơ của các chuỗi số ngẫu nhiên  $p$  được lấy trong khoảng  $a$  và  $b$  với  $p$  là số nguyên dương,  $a$  và  $b$  là số thực và  $a \leq b$ .
- **sum(n)** cho tổng các giá trị của vectơ  $n$  (dùng tính tổng).
- **cumsum(n)** cho vectơ bao gồm các giá trị tích lũy tổng của vectơ  $n$  (dùng tính các số tích lũy tổng).
- **length(v)** cho số các thành phần của vectơ  $v$ .
- **gsort(v)** cho vectơ bao gồm các số hạng của  $v$  được phân loại theo thứ tự giảm dần.
- **gsort(v, "g", "i")** cho vectơ bao gồm các số hạng của  $v$  được phân loại theo thứ tự tăng dần.
- **mean(v)** cho giá trị trung bình của vectơ bao gồm các số  $v$ .
- **stdev(v)** cho độ lệch chuẩn của vectơ các số  $v$ .
- **bar(v,n,couleur)** vẽ biểu đồ cột với  $v$  là ma trận  $X$ ,  $n$  là ma trận  $Y$ ,  $v$  và  $n$  là các vectơ cùng kích thước. Theo mặc định, **bar(n)** vẽ biểu đồ cột với các trục  $x$  và  $y$  bằng màu xanh dương với  $1, 2, 3, \dots$  là các trục  $x$ .
- **bar(v,[n1?, n2?])** vẽ biểu đồ cột kép với  $v$  là ma trận  $X$ ,  $n1$  là  $Y$ ; trục  $x$  màu xanh dương và  $n2$  là tọa độ trục  $Y$  màu xanh lá cây, với  $v$ ,  $n1$  và  $n2$  là các vectơ có cùng kích thước.
- **rand(n,p)** với  $n$  và  $p$  là số nguyên dương, cho ma trận  $n \times p$  bao gồm các số ngẫu nhiên lấy trong khoảng giá trị  $0$  và  $1$ .
- **rand()** cho một số thực ngẫu nhiên giữa  $0$  và  $1$ .
- **floor(x)** cho phần số nguyên của số thực  $x$ . Chức năng  $n$  với  $p$  là số giữa  $0$  và  $1$ , **floor(rand()+p)** sẽ là  $1$  với xác suất  $p$  và  $0$  với  $1 -$  xác suất  $p$ .



## Hiệu ứng và vektor

- **clf** nghĩa là “xóa biểu đồ” và xóa biểu đồ hiện tại trên cửa sổ hiện hành.
- **plot** cho phép vẽ các đường cong và biểu đồ phân tán theo hai chiều.
- **linspace(a,b,n)**, với  $a$  và  $b$  là số thực và  $n$  là số nguyên, xác định một vectơ gồm các giá trị đều đặn nằm giữa khoảng  $a$  và  $b$ .
- **scf** cho phép mở hoặc chiếm một cửa sổ đồ họa khác cửa sổ hiện tại.
- **surf** cho phép vẽ biểu đồ mặt phẳng 3D.
- **bar(X,Y)** trong đó  $X$  và  $Y$  là các vectơ, vẽ biểu đồ cột gồm các chuỗi giá trị  $y$  tại vị trí  $X$  trong đó có các số giá trị của  $Y$ .
- **plot(X,Y, “\*”)** vẽ biểu đồ phân tán các tọa độ  $(X(i),Y(i))$  là các ngôi sao. Có thể chọn màu.
- **plot(Y, “+”)** vẽ biểu đồ phân tán các tọa độ  $(i,Y(i))$  là dấu cộng.
- **disp(“Sentence”)** hiển thị nội dung ghi chú trong dòng lệnh lập trình.
- **disp(A)** trong đó  $A$  là một ma trận các chuỗi, hiển thị bảng các giá trị của  $A$ .
- **disp(“Sentence”+string(x))** hiển thị chuỗi và giá trị của  $x$ .
- **click** cho các tọa độ của điểm click trong Cửa sổ hiện hành.

## Tích

- **unique(v)** cho vectơ  $v$  với sự xuất hiện duy nhất của các thành phần của  $v$ .
- **sum(v)** cho tổng tất cả các phần tử của vectơ hay ma trận  $v$ .
- **prod(v)** cho tích tất cả các phần tử của vectơ hay ma trận  $v$ .
- **find(<test about v>)** cho các chuỗi của các phần tử thuộc vectơ  $v$  phù hợp với phép kiểm tra.
- **disp(x,y,...)** hiển thị các giá trị của các chuỗi trong giao diện người dùng.
- **string(x)** biến chuỗi  $x$  thành một chuỗi.
- **format(n)** trong đó  $n$  là một số nguyên lớn hơn hoặc bằng 2, hiển thị vị trí của các ký tự  $n$ , bao gồm dấu và dấu thập phân.
- **zeros(n,p)** xác định ma trận  $n \times p$  chỉ bao gồm số không.
- **feval(x,y,f)** trong đó  $x$  và  $y$  là các vectơ lần lượt có độ dài  $m$  và  $n$ , xác định ma trận  $m \times n$  trong đó phần tử  $(i,j)$  là  $f(x(i), y(j))$ .
- hàm **help** mô tả tình duy nhất trong các tài liệu của Scilab.
- **tic** khởi động đồng hồ.
- **toc** dừng đồng hồ.