

Notas de Aula

MatLab - 2

Routo Terada

www.ime.usp.br/~rt

Depto. C. da Computação - USP

Bibliografia:

**E. Y. Matsumoto, MatLab6 Fundamentos de Programação,
Edit. Érica, 2000**

**K. Chen et al., Mathematical explorations with MatLab,
Cambridge University Press 1999**

**D. Hanselman et al., MatLab 5 -- Guia do Usuário,
Editora Makron 1999**

conteúdo

- Arquivos: `fopen()`, `fclose()`, `fscanf()`, `fprintf()`
- `for-end`, `while`, `if`, `switch`
- Exemplo de gráfico 2D
- Scripts e funções
- Exemplo de função - `MaxVetor`
- `nargin`, `nargout`
- Outro exemplo de gráfico 2D
- Exemplo de sobreposição de gráficos 2D
- Números aleatórios - `rand()`
- `randperm()`, `unirand()`, `normrand()`, `exprand()`

fopen, fscanf, fclose

Forma geral de fopen é

```
fopen('nome arquivo', 'permissão')  
onde a permissão pode ser  
(em binary mode)  
'r'(read), 'w' (write), 'a' (append)  
(em text mode)  
'rt'(read), 'wt' (write), 'at' (append)
```

% Forma geral de fscanf é

```
fscanf(Na, '%f ou %d', Nnúmeros)  
fscanf(Na, '%f ou %d', [nlin,ncol])  
Se não houver Nnúmeros ou [nlin,ncol],  
leitura é até o fim do arquivo.
```

```
Narqu= fopen('c:\matentra.txt', 'rt') % abre  
[Matriz, contador]= fscanf(Narqu, '%f', [2,4])  
fclose(Narqu); % fecha
```

MatLab (Routo)

3

fopen, fscanf, fclose

```
Narqu= fopen('c:\matentra.txt', 'rt') % abre  
[Matriz, contador]= fscanf(Narqu, '%f', [2,4])  
fclose(Narqu); % fecha
```

arquivo 'c:\matentra.txt'

```
1.1 2.2 3.3 4.4  
5.5 6.6 7.7 8.8  
9.9 10 11.1 12.2
```

```
Narqu = 3  
Matriz =  
 1.1000  3.3000  5.5000  7.7000  
 2.2000  4.4000  6.6000  8.8000  
contador = 8
```

MatLab (Routo)

4

fopen, fscanf, fclose

```
Narqu2=fopen('c:\matsai.txt','wt')
contador=fopen(Narqu2,'%4.2f %4.1f %5.2f \n',Matriz)
fclose(Narqu2)
```

Arquivo 'c:\matsai.txt'

```
1.10 2.2 3.30
4.40 5.5 6.60
7.70 8.8
```

```
Narqu2 = 3
contador = 44
ans = 0
```

contador é o número de bytes gravados
Matriz é transformado em vetor no arquivo

Matlab possui os comandos de fluxo

```
for -- end,
while -- end
if -- elseif -- else -- end
```

que têm sintaxe parecida com a da C, exemplificada a seguir.

Lembre-se de *não* escrever a palavra elseif como duas palavras separadas.

```
n=9;m=3 ;
for i=1:n
  for j=1:m
    H(i,j)=1/(i+j+1);
  end
end
```

```
n=9; fatn=1;
while(n>1)
  fatn=fatn*n;
  n=n-1;
end
```

```
x=-1.1 ;
if(x<0)
  sinx= -1;
elseif(x>0)
  sinx=1;
else
  sinx=0;
end
```

Matlab possui também o comando
switch-case
que também tem sintaxe parecida com a da C, exemplificada a seguir.

```
switch <expressão>
case <teste-expressão-1>
    comandos-1 ...
case { <teste-expr-2, teste-expr-3,... }
    comandos-2 ...
...
otherwise
    comandos-3 ...
end
```

Resposta:
y = 68.8976

```
x=1.75; % x em metros
unidade= 'pol';
switch unidade % esse switch p/ converter x para
case{'polegadas','pol'} % y em polegadas
    y= x/2.54*100
case{'centímetros','cm'} % y em centímetros
    y=x/100
otherwise
    disp(['Unidade desconhecida'])
end
```

MatLab (Routo)

7

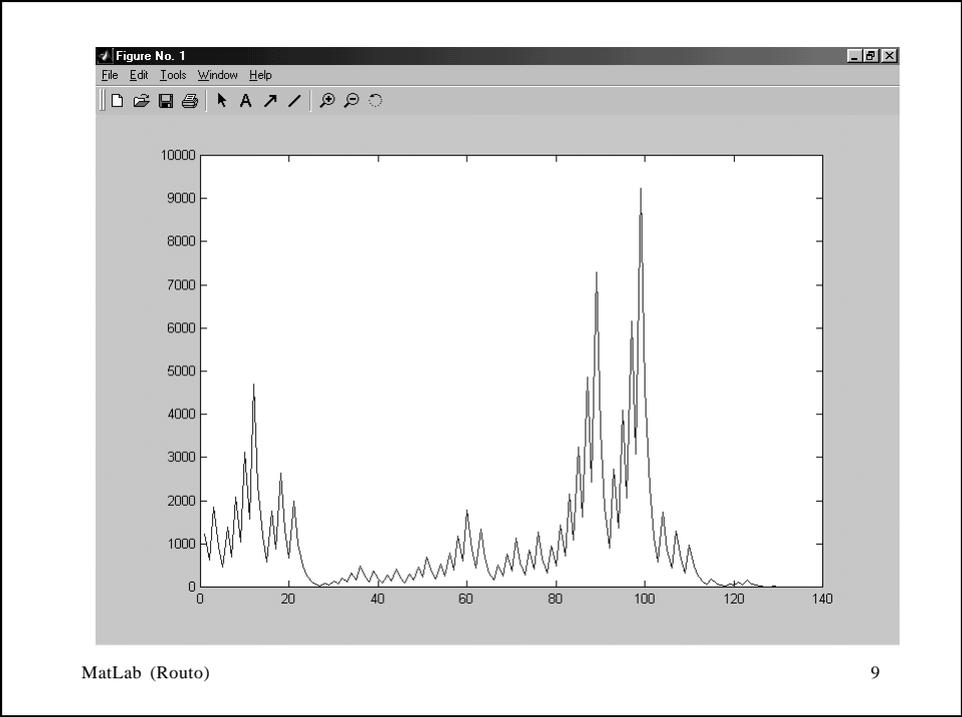
Exemplo de gráfico 2D: "sobe-e-desce"

```
x=zeros(140,1);% reserva memória para vetor 140x1
x(1)= round(abs(input('Digite um número > ')));
k=1;
while(x(k)>1) & (k<140)
    if rem(x(k),2) ==0 % x(k) é par
        x(k+1)= x(k)/2; % desce
    else % x(k) é ímpar
        x(k+1)= 3*x(k)+1; % sobe
    end
    k=k+1;
end
gradex=1:1:140; % vetor c/ 1,2,3,...140
plot(gradex,x) % gráfico do vetor x gerado
```

Gráfico a seguir para valor x(1)=1234

MatLab (Routo)

8



Scripts e funções

Subrotina (macro) deve ser guardado como M-file no sub-diretório MATLAB\work
Editar e depurar na janela M-file editor/debugger

Há dois tipos: scripts e funções

Script é apenas uma seqüência de comandos que são executados como se tivessem sido digitados na janela MatLab

$A = [1 \ 2; 3.1 \ 4.2], \det(A)$

```

» matrou1
A =
  1.0000  2.0000
  3.1000  4.2000
ans =
-2.0000

```

arquivo *matrou1.m* já editado

MatLab (Routo) 10

Scripts e funções

Função deve começar com a declaração da forma:

```
function [ps1,ps2,...psm]=nomefunc(pe1,pe2,...pen)
```

parâmetros de saída

parâmetros de entrada
(variáveis locais à função)

arquivo *AREACirc.m* já editado:

```
function [area] = AREACirc(raio)  
area=pi*raio*raio;
```

Na janela MatLab:

```
» A=AREACirc(2)
```

valor 2 passado para
'raio'

```
A = 12.5664
```

MatLab (Routo)

11

Scripts e funções

Chamada de função:

```
[as1,as2,...asm]=nomefunc[ae1,ae2,...aen]
```

Os argumentos *ae1,ae2,...aen* são passados por valor aos (i.e.,
copiados nos) parâmetros de entrada

arquivo *Circulo.m*:

```
function [area,perimetro] = Circulo(raio)  
area=pi*raio*raio;perimetro=2*pi*raio;
```

```
» [A,P]=Circulo(3.14)
```

```
A = 30.9748
```

```
P = 19.7292
```

janela MatLab

MatLab (Routo)

12

Exemplo de função

```
function [elemax, indmax] = MaxVetor(vetor)
% procura o elemento máximo dentro de um
% vetor linha ou coluna, em valor absoluto
% elemax -- elemento max em valor absoluto
% indmax -- índice do elemax
[m,n]=size(vetor);
if (m ~=1 & n ~= 1)
    erro; % não é vetor
else
    ultind=max( [m,n] );
    elemax = abs(vetor(1)); indmax= 1;
    for k = 2:ultind
        if( abs(vetor(k))>elemax)
            elemax= abs(vetor(k)); indmax= k;
        end % if
    end %for
end %else
```

MatLab (Routo)

13

Exemplos de execução de MaxVetor ()

```
» [elem,ind]=MaxVetor([-2.2 1.1 -4.4 2.2 0])
elem =
    4.4000
ind =
    3
```

```
» [elem,ind]=MaxVetor([-2.2 1.1; -4.4 2.2])
??? Undefined function or variable 'erro'.
```

Error in ==> C:\MATLABR11\work\MaxVetor.m
On line 8 ==> erro; % não é vetor

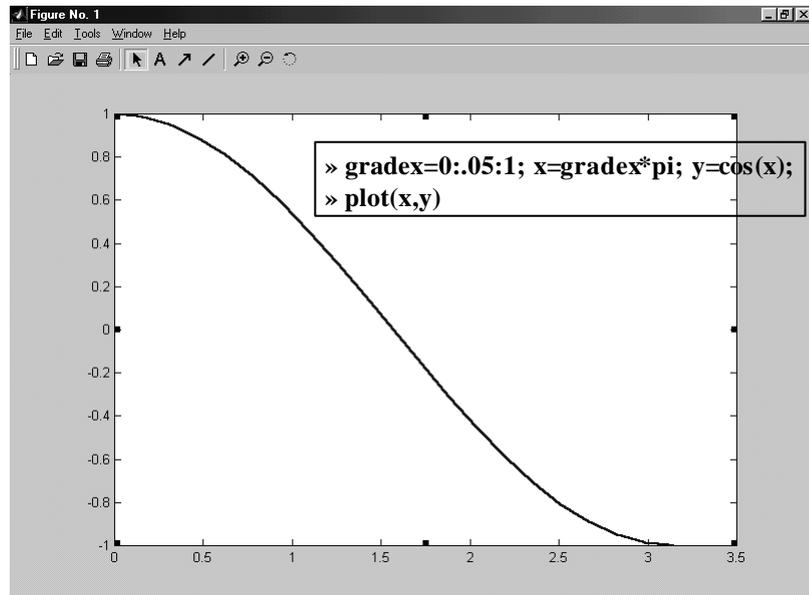
não é vetor

MatLab (Routo)

14

nargin nomefunc resulta número de args de entrada
da função *nomefunc*
nargout nomefunc idem para args de saída

```
» nargout AREACirc  
ans = 1  
» nargin AREACirc  
ans = 1  
» nargin Circulo  
ans = 1  
» nargout Circulo  
ans = 2
```

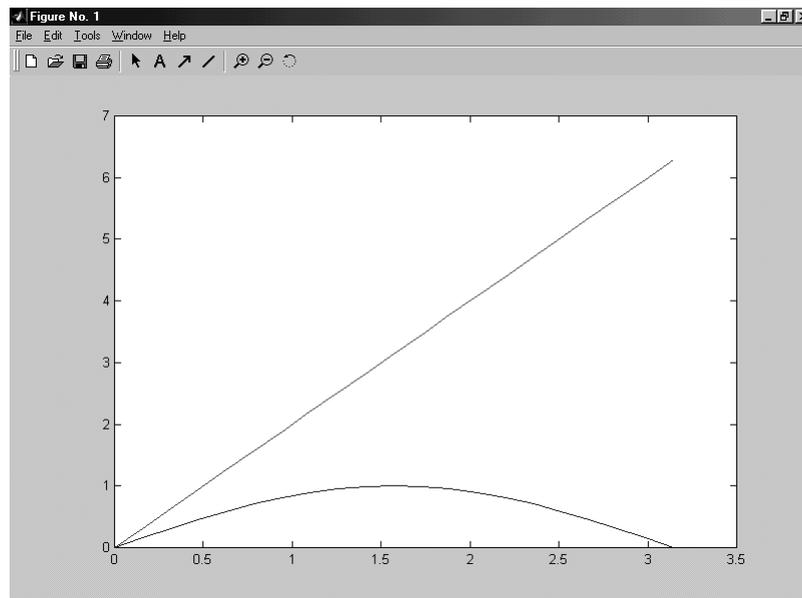


Sobreposição de gráficos 2D

```
gradex= 0:.05:1; x=gradex*pi;  
title('Gráfico de y=seno(x)')  
xlabel('eixo x')  
ylabel('eixo y')  
y= sin(x)  
plot(x,y)  
y1=2*x  
hold on % para "segurar" o gráfico  
plot(x,y1,'r') % r de red; sobrepõe  
% o gráfico de y1
```

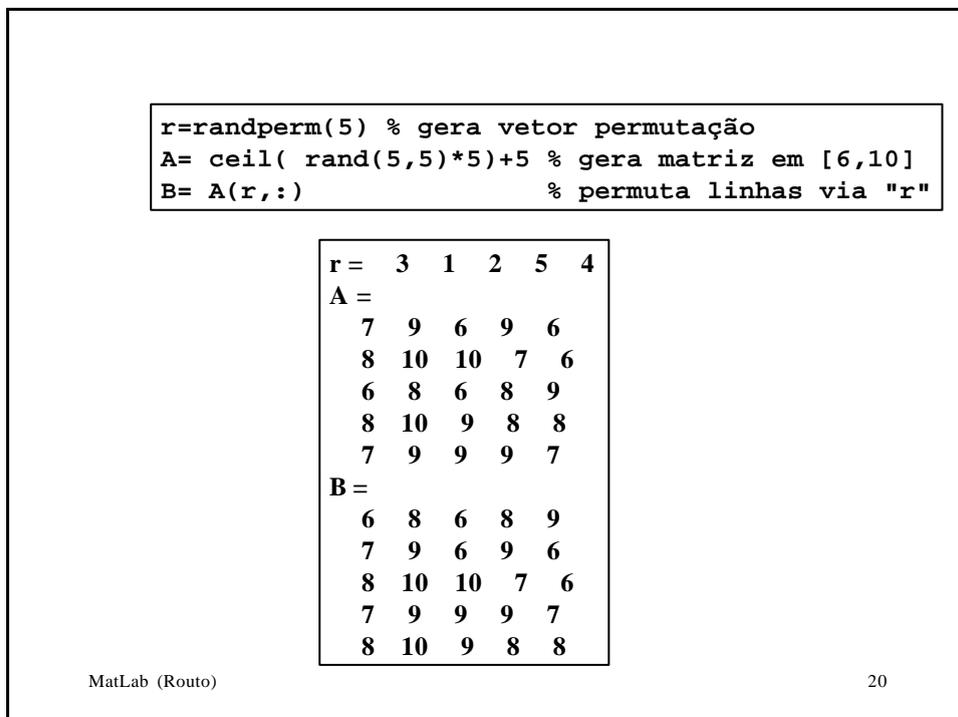
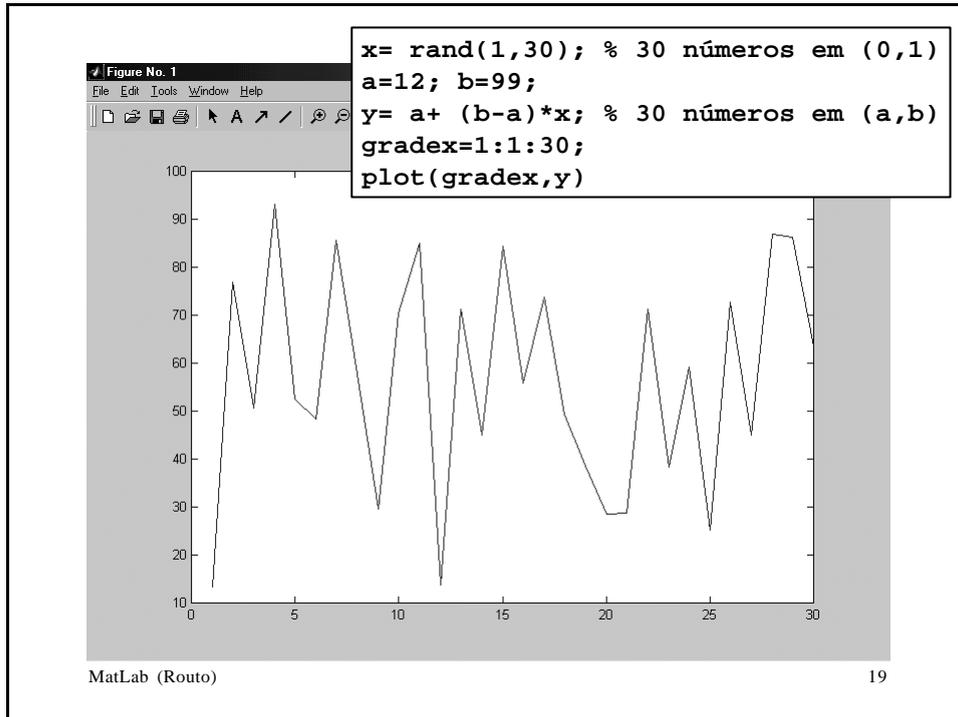
MatLab (Routo)

17



MatLab (Routo)

18



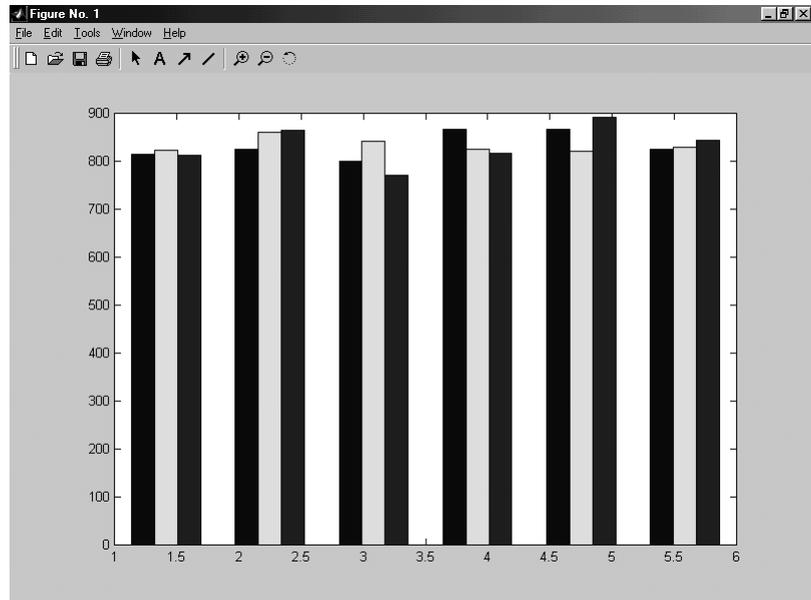
Distribuição uniforme (*unirand()*)

```
rand('seed',19); % semente =19
z=unirand(0.5,6.499,5000,3);
    % gera matriz 5000 por 3
    % em [0.5,6.499], uniforme
r=round(z);
hist(r,6) % 3 histogramas c/ intervalos iguais
```

```
function values=unirand(a,b,n,m);

% Usage:V = unirand( a,b)      % for 1 x 1
%       or V = unirand( a,b, n) % for n x n
%       or V = unirand( a,b, n, m) %for n x m
% computes a matrix of size n x m,
% uniformly distributed in (a,b)

if nargin < 2, help unirand, return, end
if nargin == 2,
    n=1; m=1;
elseif nargin == 3,
    m=n;
end
z=rand(n,m);
z=z*(b-a)+a;
values=z;
```



MatLab (Routo)

23

Distribuição normal *normrand()*

```

randn('seed',12);% semente da distr normal
y= normrand(3,5,4000,2);
    % gera 2 normais c/ méd=3,desv=5
    % 4000 números
hist(y)

```

MatLab (Routo)

24

```

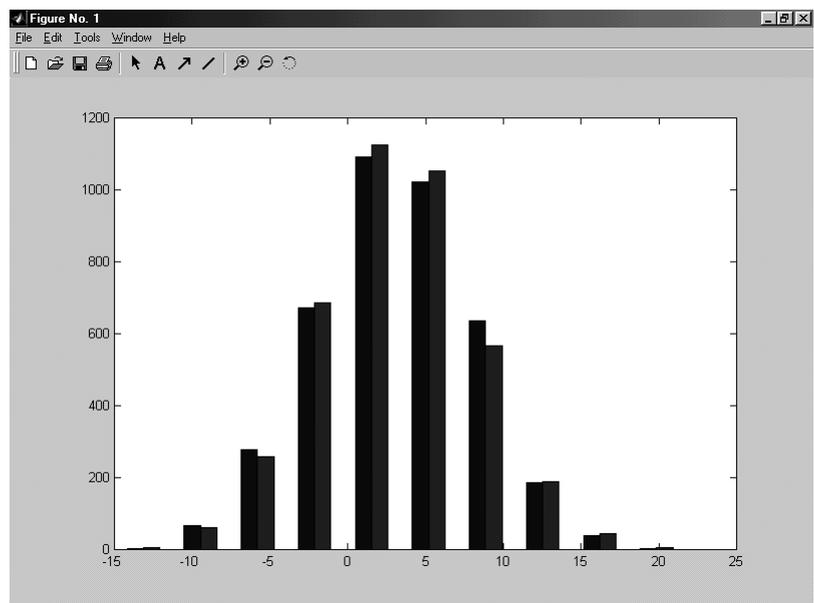
function values=normrand(mu,sigma,n,m);
% Usage:  V = normrand( mu, sigma)
% for 1 x 1
% or    V = normrand( mu, sigma, n)
% for n x n
% or    V = normrand( mu, sigma, n, m)
% for n x m
%
% computes a matrix of size n x m,
% normally distributed
% with mean mu and
% standard deviation sigma.

if nargin < 2, help normrand, return,
end
if nargin == 2,
    n=1; m=1;
elseif nargin == 3,
    m=n;
end
z=randn(n,m);
z=mu+z*sigma;
values=z;

```

MatLab (Routo)

25



MatLab (Routo)

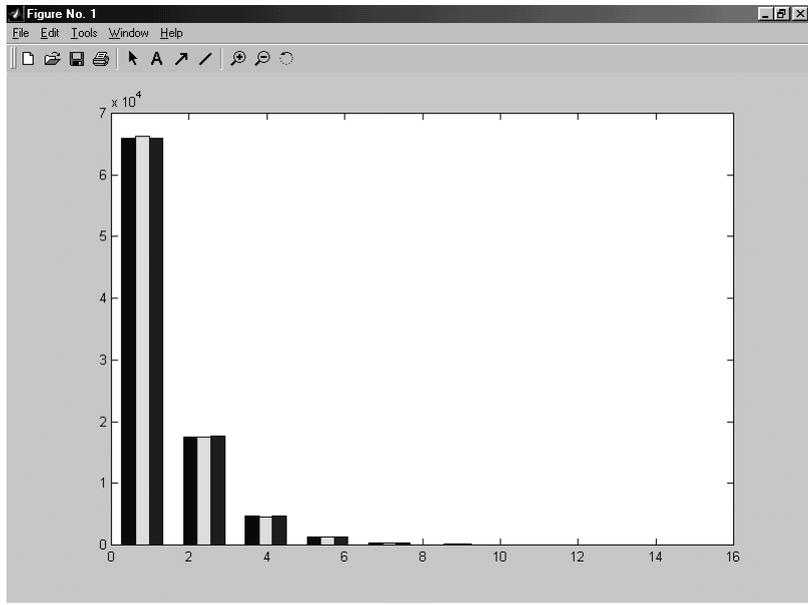
26

Distribuição exponencial (*exprand()*)

```
rand('seed',7);% semente
y= exprand(1.2,90000,3);
    % gera 1 exponencial c/ méd=1.2
hist(y)
```

```
function values=exprand(mu,n,m);
% Usage:  V = exprand( mu)
% for 1 x 1
%   or   V = exprand( mu, n)
% for n x n
%   or   V = exprand( mu, n, m)
% for n x m
%
% computes a matrix of size n x m,
%   exponentially distributed
%   with mean mu

if nargin < 1, help exprand, return, end
if nargin == 1,
    n=1; m=1;
elseif nargin == 2,
    m=n;
end
z=rand(n,m);
z=-log(1-z)*mu;
values=z;
```



MatLab (Routo)

29