

Notas de Aula

MatLab - 3

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

- Estatística básica
- Ordenação (sort) e histograma
- Gráfico (plot) após sort()
- Seleção de dados ("query"); critério de seleção
- Exemplo de seleção: primeiro quadrante
- Ajuste de curva pelo método dos mínimos quadrados -- MMQ
- MMQ usando A\y
- MMQ usando polyfit() e polyval(): exemplo de grau 1, 2, e 3
- MMQ de 2 variáveis usando polyfit2() e polyval2()

Estatística básica

```

%estatística básica
D=[2.2 1.1 3.3 4.4 5.5]
miD=min(D)
MAD=max(D)
mediaD=mean(D)
medianaD=median(D)
produtoD=prod(D)
somaD=sum(D)
somaAcumulD=cumsum(D)
desvioD=std(D)
ME=[1.1 4.4;2.2 3.3;4.4 5.5;5.5 6.6]
coefCorrelacaoME= corrcoef(ME)
% cada coluna é uma variável
covarianciaME=cov(ME)
somAcumulME=cumsum(ME)
% soma acumulada por coluna
    
```

```

D = 2.2000 1.1000 3.3000 4.4000 5.5000
miD = 1.1000
MAD = 5.5000
mediaD = 3.3000
medianaD = 3.3000
produtoD = 193.2612
somaD = 16.5000
somaAcumulD = 2.2000 3.3000 6.6000 11.0000 16.5000
desvioD = 1.7393
ME =
    1.1000 4.4000
    2.2000 3.3000
    4.4000 5.5000
    5.5000 6.6000
coefCorrelacaoME =
    1.0000 0.8485
    0.8485 1.0000
covarianciaME =
    4.0333 2.4200
    2.4200 2.0167
somAcumulME =
    1.1000 4.4000
    3.3000 7.7000
    7.7000 13.2000
    13.2000 19.8000
    
```

MatLab (Routo)

3

```

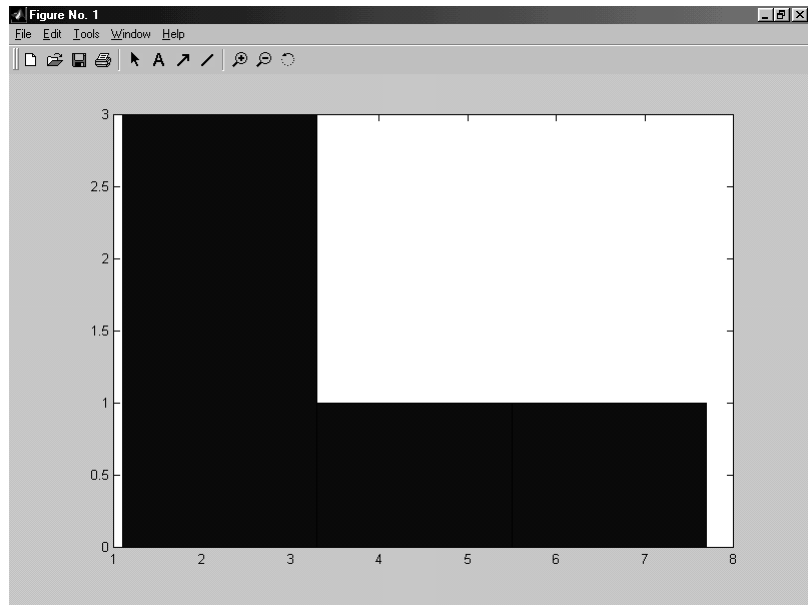
%%%%%%%%% sort() %%%%%%%%%
D= [2.1 1.1 4.4 7.7 3.3]
DAscend= sort(D) % ordena
DDescen= -sort(-D) % ordem decendente
%%%%%%%%% histograma %%%%%%%%%
hist(D,3) % default de 10 intervalos
[y x]=hist(D,3) % 3 intervalos: 1<=x<=3, 3<x<=5, 5<x<=7
    
```

```

D = 2.1000 1.1000 4.4000 7.7000 3.3000
DAscend = 1.1000 2.1000 3.3000 4.4000 7.7000
DDescen = 7.7000 4.4000 3.3000 2.1000 1.1000
y = 3 1 1
x = 2.2000 4.4000 6.6000
    
```

MatLab (Routo)

4



MatLab (Routo)

5

```

%%%%%%%%%% plot() %%%%%%%%%%%
% x:fertilizante, y:produção
x=[2 6 4 8 12 10 18 0 16 14]';
y=[4.70 4.84 5.02 5.38 5.47 5.61 5.95 4.01 5.90 5.80]';
[sx k]= sort(x); % ordenado em sx, índices originais em k
sy=y(k) % reordena y para ser coerente com x original
[x y k sx sy]
plot(sx,sy,'g*') % g de green, *usada no gráfico

```

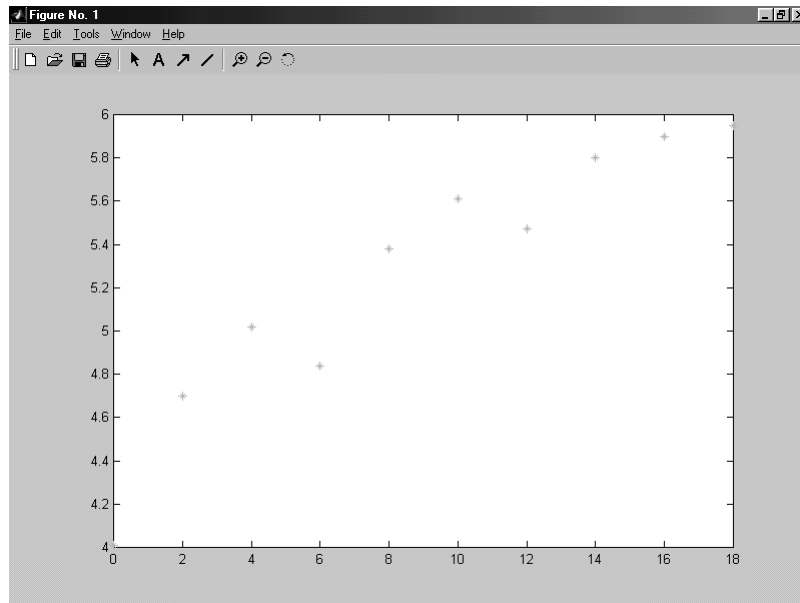
```

sy = 4.0100 4.7000 5.0200 4.8400
5.3800 5.6100 5.4700 5.8000 5.9000 5.9500
ans =
2.0000 4.7000 8.0000 0 4.0100
6.0000 4.8400 1.0000 2.0000 4.7000
4.0000 5.0200 3.0000 4.0000 5.0200
8.0000 5.3800 2.0000 6.0000 4.8400
12.0000 5.4700 4.0000 8.0000 5.3800
10.0000 5.6100 6.0000 10.0000 5.6100
18.0000 5.9500 5.0000 12.0000 5.4700
0 4.0100 10.0000 14.0000 5.8000
16.0000 5.9000 9.0000 16.0000 5.9000
14.0000 5.8000 7.0000 18.0000 5.9500

```

MatLab (Routo)

6



MatLab (Routo)

7

Importante : seleção de dados

```

%%%%%%%% query %%%%%%%%%
d= (y>=5) % query só daqueles que satisfazem y>=5
      % 1 se satisfaz, 0 c.contrário
t= sum(d) % número dos que satisfizeram query
[v k]= sort(-d); % traz os que satisfizeram para o início
k=k(1:t) % encurta o vetor k para t elementos
x1= x(k); y1= y(k); % escolhe só os que satisfizeram
[x1 y1]
plot(x1,y1,'r+')

```

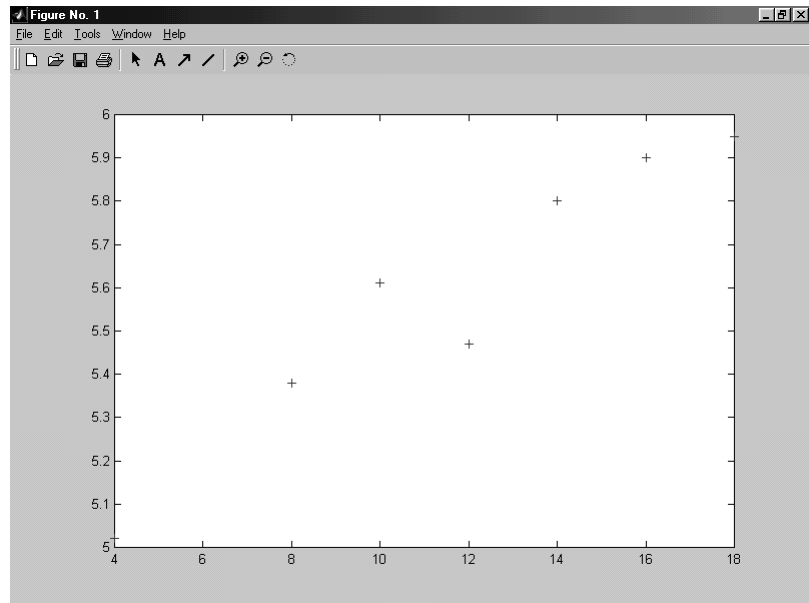
```

d=
0
0
1
1
1
1
1
1
0
1
1
1
1
1
10
ans =
4.0000 5.0200
8.0000 5.3800
12.0000 5.4700
10.0000 5.6100
18.0000 5.9500
16.0000 5.9000
14.0000 5.8000

```

MatLab (Routo)

8



MatLab (Routo)

9

```

%% criar 42 pontos (x,y) e selecionar só os que estão
%% no segundo quadrante
t=0: 0.15: 2*pi;
nconta= length(t) % número de pontos
xa=cos(t); ya=sin(t); % cria os 42 pontos
d= (ya>=0) & (xa<=0); % seleciona só no 2o. quadrante
t=sum(d)
[v i]= sort(-d);
i= i(1:t)
x1= xa(i); y1= ya(i);
plot(x1,y1, 'b*')

```

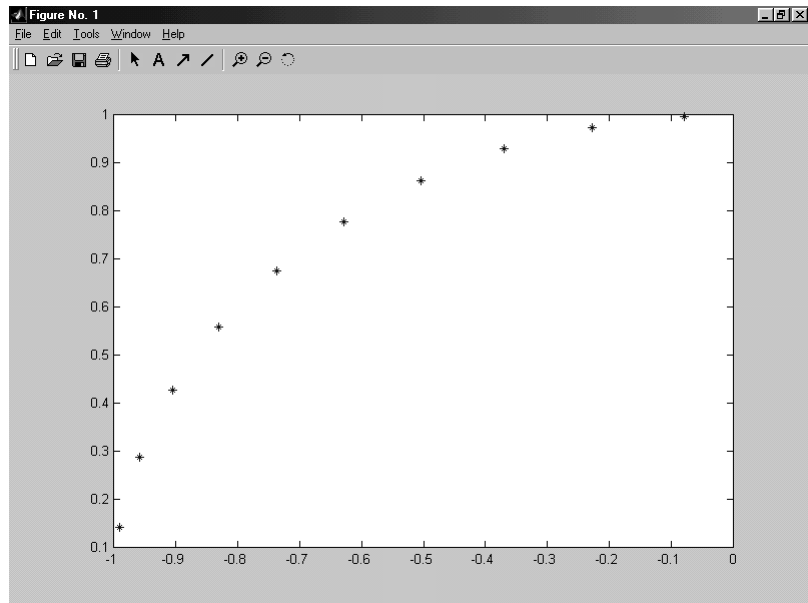
```

nconta =
    42
t =
    10
i =
    12 13 14 15 16 17 18 19 20 21

```

MatLab (Routo)

10



MatLab (Routo)

11

```

%% método dos mínimos quadrados MMQ
x=[2 6 4 8 12 10 18 0 16 14]';
y=[4.70 4.84 5.02 5.38 5.47 5.61 5.95 4.01 5.90 5.80]';

%% ajustar uma reta pelo MMQ
A= [ones(size(x)) x] % só 1 na primeira coluna
% ya_1 1 x_1
% ya_2 1 x_2
% ... = ..... c_1 c_2
% ya_n 1 x_n
c=A\y % solução de Ac=y representa reta ya_j=c_1+c_2 x_j
ya= c(1)+c(2)*x;
plot(x,y,'r*',x,ya,'b')
% calcular residuo
yaux=A*c;
res=y-yaux;
ResiduoMMQ=res'*res % res transposto * res

```

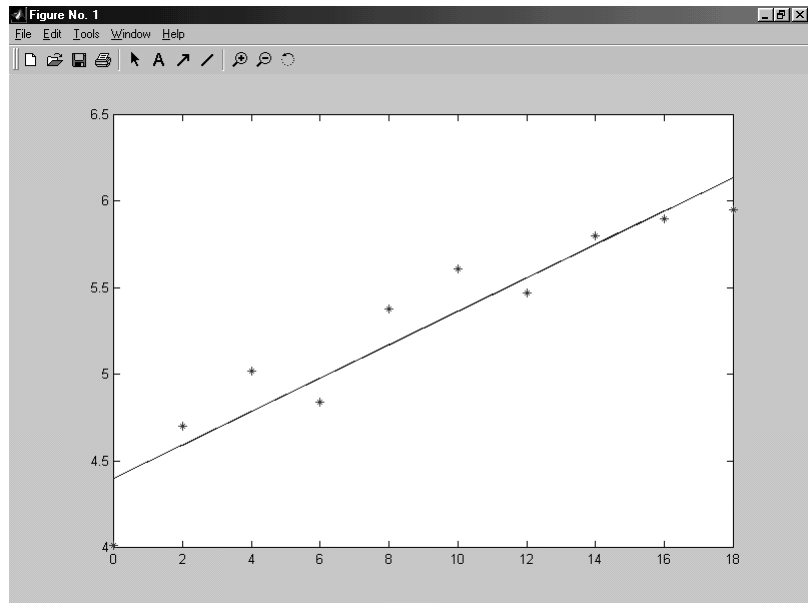
```

A=
1 2
1 6
1 4
1 8
1 12
1 10
1 18
1 0
1 16
1 14
c=
4.3985
0.0966
ResiduoMMQ =
0.3880

```

MatLab (Routo)

12



MatLab (Routo)

13

```

%%% polyfit() e polyval() para MMQ
Coef=polyfit(x,y, 1) % 1 significa reta, i.e., polin. grau 1
%      ya_j = Coef_2 + Coef_1 x_j Coef é um vetor linha!
%
% para calcular resíduo, use polyval()
ya=polyval(Coef,x);
res=y-ya;
ResiduoMMQ=res'*res

```

```

Coef =
    0.0966    4.3985
ResiduoMMQ =
    0.3880

```

MatLab (Routo)

14

```

% polyfit() aceita qualquer grau k>=1
%% polyfit() de grau 2 para MMQ
Coef=polyfit(x,y, 2)
%   ya_j = Coef_3 + Coef_2 x_j + Coef_1 (x_j)^2
%
% para calcular resíduo, use polyval()
ya=polyval(Coef,x);
res=y-ya;
ResiduoMMQ=res'*res

[sx k]= sort(x); % ordenado em sx, índices originais em k
sya=ya(k) % reordena y para ser coerente com x original
sy=y(k);
plot(sx,sy,'b*',sx,ya(k),'r')

```

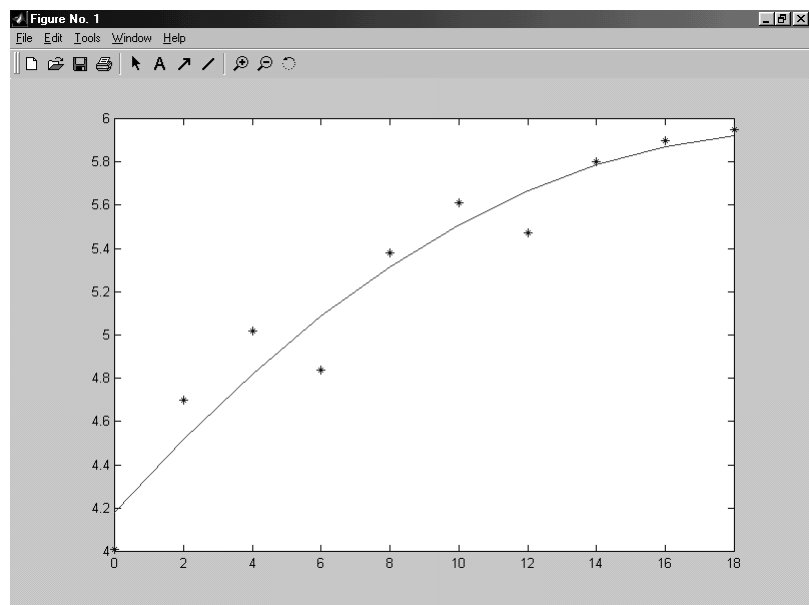
```

Coef = -0.0045  0.1776  4.1826
ResiduoMMQ =  0.2170
sya =
  4.1826
  4.5198
  4.8210
  5.0861
  5.3153
  5.5085
  5.6658
  5.7870
  5.8723
  5.9215

```

MatLab (Routo)

15



MatLab (Routo)

16


```

%% polyfit() de grau 3 para MMQ
Coef=polyfit(x,y, 3)
% ya_j = Coef_4 + Coef_3 x_j + Coef_2 (x_j)^2 + Coef_1 (x_j)^3
% para calcular residuo, use polyval()
ya=polyval(Coef,x);
res=y-ya;
ResiduoMMQ=res'*res

[sx k]= sort(x); % ordenado em sx, índices originais em k
sya=ya(k) % reordena y para ser coerente com x original
sy=y(k);
plot(sx,sy,'b*',sx,ya(k),'r')

```

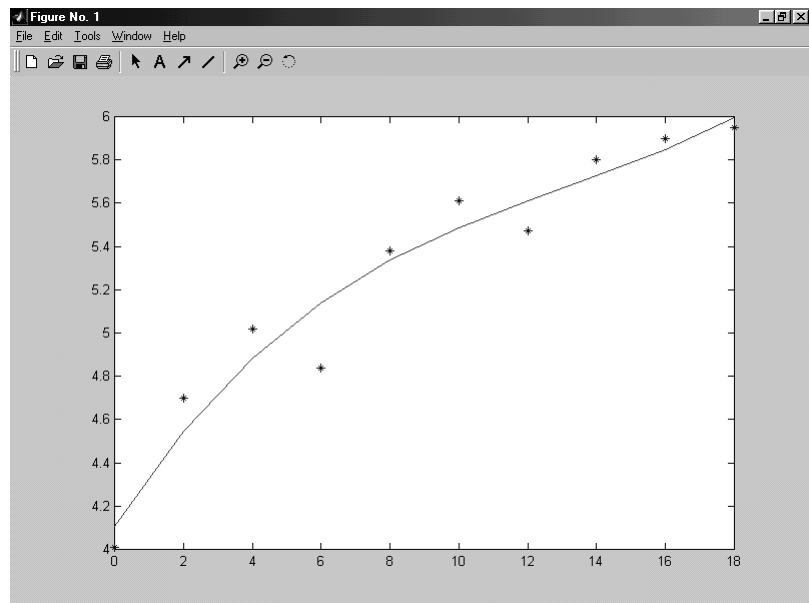
```

Coef = 0.0004 -0.0144 0.2451 4.1088
ResiduoMMQ = 0.1905
sya =
4.1088
4.5444
4.8825
5.1407
5.3364
5.4874
5.6113
5.7255
5.8477
5.9954

```

MatLab (Routo)

17



MatLab (Routo)

18

```

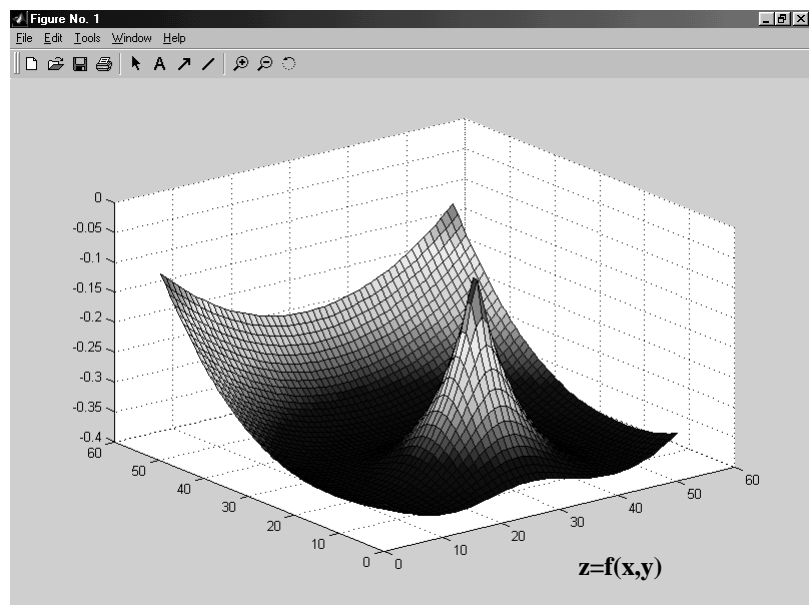
%% m3o m3to MQ para 2 vari3veis m3o m3o m3o
%% polyfit2 e polyval2 em disquete m3o m3o
x0= 0:0.02:1; y0=x0;
[x y]= meshgrid(x0,y0);
R=sqrt( (x-1/2).^2 + (y-1/4).^2);
z=R.*log(R+eps); % criou dados z em fun33o de x e y
C1= polyfit2( x,y,z, 'linear'); % MMQ linear
C2= polyfit2( x,y,z, 'cubic' ); % MMQ c3bico
% a seguir c3culos surf plot e mesh plot
Z1= polyval2( C1, x,y, 'linear');
Z2= polyval2( C2, x,y, 'cubic');
surf(z); hold on;
disp('pausa'); pause;
mesh(Z2);
% c3culo do res3duo
resMMQ=sum(sum((Z2-z)'*(Z2-z)))

```

resMMQ =
12.2072

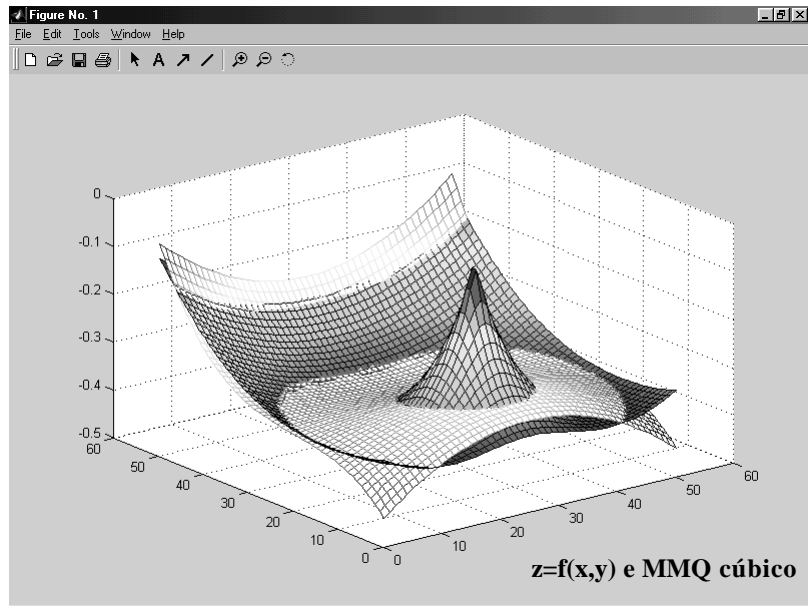
MatLab (Routo)

19



MatLab (Routo)

20



MatLab (Routo)

21

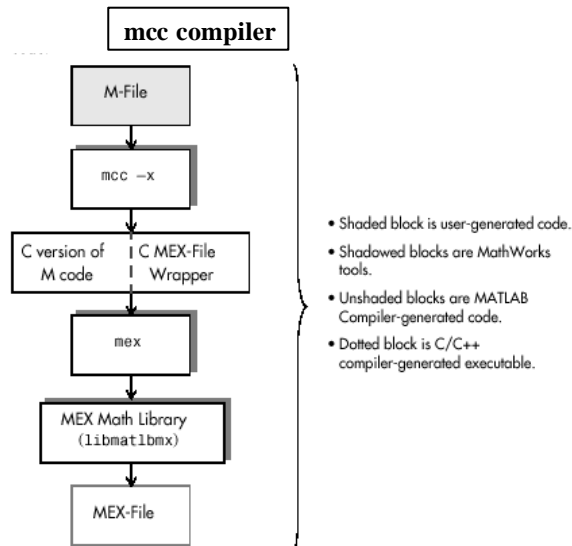


Figure 1-1: Developing MEX-Files

MatLab (Routo)

22

Statistics

| | |
|---|-------|
| Introduction | 1-2 |
| Probability Distributions | 1-5 |
| Descriptive Statistics | 1-42 |
| Cluster Analysis | 1-50 |
| Linear Models | 1-65 |
| Nonlinear Regression Models | 1-79 |
| Hypothesis Tests | 1-85 |
| Multivariate Statistics | 1-91 |
| Statistical Plots | 1-103 |
| Statistical Process Control (SPC) | 1-110 |
| Design of Experiments (DOE) | 1-115 |
| Demos | 1-125 |
| References | 1-134 |

The Financial Toolbox. The Financial Toolbox operates with MATLAB to provide a robust set of financial functions essential to financial and quantitative analysis. Applications include pricing securities, calculating interest and yield, analyzing derivatives, and optimizing portfolios. The Financial Toolbox requires the Statistics and Optimization Toolboxes. The Simulink graphical interface is recommended for Monte Carlo and non-stochastic simulations for pricing fixed-income securities, derivatives, and other instruments.

The Statistics Toolbox. The Statistics Toolbox provides a set of M-files for statistical data analysis, modeling, and Monte Carlo simulation, with GUI-based tools for exploring fundamental concepts in statistics and probability.

The Symbolic Math Toolbox. The Symbolic Math Toolbox gives MATLAB an integrated set of tools for symbolic computation and variable-precision arithmetic, based on Maple V®. The Extended Symbolic Math Toolbox adds support for Maple programming plus additional specialized functions.

