



Khoa Công Nghệ Thông Tin Trường Đại Học Cần Thơ

Tính toán khoa học với GNU Octave và GNU Plot



Đỗ Thanh Nghị
dtnghi@cit.ctu.edu.vn

Cần Thơ
25-07-2012

Nội dung

- ★ Giới thiệu về Octave
- ★ Bắt đầu với Octave
- ★ Giải quyết bài toán đại số tuyến tính

Nội dung

- ★ Giới thiệu về Octave
- ★ Bắt đầu với Octave
- ★ Giải quyết bài toán đại số tuyến tính

Octave là gì?

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

- ★ OCTAVE được đề xuất bởi J.B. Rawlings (đại học Wisconsin-Madison) và J.G. Ekerdt (đại học Texas)
- ★ Version đầu tiên : 04-01-1993
- ★ OCTAVE dễ học, là phần mềm miễn phí mã nguồn mở, thích hợp cho đại số tuyến tính và xử lý ảnh

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

Đặc điểm

- ★ Dùng để giải quyết các bài toán số học, đặc biệt là đại số tuyến tính
- ★ Octave có ngôn ngữ lập trình riêng
- ★ Trong môi trường của Octave có thể trình bày dữ liệu bằng nhiều cách khác nhau
- ★ Có nhiều hàm để viết ứng dụng dễ dàng
- ★ Được sử dụng rộng rãi trong nhiều ngành khoa học

Nội dung

- ★ Giới thiệu về Octave
- ★ Bắt đầu với Octave
- ★ Giải quyết bài toán đại số tuyến tính



Giới thiệu về Octave



Bắt đầu với Octave



Giải quyết bài toán đại số tuyến tính

Bắt đầu với Octave

★ Vào môi trường Octave:

Start → Program → Octave (windows)

octave (linux)

★ Thoát khỏi môi trường Octave:

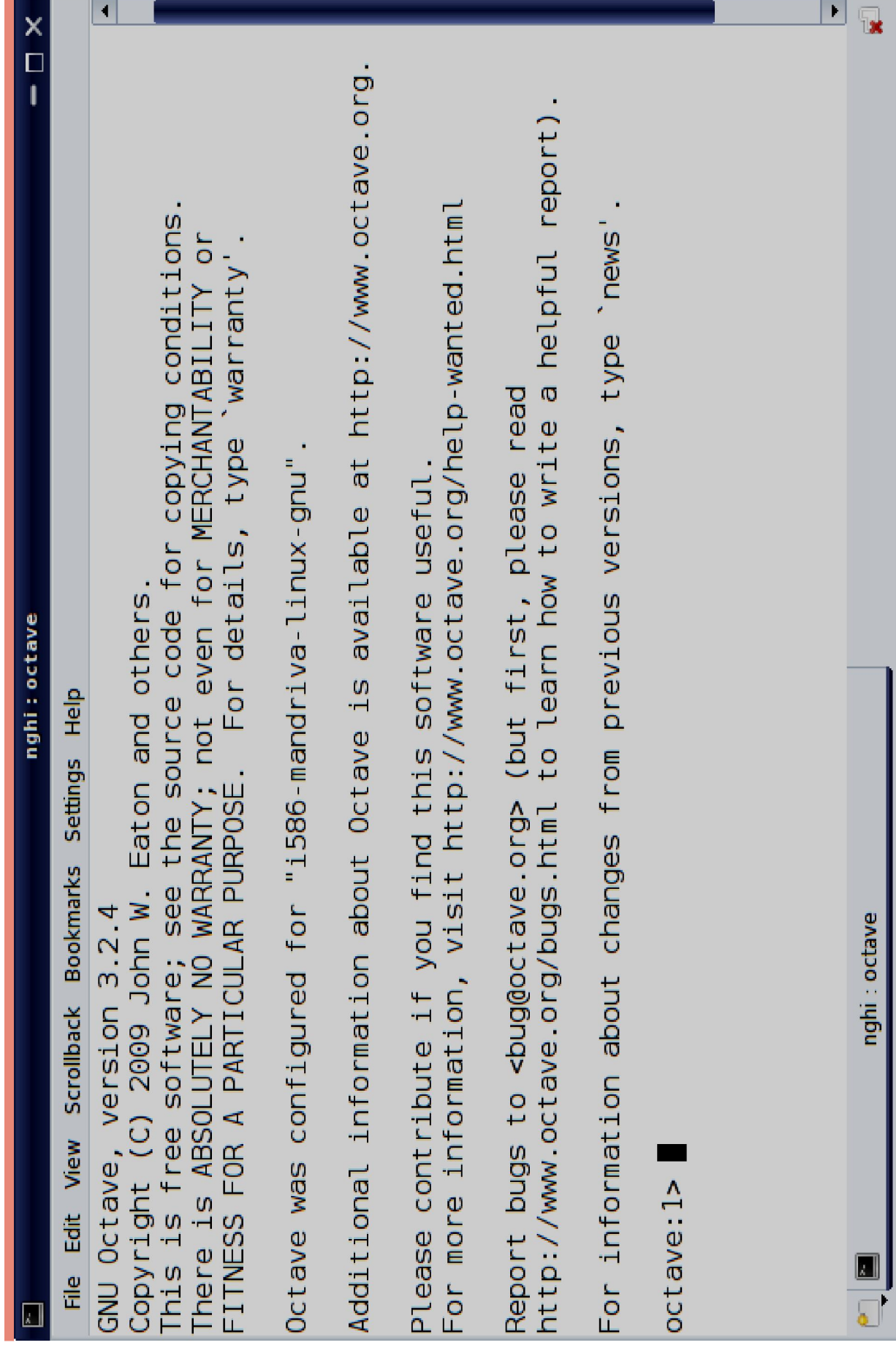
Nhập vào “quit” hoặc “exit”

Bắt đầu với Octave

★ Giới thiệu về Octave

★ **Bắt đầu với Octave**

★ Giải quyết bài toán đại số tuyến tính



```
nghi : octave
File Edit View Scrollback Bookmarks Settings Help

GNU Octave, version 3.2.4
Copyright (C) 2009 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type `warranty'.

Octave was configured for "i586-mandriva-linux-gnu".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html

Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).

For information about changes from previous versions, type `news'.

octave:1> █
```

★ Giới thiệu về Octave

★ **Bắt đầu với Octave**

★ Giải quyết bài toán đại số tuyến tính

Bắt đầu với Octave

★ Cách viết chương trình:

Viết lệnh trong Octave tương tự như việc viết một biểu thức trong toán học.

Ví dụ : `octave:1> a=2`

`octave:2> b=3`

`octave:3> c=a+b`

★ Cách xem kết quả:

Kết quả được Octave trả về ngay khi người sử dụng nhập xong yêu cầu lệnh.

★ Chú ý: **phân biệt ký tự thường hoa (như C)**

★ Giới thiệu về Octave

★ **Bắt đầu với Octave**

★ Giải quyết bài toán đại số tuyến tính

Bắt đầu với Octave

★ Các thao tác cơ bản trên tập tin:

★ Lưu dữ liệu : save anyname

★ Toàn bộ dữ liệu được lưu lại với tên người dùng đặt

★ Gọi lại dữ liệu : load anyname

★ Lưu lại biến : save tên_file [biến1,biến2,...]

★ Để lưu giữ những biến mà người dùng cần sử dụng lại nhiều lần.

★ Gọi lại biến : load tên_file

★ Gọi trợ giúp để biết cách sử dụng hàm: help tên_hàm

★ Giới thiệu về Octave

★ **Bắt đầu với Octave**

★ Giải quyết bài toán đại số tuyến tính

Bắt đầu với Octave

★ Tính toán trên Octave:

★ Sử dụng những phép tính đơn giản như :

$+$, $-$, $*$, $/$, $\dots$

★ Sử dụng những hàm có sẵn :

$\sin$, $\cos$, abs , $\dots$

★ Sử dụng những hàm do chính người làm tự tạo

★ Giới thiệu về Octave

★ **Bắt đầu với Octave**

★ Giải quyết bài toán đại số tuyến tính

Bắt đầu với Octave

★ Biến

- ★ Dùng để lưu trữ lại kết quả hoặc giá trị sử dụng lại
- ★ Người dùng không cần khai báo kiểu biến trước
- ★ Ví dụ:

```
deg = 3;  
new= 3*deg
```



Giới thiệu về Octave



Bắt đầu với Octave



Giải quyết bài toán đại số tuyến tính

Phép toán cơ bản

★ Các phép toán số học

octave:1> 2+2

ans = 4

★ Hàm toán học cơ bản

octave:2> deg = pi/180

deg = 0.017453

*octave:3> sin(90*deg)*

ans = 1

octave:4> pi

pi = 3.1416

*octave:5> deg = pi/180; sin(45*deg)*

ans = 0.70711



Giới thiệu về Octave



Bắt đầu với Octave



Giải quyết bài toán đại số tuyến tính

Phép toán cơ bản

```
octave:1> a=3 ;
```

```
octave:2> b = a + 5
```

```
b = 8
```

```
octave:3> 1/0
```

```
warning : division by zero
```

```
ans = Inf
```

```
octave:4> e^-ans
```

```
ans = 0
```

```
octave:5> a = 2+4i
```

```
a = 2+4i
```

```
octave:6> b = 3-6*j
```

```
b = 3-6i
```

```
octave:7> a+b
```

```
ans = 5-2i
```

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

Mảng và véc-tơ

★ Vectơ được xem là mảng trong ngôn ngữ Octave

★ Tạo mảng :

Cách dễ sử dụng nhất là các phần tử trong mảng được đặt trong dấu []

Ví dụ: $a = [1 \ 4 \ 5]$

$a = 1 \ 4 \ 5$

★ Dấu ':' trong Octave :

Ví dụ: $e = 2 : 6$

$e = 2 \ 3 \ 4 \ 5 \ 6$



Giới thiệu về Octave



Bắt đầu với Octave



Giải quyết bài toán đại số tuyến tính

Mảng và véc-tơ

octave:1> v = [1 2 3]

v = 1 2 3

octave:2> v = [1 ; 2 ; 3]

v = 1

2

3

octave:3> v = 1 : 5

v = 1 2 3

4

5

octave:4> v = 2 : 0.3 : 4

v = 2.000 2.3000 2.6000 2.9000 3.2000 3.5000 3.8000



Giới thiệu về Octave



Bắt đầu với Octave



Giải quyết bài toán đại số tuyến tính

Mảng và véc-tơ

Để truy xuất các phần tử của véc-tơ, ta sử dụng cặp dấu ngoặc ():

```
octave:1> v = [1 2 3]
```

```
V = 1 2 3
```

```
octave:2> v (2)
```

```
ans = 2
```

```
octave:3> v = [3 5 7];
```

```
octave:4> v (1:2)
```

```
ans = 3 5
```

```
octave:5> v ([1 3])
```

```
ans = 3 7
```

- ★ Giới thiệu về Octave
- ★ **Bắt đầu với Octave**
- ★ Giải quyết bài toán đại số tuyến tính

Ma trận

Khai báo ma trận cũng tương tự như véc-tơ, các hàng được phân cách với nhau bằng dấu chấm phẩy (;) hoặc xuống dòng:

octave:1> D = [1 2 3]

D = 1 2 3

octave:2> D = [D; 4 5 6]

D = 1 2 3
4 4 6

octave:3> D = [D; 7 8 9]

D = 1 2 3
4 5 6
7 8 9

Nội dung

- ★ Giới thiệu về Octave
- ★ Bắt đầu với Octave
- ★ Giải quyết bài toán đại số tuyến tính

GNU OCTAVE CHO ĐẠI SỐ TUYẾN TÍNH

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

★ Các phép toán trên ma trận và véc-tơ :

★ cộng (+),

★ trừ (-),

★ nhân (*),

★ chia (/),

★ lũy thừa (^).

★ các phép toán tác động lên từng phần tử tương ứng của 2 véc-tơ (ma trận) với nhau ví dụ như nhân từng phần tử tương ứng của 2 véc-tơ (ma trận) với nhau. Để thực hiện các phép toán này ta sử dụng dấu chấm (.) trước các phép toán.

GNU OCTAVE CHO ĐẠI SỐ TUYẾN TÍNH

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

```
octave:1> A = [1 2 3; 4 5 6]
```

```
A =
```

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

```
octave:2> A'
```

```
ans =
```

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

```
octave:3> A(1,2)
```

```
ans = 2
```

GNU OCTAVE CHO ĐẠI SỐ TUYẾN TÍNH

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

```
octave:4> A(1,:)
ans =
```

```
1 2 3
```

```
octave:5> AtA = A'*A
```

```
AtA =
```

```
17 22 27
```

```
22 29 36
```

```
27 36 45
```

```
octave:6> B = [-1 2 1; 0 3 1]
```

```
B =
```

```
-1 2 1
```

```
0 3 1
```

GNU OCTAVE CHO ĐẠI SỐ TUYẾN TÍNH

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

```
octave:7> C = A + B
```

```
C =
```

```
0 4 4
```

```
4 8 7
```

```
octave:8> D = A*B'
```

```
D =
```

```
6 9
```

```
12 21
```

```
octave:9> det(D)
```

```
ans = 18
```

GNU OCTAVE CHO ĐẠI SỐ TUYẾN TÍNH

- ★ Giới thiệu về Octave
- ★ Bắt đầu với Octave
- ★ Giải quyết bài toán đại số tuyến tính

octave:8> I = eye(3)

I =

Diagonal Matrix

1 0 0
0 1 0
0 0 1

octave:9> e = ones(1,3)

e =

1 1 1

GNU OCTAVE CHO ĐẠI SỐ TUYẾN TÍNH

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

```
octave:10> m = diag([1 2 3])
```

m =

Diagonal Matrix

```
1 0 0
0 2 0
0 0 3
```

```
octave:11> r = rand(3,4)
```

r =

```
0.538382 0.220173 0.435500 0.100588
0.302560 0.431150 0.798796 0.470263
0.053377 0.128410 0.565626 0.452748
```

GNU OCTAVE CHO ĐẠI SỐ TUYẾN TÍNH

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

```
octave:1> a = [1 3 5 -1 0];  
octave:2> b = [1 2 3 4 5];
```

```
octave:3> a.*b  
ans = 1    6    15   -4    0
```

```
octave:4> b.^2  
ans = 1    4    9    16   25
```

```
octave:5> 2.^b  
ans = 2    4    8    16   32
```

GNU OCTAVE CHO ĐẠI SỐ TUYẾN TÍNH

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

```
octave:1> a = [1 3 5 -1 0];  
octave:2> b = [1 2 3 4 5];
```

```
octave:3> dot(a,b)  
ans = 18
```

```
octave:4> sum(a.*b)  
ans = 18
```

```
octave:5> a*b'  
ans = 18
```

GNU OCTAVE CHO ĐẠI SỐ TUYẾN TÍNH

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

★ Một hệ phương trình tuyến có dạng:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

...

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m$$

★ Hay có thể được viết lại dưới dạng ma trận:

$$Ax = b$$

★ Để tìm x ta biến đổi:

$$Ax = b \Leftrightarrow A^{-1}Ax = A^{-1}b \Leftrightarrow x = A^{-1}b$$

Nếu A khả nghịch, ta tìm được x .



Giới thiệu về Octave



Bắt đầu với Octave



Giải quyết bài toán đại số tuyến tính

Ví dụ

★ Giải hệ phương trình :

$$x + y = 3$$

$$2x - 3y = 5$$

★ Trong Octave, ta tìm x và y như sau:

`octave:1> A = [1 1; 2 -3];`

`octave:2> b = [3; 5];`

`octave:3> inv(A)*b`

`ans =`

`2.80000`

`0.20000`

Vậy $x = 2.8$ và $y = 0.2$

Nhận xét

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

★ Sử dụng ma trận nghịch đảo để giải hệ phương trình là một phương pháp không hiệu quả về mặt thời gian

★ Có một cách khác giải nhanh hơn nhiều và phương pháp này tương đương với phương pháp khử của Gauss.

★ Octave định nghĩa 2 “phép chia” giữa 2 ma trận như sau:

$A \setminus B$ tương đương với $A^{-1}B$, dùng để giải hệ phương trình $Ax = B$

A/B tương đương với AB^{-1} , dùng để giải hệ phương trình $xA = B$

★ Quay lại ví dụ trên:

octave:3> A\b

ans =

2.80000

0.20000

Nhận xét

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

★ Cả 2 phương pháp cùng cho một kết quả như nhau

★ Phương pháp chia ($A \setminus b$) có một ưu điểm là nếu ma trận A không có ma trận nghịch đảo, Octave sẽ tìm một nghiệm gần đúng thỏa mãn hệ phương trình với sai số nhỏ nhất

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

Giá trị riêng – Véc-tơ riêng

★ Bài toán $Ax = \lambda x$, nghiệm của bài toán này là véc-tơ riêng (eigenvector) và giá trị riêng (eigenvalue) của ma trận A. Octave hỗ trợ hàm **eig** cho phép tìm giá trị riêng và véc-tơ riêng của ma trận

Ví dụ:

```
octave:1> A = [1 3 -2; 3 5 1; -2 1 4];
```

```
octave:2> eig(A)
```

```
ans =
```

```
-1.51204
```

```
4.90448
```

```
6.60756
```

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

Ví dụ

★ Để tìm véc-tơ riêng, ta phải sử dụng 2 biến để lưu kết quả trả về

octave:1> [V,D] = eig(A)

V = -1.818394 -0.315302 -0.480434
0.434733 0.207055 -0.876433
-0.375818 0.926128 0.032380

D = -1.51204 0.00000 0.00000
0.00000 4.90448 0.00000
0.00000 0.00000 6.60756

★ Mỗi cột của V là một véc-tơ riêng

★ Các giá trị riêng nằm trên đường chéo chính của D

★ Giới thiệu về Octave

★ Bắt đầu với Octave

★ Giải quyết bài toán đại số tuyến tính

Ví dụ

★ Để kiểm tra kết quả của hàm eig, ta có thể tính $V \cdot D \cdot \text{inv}(V)$, kết quả phải là A.

octave:2> $V \cdot D \cdot \text{inv}(V)$

ans =

1.00000	3.00000	-2.00000
3.00000	5.00000	1.00000
-2.00000	1.00000	4.00000

Nội dung

- ★ Giới thiệu về Octave
- ★ Bắt đầu với Octave
- ★ Giải quyết bài toán đại số tuyến tính
- ★ Lập trình

Hàm

★ Hàm

★ Hàm có sẵn :

★ Octave cung cấp khá nhiều hàm cho người dùng để dàng lựa chọn

★ Cách sử dụng: nhập **tên hàm(đối số)**

Ví dụ : $\sin(\pi)$, $\exp(1)$,...

★ Hàm tự tạo:

Người sử dụng có thể tự tạo những thư viện hàm cho Octave trong môi trường soạn thảo text như emacs, text editor.

Lập trình script

- ★ Scripts: là một file văn bản chứa các câu lệnh
- ★ Tạo và chỉnh sửa các scripts:
 - ★ Sử dụng các trình soạn thảo như text editor, emacs... để soạn thảo các scripts files.
 - ★ Cú pháp: chỉ đơn thuần bao gồm các câu lệnh và không được bắt đầu bằng từ khóa *Function*.

★ Ví dụ:

```
clear;  
a=1;  
b=2;  
c=a+b
```

Lập trình script

- ★ Scripts: là một file văn bản chứa các câu lệnh
- ★ Tạo và chỉnh sửa các scripts:

★ Lưu file đã soạn thảo: vào file → save buffer as...
với phần mở rộng là đuôi .m. Ví dụ lưu file vừa tạo
với tên tong.m

★ Khi cần thiết phải chỉnh sửa, gõ lệnh edit từ cửa
sổ lệnh của Octave
Octave##> edit [<script's name>]

★ Ví dụ: Octave##> edit tong.m

Lập trình script

Chạy và sửa các lỗi của scripts:

★ Khi muốn sử dụng các scripts đã tạo, gõ vào tên scripts từ dấu nhắc của Octave ở cửa sổ lệnh:

Octave##> <script's name>

★ Ví dụ: Octave##> tong.m

khi đó Octave sẽ cho ra kết quả bằng 3.

★ Nếu phát hiện scripts vừa tạo có lỗi, Octave sẽ gửi thông điệp thông báo

★ Trở lại trình soạn thảo, sửa lỗi và save file lại lần nữa, sau đó gõ lại tên scripts files và thực thi chương trình

Lập trình function

Hàm cho phép bạn tự định nghĩa các câu lệnh và sử dụng cho các mục đích về sau.

Cú pháp tổng quát:

```
function [out1, out2,...]=fname[inp1, inp2,...]  
    body  
endfunction
```

Lập trình function

Ví dụ: viết hàm tính giai thừa

```
function gthua=giaithua(n)
    gthua=1;
    for i=2:n
        gthua=gthua*i;
    endfor
endfunction
```

Cách gọi hàm:

Từ dấu nhắc của Octave gõ vào tên hàm và đối số,
ví dụ để gọi hàm tính giai thừa vừa định nghĩa
Octave# #> giaithua(5)

Lệnh điều khiển

Câu lệnh if...else

- Cú pháp:
if điều kiện1
 công việc 1
elseif điều kiện 2
 công việc 2
else điều kiện n
 công việc n
endif
- Ví dụ:
if (rem (x, 2) == 0)
 printf ("x la so chan\n");
else
 printf ("x la so le\n");
endif

Lệnh điều khiển

Câu lệnh switch...case :

– Cú pháp :

```
switch x
case x1      công việc 1
case x2      công việc 2
.....
otherwise   công việc n
end
```

– Ví dụ:

```
switch diem
case 5      printf (" ket qua tb ");
case 10     printf (" ket qua gioi ");
end
```

Lệnh vòng lặp

- Câu lệnh for:

- Cú pháp :

```
for biến = vector  
    công việc
```

```
endfor
```

với vector có dạng 'i : n'

- Ví dụ: viết đoạn chương trình tạo ra một vector chứa 10 giá trị đầu trong dãy Fibonacci.

```
fibonacci = ones(1, 10);  
for i = 3:10  
    fibonacci(i) = fibonacci(i-1) + fibonacci(i-2);  
endfor
```

Lệnh vòng lặp

Câu lệnh while:

– Cú pháp :
 while điều kiện
 công việc
 endwhile

– Ví dụ: viết lại đoạn chương trình tạo ra vector chứa 10 giá trị đầu trong dãy số Fibonacci.

```
fibonacci = ones(1, 10);  
i = 3;  
while (i <= 10)  
    fibonacci(i) = fibonacci(i-1) + fibonacci(i-2);  
    i++;  
endwhile
```

Lệnh vòng lặp

Câu lệnh do...until

– Cú pháp:

do

body

until điều kiện đúng

– Ví dụ: viết lại đoạn chương trình tạo vector chứa 10 giá trị đầu của dãy Fibonacci.

```
fibonacci = ones(1, 10);
```

```
i = 2;
```

```
do
```

```
    i++;
```

```
    fibonacci(i) = fibonacci(i-1) + fibonacci(i-2);
```

```
until (i == 10)
```

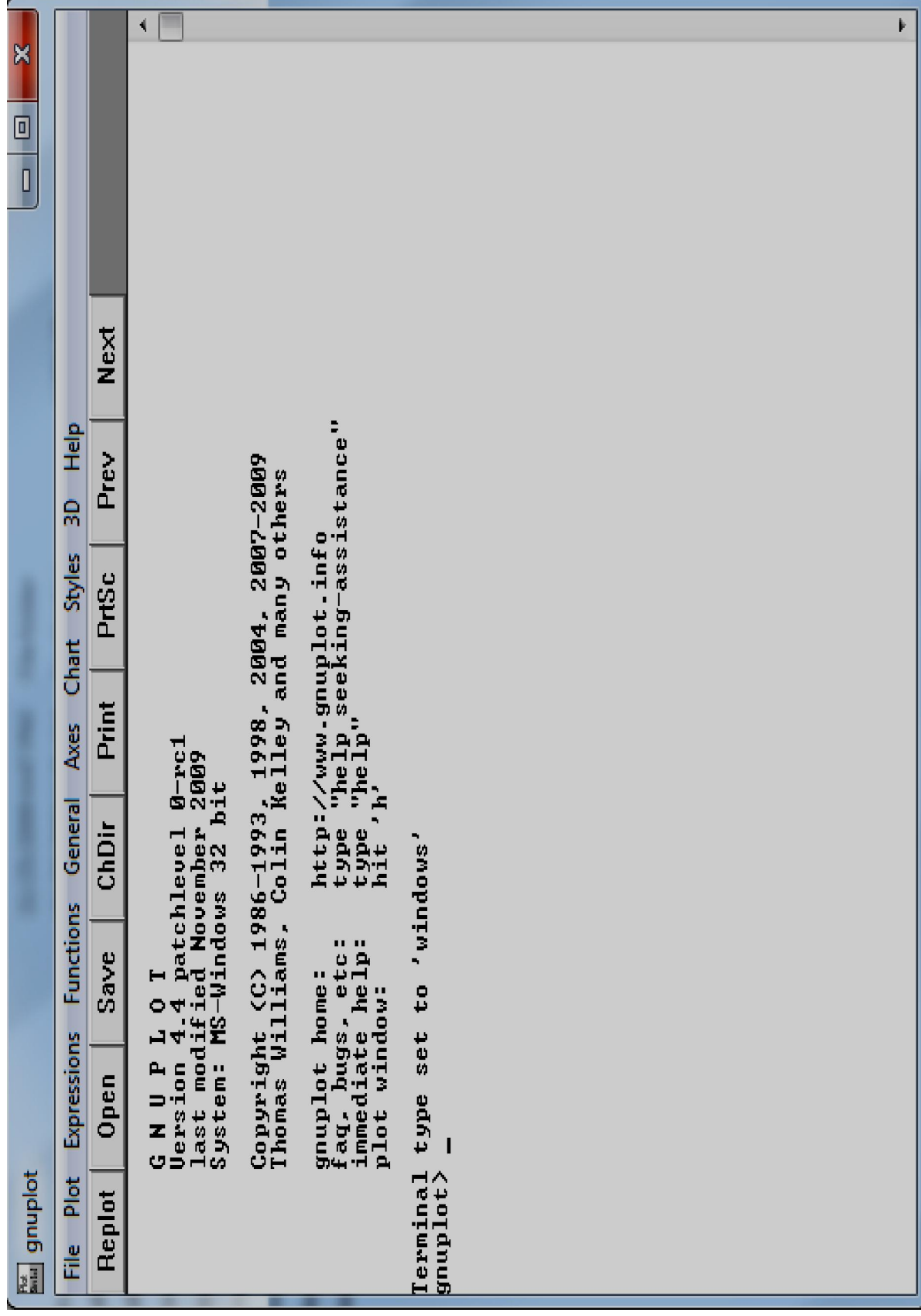
Nội dung

- ★ Giới thiệu về Octave
- ★ Bắt đầu với Octave
- ★ Giải quyết bài toán đại số tuyến tính
- ★ Lập trình
- ★ GNU Plot

GNUPlot

- ★ GNUPlot được đề xuất bởi C. Kelley và T. Williams năm 1986 để vẽ đồ thị của hàm số và dữ liệu
- ★ GNU Plot cung cấp môi trường tương tác lệnh để vẽ đồ thị rất phong phú
- ★ GNU Plot là phần mềm tự do, có thể chạy được trên nhiều hệ điều hành như Wins, Linux/Unix, Mac

GNUPlot



GNUPlot

★ Cú pháp:

```
plot {<ranges> }  
      {<function> | {"<datfile>" {datfile-modif}}} }  
      {axes <axes> } {<title-spec> } {with <style> }  
      {, {definitions,} <function> ... }
```

★ Trợ giúp
help plot

GNUPlot

★ Các hàm:

`abs(x)`

Trị tuyệt đối của x

`acos(x)`

Arc-cosine của x

`asin(x)`

Arc-sine của x

`atan(x)`

Arc-tangent của x

`cos(x)`

Cosine của x

`cosh(x)`

Hyperbolic cosine của x

`erf(x)`

Hàm lỗi của x

`exp(x)`

Hàm e^x

`inverf(x)`

Hàm lỗi nghịch đảo của x

`invnorm(x)`

Phân phối chuẩn nghịch đảo của x

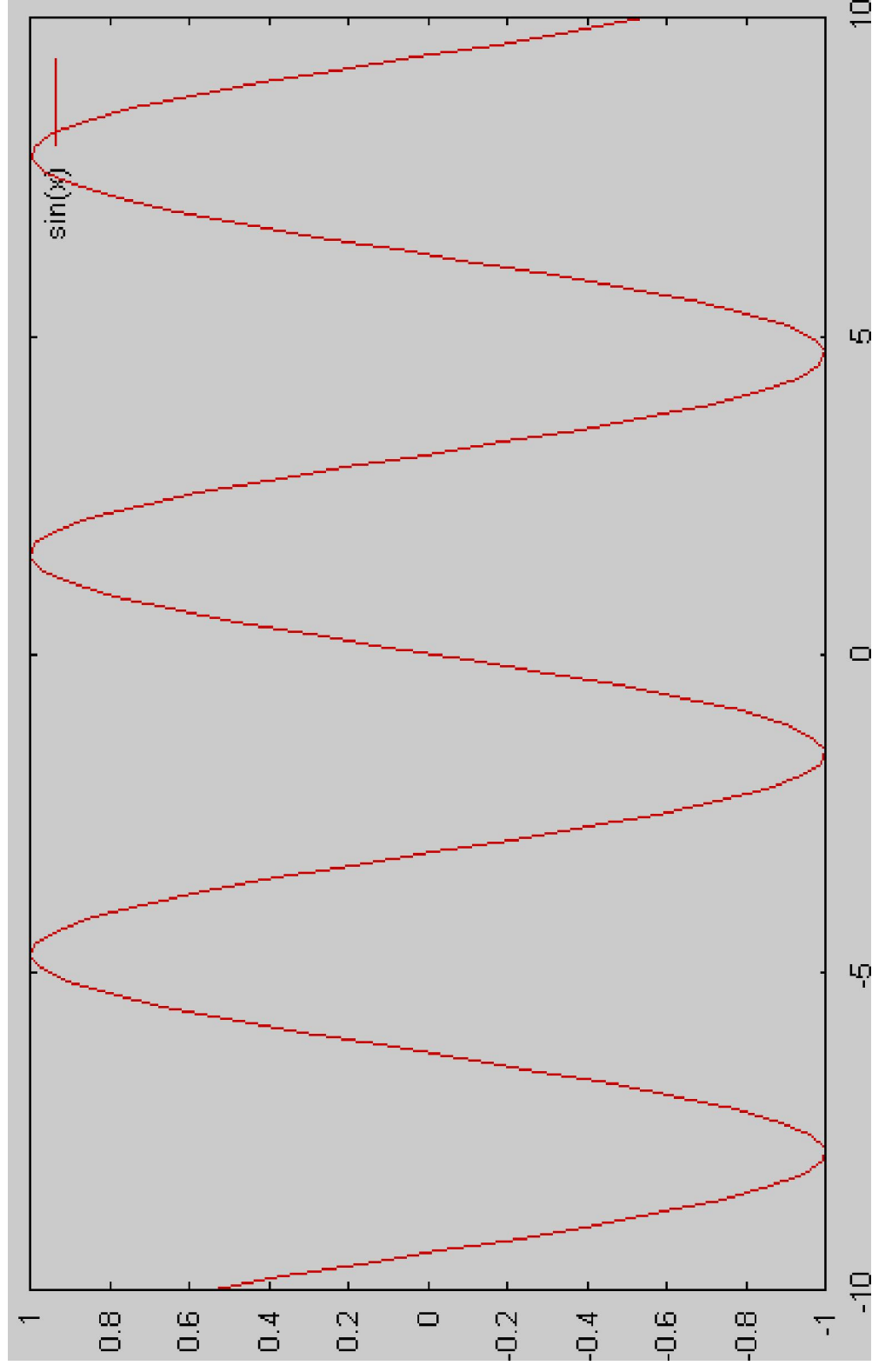
GNUPlot

★ Các hàm:

$\log(x)$	Hàm ln của x
$\log10(x)$	Hàm log cơ số 10 của x
$\text{norm}(x)$	Hàm phân phối chuẩn Gauss
$\text{rand}(x)$	Hàm ngẫu nhiên
$\text{sgn}(x)$	Trả về 1 nếu $x > 0$, trả về -1 nếu $x < 0$, trả về 0 nếu $x = 0$
$\sin(x)$	Hàm sine của x
$\sinh(x)$	Hyperbolic sine của x
$\text{sqrt}(x)$	Căn bậc 2 của x
$\tan(x)$	Hàm tangent của x
$\tanh(x)$	Hyperbolic tangent của x

GNUPlot

`plot sin(x)`



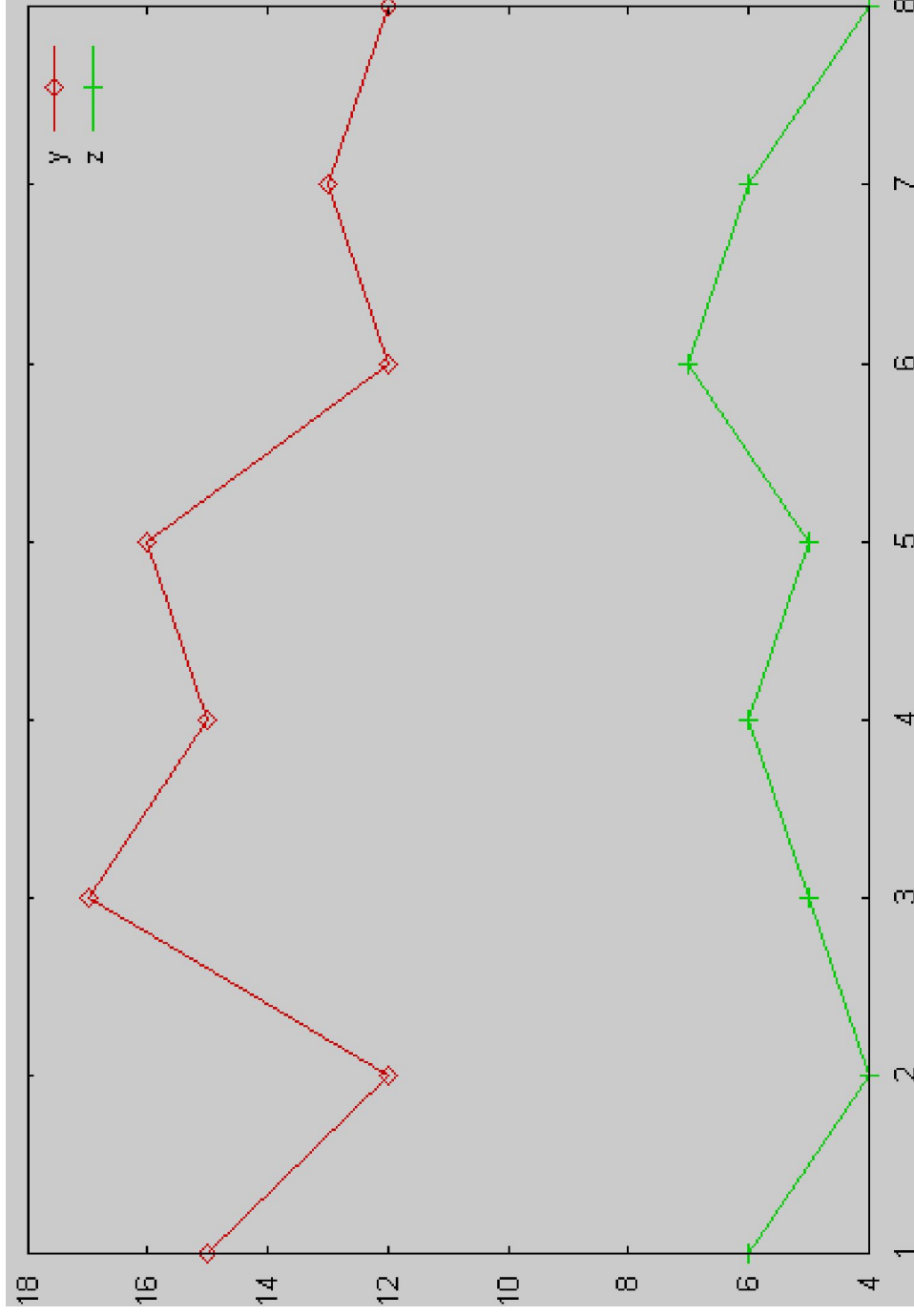
GNUPlot

★ Tập tin **mydata.dat**:

```
# My data
# x y z
1 15 6
2 12 4
3 17 5
4 15 6
5 16 5
6 12 7
7 13 6
8 12 4
```

GNUPlot

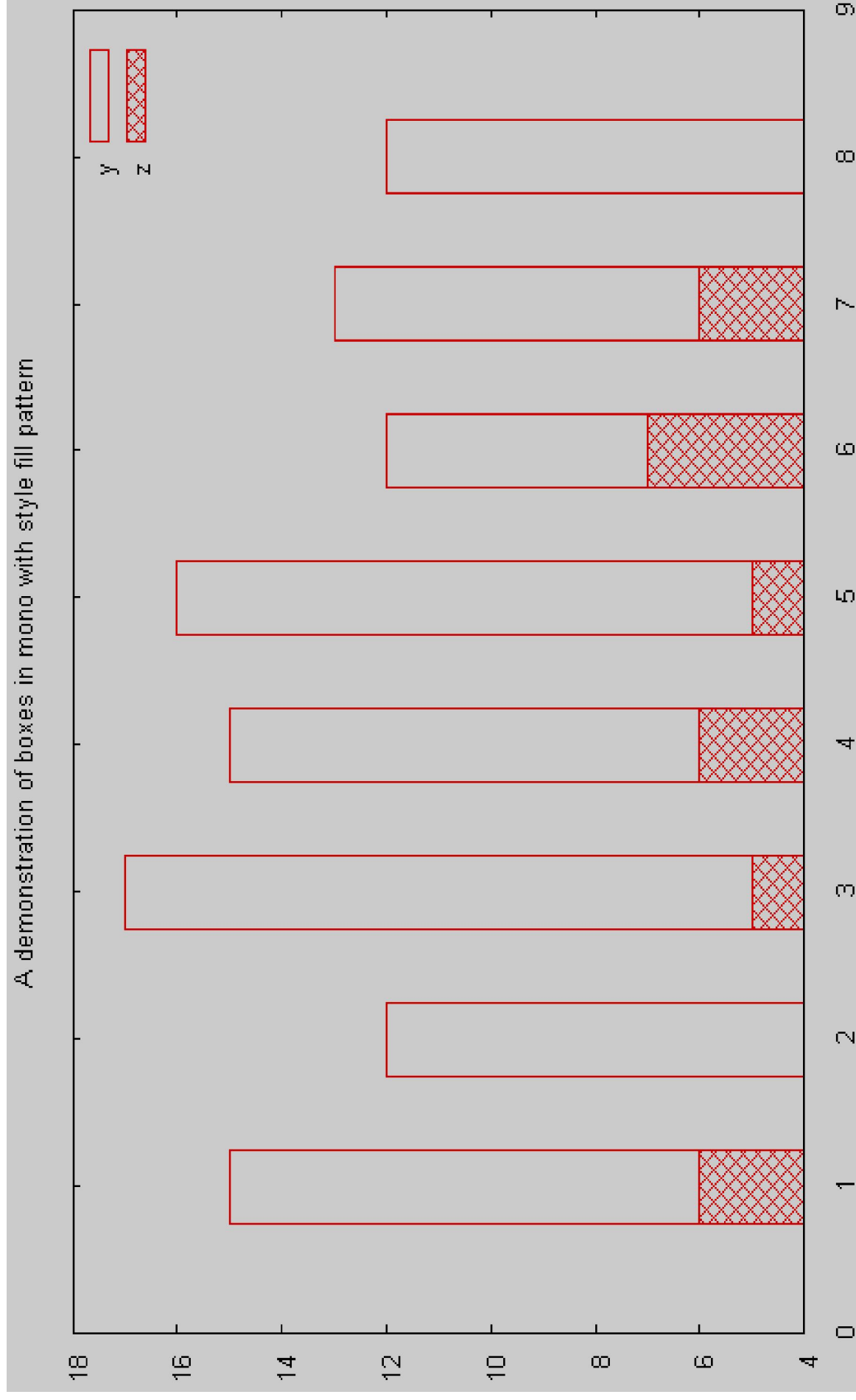
plot "mydata.dat" using 1:2 title 'y' with linespoints, \
"mydata.dat" using 1:3 title 'z' with linespoints



GNUPlot

```
set title "A demonstration of boxes in mono with style  
fill pattern"  
set xr [0:9]  
set boxwidth 0.5  
set style fill pattern border  
plot "mydata.dat" using 1:2 title 'y' with boxes lt 1, \  
      "mydata.dat" using 1:3 title 'z' with boxes lt 1
```

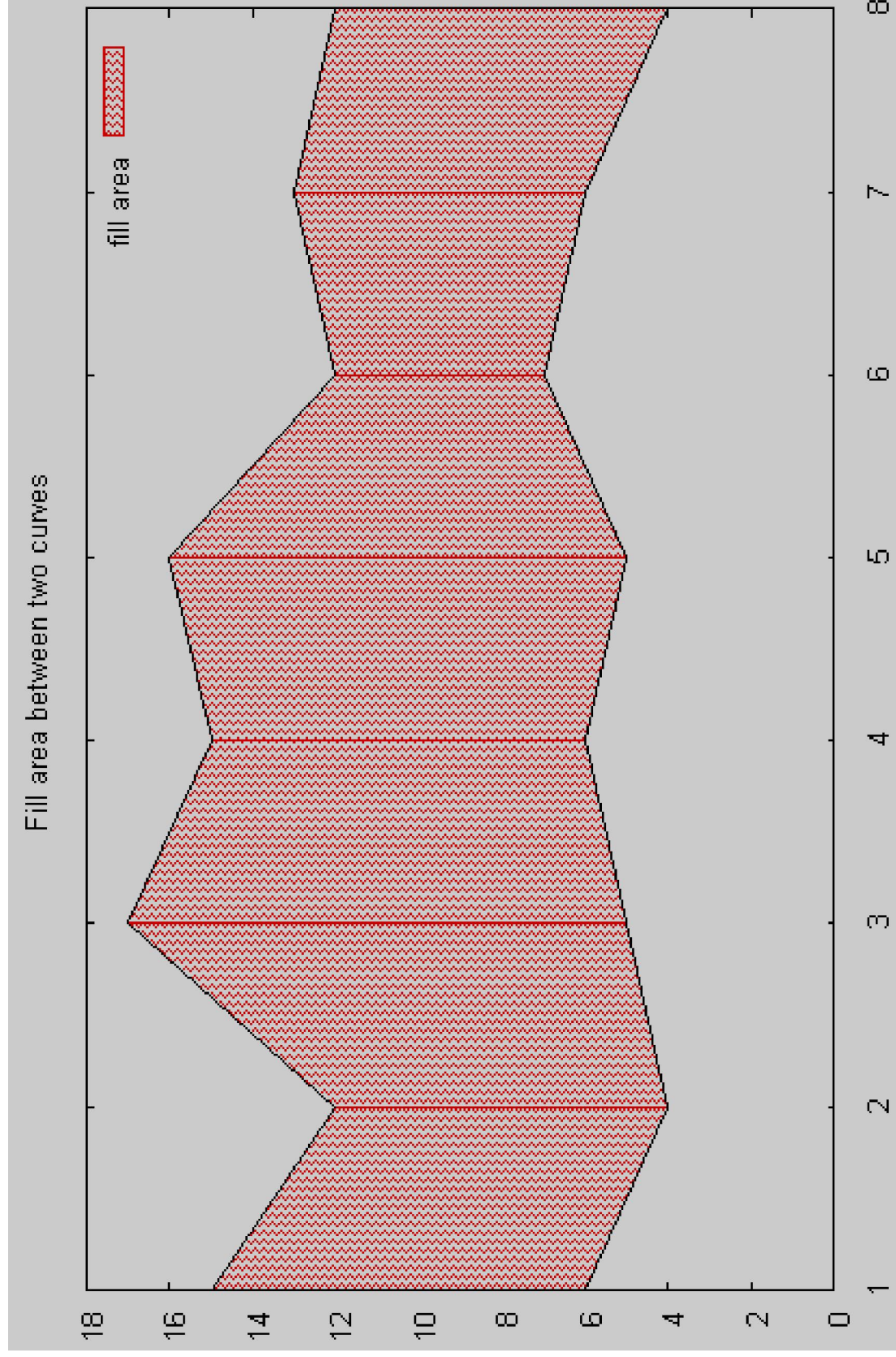
GNUPlot



GNUPlot

```
set title "Fill area between two curves"
set style fill pattern 2
set xr [1:*]
set yr [0:*]
plot "mydata.dat" using 1:2:3 title "fill area" with
filledcurves lt 1, "mydata.dat" using 1:2 lt -1 notitle,
"mydata.dat" using 1:3 lt -1 notitle
```

GNUPlot



GNUPlot

★ Các tùy chọn:

`set title "title name"`

Đặt tiêu đề cho đồ thị

`set xlabel "x label"`

Đặt tiêu đề cho trục x

`set ylabel "y label"`

Đặt tiêu đề cho trục y

`set xrange [v1:v2]`

Đặt range giá trị của x từ v1-v2

`set yrange [v1:v2]`

Đặt range giá trị của y từ v1-v2

`set autoscale`

Tự động scaling ranges

`unset logscale;`

Sử dụng logscale của trục y

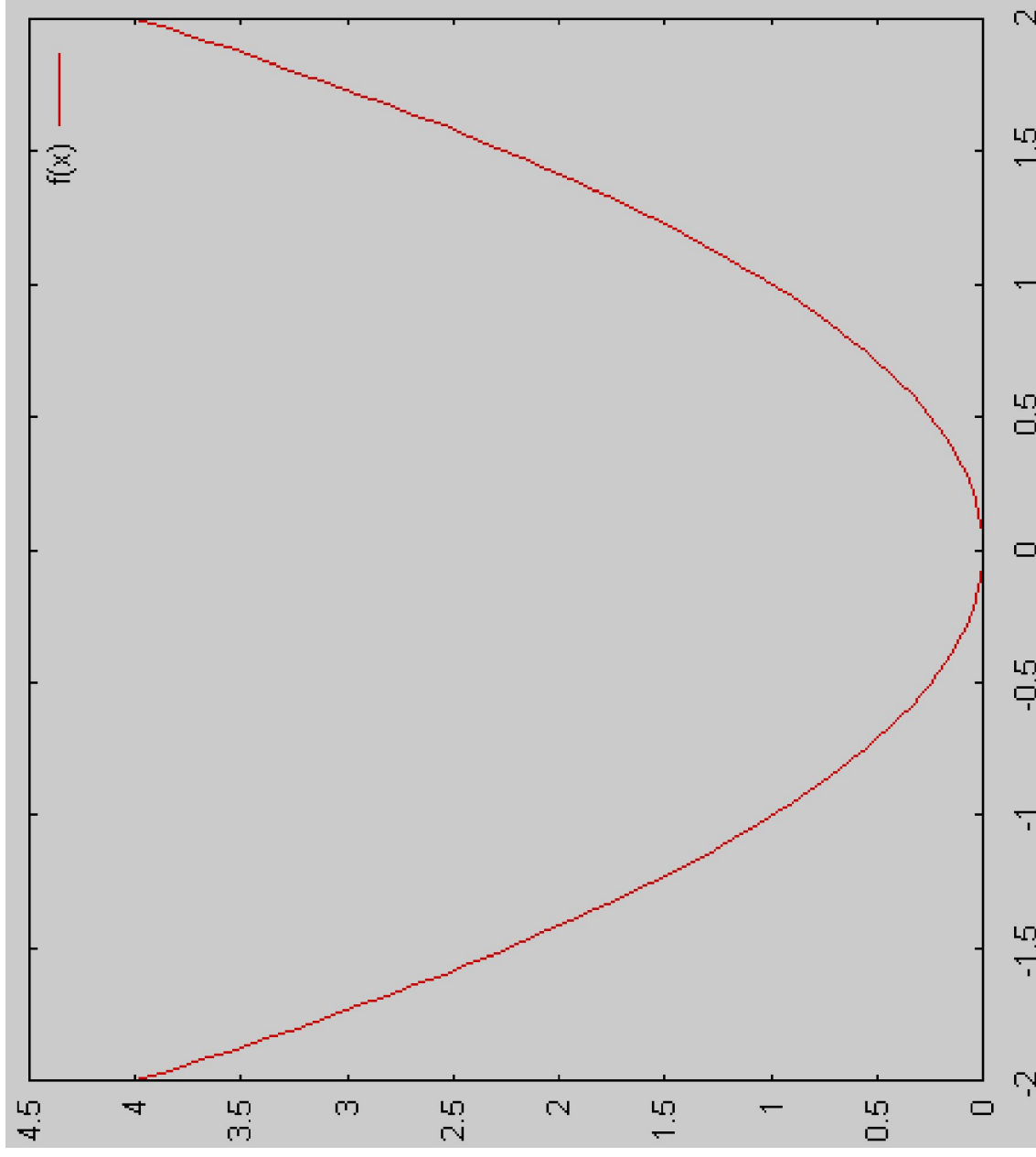
`set logscale y`

`unset; set multiplot`

Đặt mode vẽ nhiều đồ thị

GNUPlot

```
f(x) = x*x  
set xrange [-2:2]  
plot f(x) title 'f(x)'
```

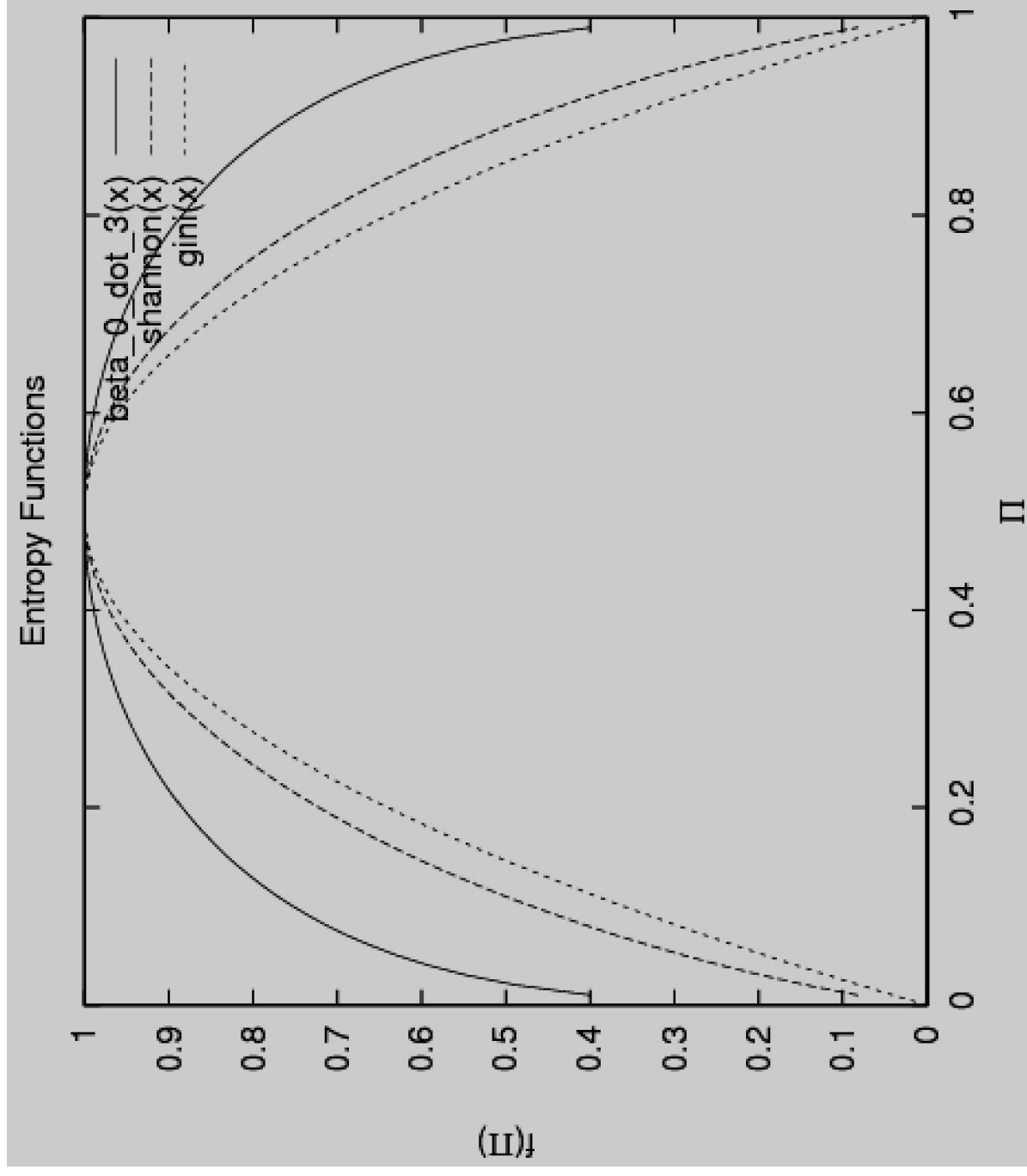


GNUPlot

```
set size 1.0, 0.6
set terminal postscript portrait enhanced mono dashed lw 1
"Helvetica" 14
set output "my_plot.ps"
set title "Entropy Functions"
set xlabel "{/Symbol P}"
set ylabel "f({/Symbol P})"
set xr [0:1]

pow(a,k) = exp(k*log(a))
shannon(x) = (-x*log(x)/log(2) - (1-x)*log(1-x)/log(2))
beta_0_dot_3(x) = (pow(2,-0.7)/(pow(2,-0.7)-1))*(1-
pow(x,0.3) - pow(1-x,0.3))
gini(x) = 2*(1- x*x - (1-x)*(1-x))
plot beta_0_dot_3(x), shannon(x), gini(x)
```

GNUPlot



GNUPlot

★ Tập tin **candlesticks.dat**:

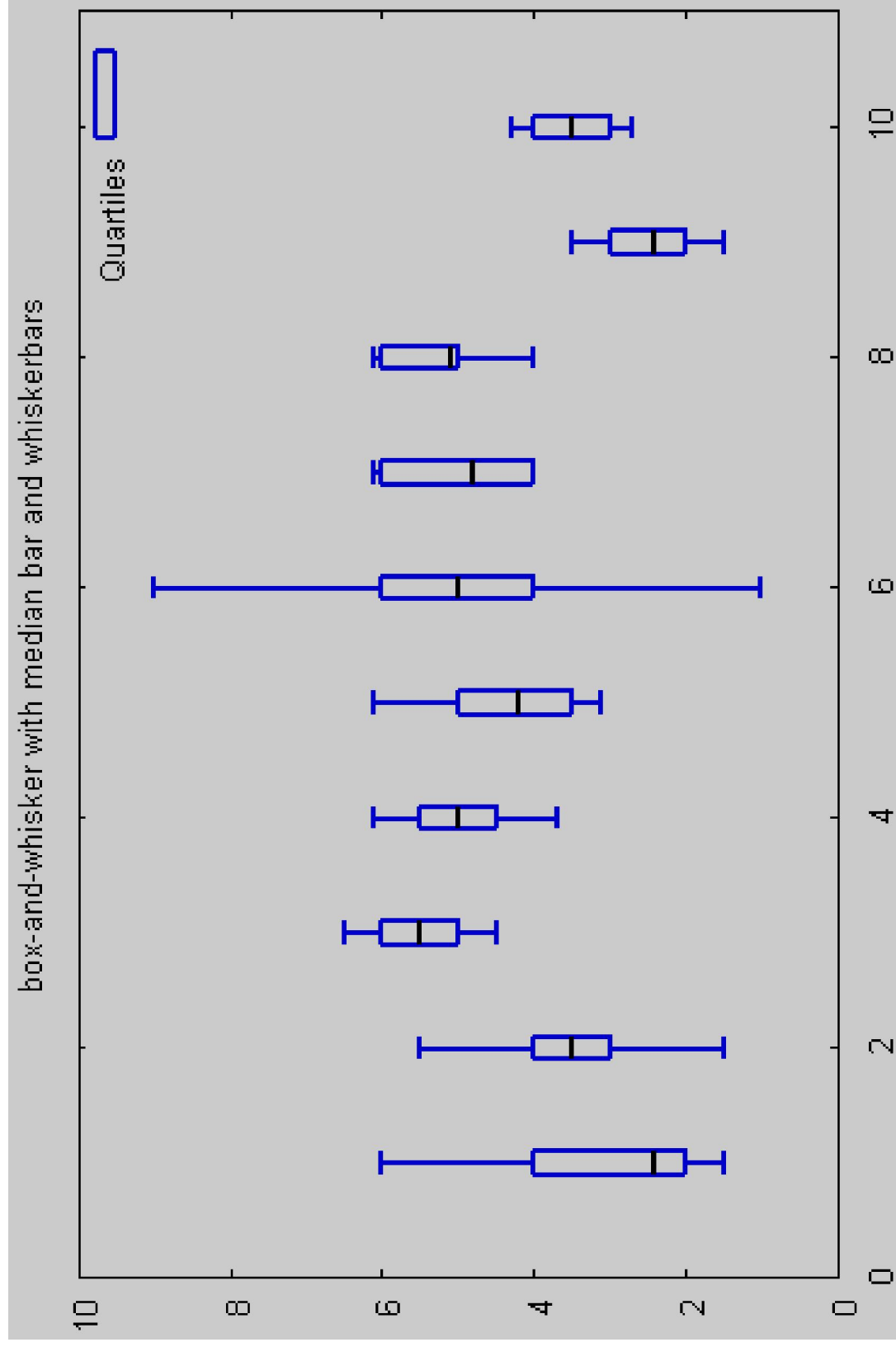
Candlesticks.dat

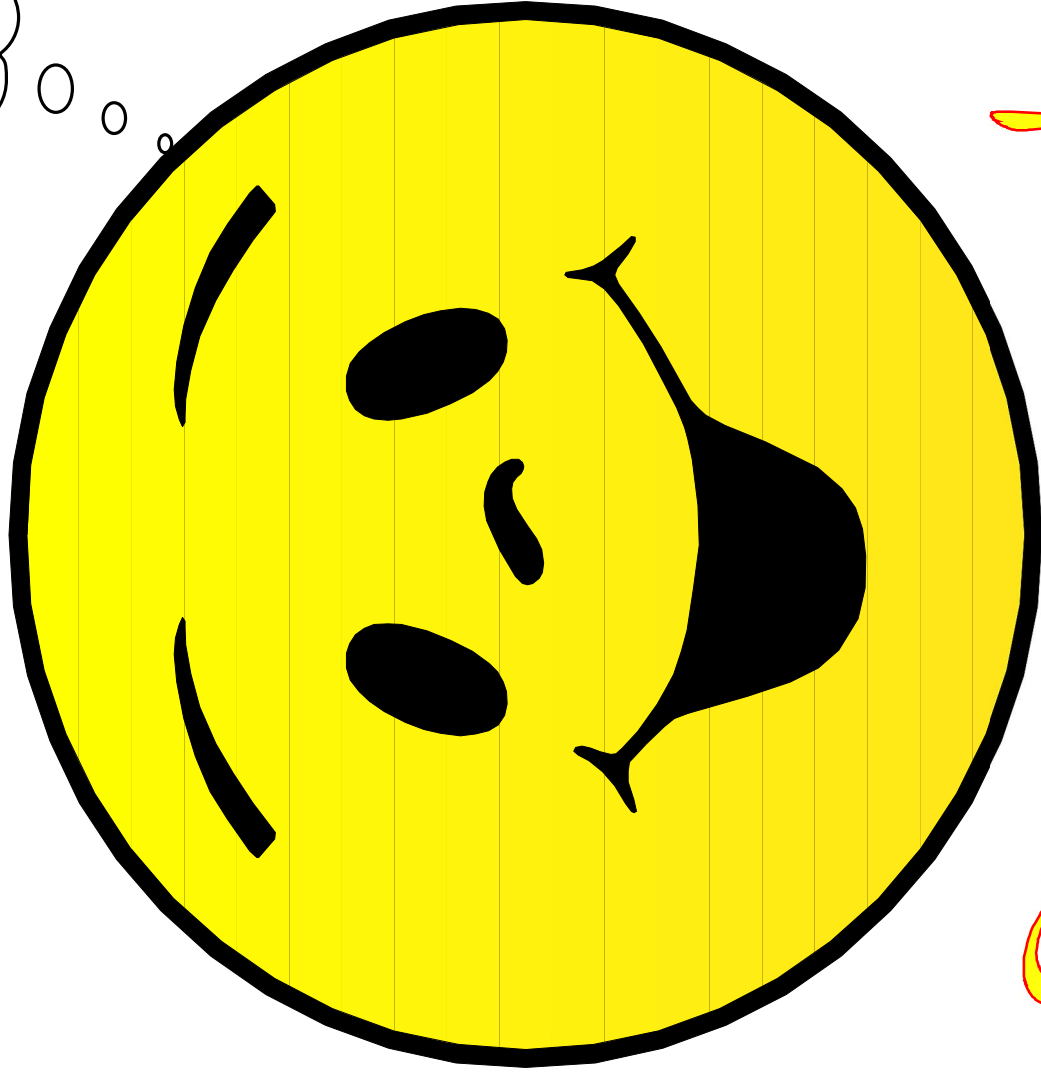
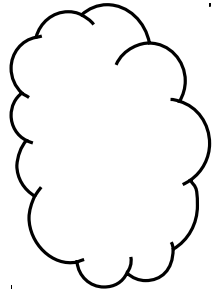
#	min	low(Q1)	med(Q2)	up(Q3)	max
1	1.5	2	2.4	4	6.
2	1.5	3	3.5	4	5.5
3	4.5	5	5.5	6	6.5
4	3.7	4.5	5.0	5.5	6.1
5	3.1	3.5	4.2	5	6.1
6	1	4	5.0	6	9
7	4	4	4.8	6	6.1
8	4	5	5.1	6	6.1
9	1.5	2	2.4	3	3.5
10	2.7	3	3.5	4	4.3 ⁶⁴

GNUPlot

```
set boxwidth 0.2 absolute
set title "box-and-whisker with median bar and
whiskerbars"
set xr [0.00000 : 11.0000 ] noreverse nowriteback
set yr [ 0.00000 : 10.0000 ] noreverse nowriteback
plot "candlesticks.dat" using 1:3:2:6:5 with
candlesticks lt 3 lw 2 title "Quartiles" whiskerbars,
"candlesticks.dat" using 1:4:4:4:4 with candlesticks
lt -1 lw 2 notitle
```

GNUPlot





Cám on!