

Lecture 13: Matlab 简介(二)

张伟平

Monday 7th December, 2009

Contents

1	Programming	1
1.1	M file	1
1.1.1	Program Control Statements	2
1.1.2	M-File Functions	13
1.2	anonymous functions	18
2	Computational statistics with Matlab	21
2.1	Functions on Probability and Statistics	21
2.1.1	Probability distribution	21
2.1.2	Descriptive statistics	23
2.1.3	Statistical plotting	24
2.1.4	Linear model	24
2.1.5	Multivariate Statistics	26
2.2	Monte Carlo with Matlab	28
2.2.1	Monte Carlo Assessment of Hypothesis Testing	28
2.2.2	MCMC with matlab	31

3	Symbolic computation with matlab	37
3.1	Creating Symbolic Variables and Expressions	37
3.2	Calculus	39
4	Optimization	42
4.1	Unconstrained Minimization Example	42
4.2	Nonlinear Inequality Constrained Example	43

Chapter 1

Programming

Matlab 有两种命令执行方式: 一种就是在命令窗口进行交互式命令执行. 用户输入一个命令, Matlab解释 执行一个命令. 这种方式在有些场合(比如需要循环)是不方便的; 另一种执行方式就是M文件执行方式. 用户将所有命令放在一个文件里(称为M文件, 以`.m`为扩展名), matlab执行该文件里所有命令直至结束.

1.1 M file

用matlab语言编写的程序称为M文件. 它是由若干个matlab命令构成的, 用以实现某些操作或者某个算法. M文件有两类, 一类称为命令文件(Script file), 一类称为函数文件(Function file). 它们的扩展名都为`.m`. 区别在于

- 命令文件没有输入参数, 也不返回输出参数; 函数文件可以有输入参数, 也可以有返回输出参数.
- 命令文件可以在命令窗口里输入该文件名称, 即可运行里面的命令; 函数文件则不行, 须以函数调用的方式来调用此函数.
- 命令文件对工作空间里的变量直接操作, 结果也完全返回到工作空间里. 而函数文件里定义的变量为局部变量, 函数执行完毕后就被清除.

1.1.1 Program Control Statements

程序的控制分为下面几种:

❁ 顺序结构

顺序结构是最简单的程序结构, 程序中的语句将按照顺序依次执行, 直至最后一个语句. 一般涉及到数据的输入, 数据的计算, 数据的输出.

input 输入语句 语法结构

`input` (提示信息, 选项)

[↓Code](#)

其中提示信息为一个字符串, 用于提示用户输入什么样的数据, 其中可以加多个换行符`\n`. 例如

[↑Example](#)

```
A=input('输入A矩阵:\n')
```

[↓Example](#)

执行该语句后, 首先在屏幕上显示'输入A矩阵', 然后等用户从键盘上输入该矩阵. `input` 语句 不加选项时表示输入数值, 而当选项为's'时, 则允许用户输入一个字符串. 例如

[↑Example](#)

```
nm=input('what''s your name?', 's')
```

[↓Example](#)

disp 输出语句 `disp`语句用于向命令窗口显示指定信息. 例如

```
A='statistics';  
disp(A)  
A=[1,2,3;2,4,5];  
disp(A)
```

[↑Example](#)

[↓Example](#)

pause 暂停语句

语法

```
pause #程序暂停, 等待用户按任何键以继续  
pause(n) # 暂停n秒  
pause on # 使得下续的pause或pause(n)执行  
pause off #使得下续的pause或pause(n)不执行
```

[↑Code](#)

✿ 条件控制— if, switch

if, else, and elseif 语句

语法结构

```
if logical_expression
    statements
elseif logical_expression
    statements
else
    statements
end
```

例如


```
if rem(a, 2) == 0
    disp('a is even')
    b = a/2;
end
```

[↓Example](#)

以及

```
if n < 0                % If n negative, display error message.
    disp('Input must be positive');
elseif rem(n,2) == 0 % If n positive and even, divide by 2.
    A = n/2;
else
    A = (n+1)/2;      % If n positive and odd, increment and divide.
end
```

[↑Example](#)

[↓Example](#)

switch 语句

switch 语句根据表达式值的不同执行不同的语句. 语法结构

```
switch expression (scalar or string)
  case value1
    statements          % Executes if expression is value1
  case value2
    statements          % Executes if expression is value2
  .
  .
  .
  otherwise
    statements          % Executes if expression does not
                       % match any case
end
```

[↑Code](#)

[↓Code](#)

例如

```
switch var
```

[↑Example](#)

```
case 1
    disp('1')
case {2,3,4}
    disp('2 or 3 or 4')
case 5
    disp('5')
otherwise
    disp('something else')
end
```

[↓Example](#)

❁ 循环控制– for, while, continue, break

for 语句

语法结构

```
for index = start:increment:end
    statements
end
```

[↑Code](#)

[↓Code](#)

例如

[↑Example](#)

```
for n = 2:6
    x(n) = 2 * x(n - 1);
end
%-----
for m=1:5
    for n = 1:100
        A(m, n) = 1/(m + n - 1);
    end
end
%-----
for k=A
    statements
end
```

[↓Example](#)

最后一个例子里A表示一个数组. 比如当A为 $m \times n$ 矩阵时, 语句`statements`循环 n 次, 每次将矩阵A的一列赋给 k .

while 语句 语法结构

```
while expression
    statements
end
```

[↑Code](#)

[↓Code](#)

例如

```
n = 1;
while prod(1:n) < 1e100
    n = n + 1;
end
```

[↑Example](#)

[↓Example](#)

在for,while 循环里, 可以使用continue,break两个语句来控制循环在满足某个条件时, 跳到下一个循环, 或者终止循环. 例如

```
%计算100至200之间第一个被21整除的数
for n = 100:200
    if rem(n,21)~=0
        continue
    end
end
break
end
n
```

[↑Example](#)

[↓Example](#)

❁ 错误控制- try, catch

语法结构

```
try
```

[↑Code](#)

```
    语句组1
catch
    语句组2
end
```

[↓Code](#)

*try*语句先试探性的执行语句组1, 如果出现错误, 则将错误信息保留在变量*lasterr*里, 转去执行语句组2.

✿ 程序停止—*return* 语句

*return*语句可以使得程序在某些地方提前结束. 例如

```
function d = det(A)
%DET det(A) is the determinant of A.
if isempty(A)
    d = 1;
    return
else
    ...
end
```

[↑Example](#)

1.1.2 M-File Functions

函数文件是另一种M文件, 每个函数文件都定义一个函数. 事实上, matlab提供的大多数标准函数都是由函数文件定义.

函数文件的结构

函数文件由**function**语句引导, 其基本结构为

function 输出参数表=函数名(输入参数表)	函数定义行
%函数名, 基本功能	H1行
%详细的说明	help 文字
% 文件版本修改信息等	comment
函数体语句部分	

[↑Code](#)[↓Code](#)

❖ 函数的命名规则和变量相同, 当输出参数多于一个时, 要用方括号将输出参数括起来.

❖ 函数文件名通常由函数名加.m组成, 不过函数文件名也可以和函数名不同, 此时 matlab就会忽略函数名, 而采用函数文件名. 因此调用时使用函数文件名.

❖ 关于注释部分: H1行一般包括函数名及其基本功能描述, 用于lookfor关键词或者 help在线帮助时使用. help 文字部分用于在线帮助. 与help文字部分空一行的部分, 即comment部分, 用于 函数文件的作者, 版本, 修改等等信息.

例如求半径为 r 的圆的面积.

```
function [s p]=fcircle(r)
% FCIRCLE calculates the area and perimeter of a circle of radii r
% r 圆半径
% s 圆面积
% p 圆周长
```

[↑Code](#)

%2009.11.30日编写

```
s=pi*r*r;  
p=2*pi*r;
```

[↓Code](#)

将以上文件存为fcircle.m, 然后在命令窗口调用

```
[s p]=fcircle(10)  
s =  
    314.1593  
p =  
    62.8319
```

[↑Example](#)

可以使用help 和 lookfor 命令来查找此函数, 如help fcircle; lookfor perimeter;等.

[↓Example](#)

函数的调用

函数调用的格式为

[输出参数表]= 函数(输入参数表)

[↑Code](#)

[↓Code](#)

例如将直角坐标转换为极坐标:

```
% tran.m
function [rho theta]=tran(x,y)
rho=sqrt(x^2+y^2);
theta=atan(y/x);
```

[↑Example](#)

```
%引用tran.m的命令文件
x=input('Please input x=');
y=input('Please input y=');
[rho,theta]=tran(x,y);
rho
theta
```

[↓Example](#)

函数也可以嵌套调用, 比如计算 $n!$:

[↑Example](#)

```
% fact.m
function f=fact(n)
if n<=1
    f=1;
else
    f=fact(n-1)*n;
end
%-----
s=0;
for i=1:5
    s=s+fact(i);
end
s
```

[↓Example](#)

全局变量和局部变量

函数文件里定义的变量都是局部变量，不能在其他函数文件或工作空间里引用。可以使用命令 **global** 将某个变量定义为全局变量，即在整个工作空间里都

有效. 例如

```
% wadd.m
function f=wadd(x,y)
global alpha beta
f=alpha*x+beta*y;
%在命令窗口
global alpha beta
alpha=1; beta=2;
wadd(1,2)
```

[↑Example](#)

[↓Example](#)

1.2 anonymous functions

匿名函数可以方便快捷的创建(简单)函数, 而不需要创建M文件. 其使用语法为

[↑Code](#)

```
fhandle = @(arglist) expr
```

[↓Code](#)

例如

```
sqr = @(x) x.^2;  
a = sqr(5)  
quad(sqr, 0, 1) % integrate from 0 to 1
```

[↑Example](#)

```
sumAxBy = @(x, y) (A*x + B*y);  
sumAxBy(5, 7)
```

```
A = {@(x)x.^2, @(y)y+10, @(x,y)x.^2+y+10} %cell array  
A{1}(4) + A{2}(7)
```

```
a = 1.3;   b = .2;   c = 30;  
parabola = @(x) a*x.^2 + b*x + c;  
fplot(parabola, [-25 25])  
-----
```

```
a = -3.9;    b = 52;    c = 0;
fplot(parabola, [-25 25])
-----
a = -3.9;    b = 52;    c = 0;
parabola = @(x) a*x.^2 + b*x + c;
fplot(parabola, [-25 25])
```

[↓Example](#)

Chapter 2

Computational statistics with Matlab

2.1 Functions on Probability and Statistics

2.1.1 Probability distribution

我们列出Matlab里常用的有关概率分布的函数:

分布名称	概率函数(密度)	累积分布函数	随机数产生	分布函数的逆
离散均匀分布	unidpdf	unidcdf	unidrnd	unidinv
二项分布	binopdf	binocdf	binornd	binoinv
Poisson分布	poisspdf	poisscdf	poissrnd	poissinv
几何分布	geopdf	geocdf	geornd	geoinv
超几何分布	hygepdf	hygecdf	hygernd	hygeinv
负二项分布	nbnpdf	nbincdf	nbinnrnd	nbinninv

分布名称	概率函数(密度)	累积分布函数	随机数产生	分布函数的逆
正态分布	normpdf	normcdf	normrnd	norminv
指数分布	exppdf	expcdf	exprnd	expinv
均匀分布	unifpdf	unifcdf	unifrnd	unifinv
χ^2 分布	chi2pdf	chi2cdf	chi2rnd	chi2inv
t 分布	tpdf	tcdf	trnd	tinvs
F 分布	fpdf	fcdf	frnd	finvs
$Beta$ 分布	betapdf	betacdf	betarnd	betainvs
Γ 分布	gampdf	gamcdf	gamrnd	gaminvs
多元正态分布	mvnpdf		mvnrnd	

另外, 对每个概率分布, 计算其中的参数估计(点估计和区间估计)函数为该分布名后加"fit", 比如 `binofit`, `normfit`等等.

计算每个分布矩(期望和方差)可以使用函数分布名`stat`, 例如 `binostat`, `poisstat`, `normstat`等等.

例对画 Γ 分布的密度图形.

```
x = 0:.05:3;
```

↑Example

```
y1 = gampdf(x,1,1/1);
y2 = gampdf(x,2,1/2);
y3 = gampdf(x,3,1/3);
% Plot the functions.
plot(x,y1,'r',x,y2,'g',x,y3,'b')
title('Gamma Distribution')
xlabel('X')
ylabel('f(x)')
text(0.141,0.9,'\leftarrow\lambda=t=1','FontSize',18)
text(1.075,0.7,'\leftarrow\lambda=t=3','FontSize',18)
text(0.68,0.4,'\lambda=t=2\rightarrow','FontSize',18)
```

[↓Example](#)

2.1.2 Descriptive statistics

部分常用描述性统计量计算函数:

函数	作用	函数	作用
mean	均值	moment	中心矩
median	中位数	corrcoef	相关系数
var	方差	range	样本值区间
std	标准差	cov	样本协方差矩阵

2.1.3 Statistical plotting

部分常用统计绘图函数:

函数	作用	函数	作用
boxplot	盒图	lsline	对当前图上添加LS拟合直线
cdfplot	经验分布函数图	errorbar	绘制errorbar图
normplot	图形正态检验	qqplot	Q-Q图

2.1.4 Linear model

与线性模型有关的几个函数

函数	作用
regression	回归
regstats	回归诊断
ridge	岭回归
stepwisefit	使用逐步回归拟合
leverage	杠杆点

↑Example

```
load carsmall
x1 = Weight;
x2 = Horsepower; % Contains NaN data
y = MPG;

%Compute regression coefficients for
%a linear model with an interaction term:

X = [ones(size(x1)) x1 x2 x1.*x2];
b = regress(y,X) % Removes NaN data
Plot the data and the model:

scatter3(x1,x2,y,'filled')
hold on
x1fit = min(x1):100:max(x1);
```

```
x2fit = min(x2):10:max(x2);  
[X1FIT,X2FIT] = meshgrid(x1fit,x2fit);  
YFIT = b(1) + b(2)*X1FIT + b(3)*X2FIT + b(4)*X1FIT.*X2FIT;  
mesh(X1FIT,X2FIT,YFIT)  
xlabel('Weight')  
ylabel('Horsepower')  
zlabel('MPG')  
view(50,10)
```

[↓Example](#)

2.1.5 Multivariate Statistics

几个简单的多元分析函数

函数	作用
linkage	使用单连接法计算距离
cluster	使用linkage的输出创建类
kmeans	kmeans聚类
dendrogram	画聚类图
canoncorr	典则分析
classify	判别分析
pcacov	协方差矩阵的主成分分析

2.2 Monte Carlo with Matlab

2.2.1 Monte Carlo Assessment of Hypothesis Testing

通过Monte Carlo方法来计算经验的一型和二型错误率, 以此来评价一个检验.

例: 假设我们有数据

```
465 449 468 446 447 456 442 433 447 456 438 449
463 452 435 447 447 450 444 450 454 456 456 454 446
```

假设方差 $\sigma = 7.8^2$, 以此检验均值假设

$$H_0 : \mu = 454 \leftrightarrow H_1 : \mu < 454$$

显然这是一个标准的 Z 检验. 我们来通过Monte Carlo方法计算此检验的一型和二型错误率.

```
x=[465 449 468 446 447 456 442 433 447 456 438 449 ...
```

↑Example

```

463 452 435 447 447 450 444 450 454 456 456 454 446];
n=length(x);
M = 1000;
alpha = 0.05;
sigma=7.8; sigxbar=sigma/sqrt(n);
% Get the critical value, using z as test statistic.
cv = norminv(alpha,0,1);
% Start the simulation.
Im = 0;
for i = 1:M
% Generate a random sample under H_0.
xs = sigma*randn(1,n) + 454;
Tm = (mean(xs)-454)/sigxbar;
if Tm <= cv % then reject H_0
Im = Im +1;
end
end
alphahat = Im/M;

% Now check the probability of Type II error.
% Get some alternative hypotheses:

```



```
mualt = 445:458;
betahat = zeros(size(mualt));
for j = 1:length(mualt)
    Im = 0;
    % Get the true mean.
    mu = mualt(j);
    for i = 1:M
        % Generate a sample from H_1.
        xs = sigma*randn(1,n) + mu;
        Tm = (mean(xs)-454)/sigxbar;
        if Tm > cv % Then did not reject H_0.
            Im = Im +1;
        end
    end
    betahat(j) = Im/M;
end
% Get the estimated power.
powhat = 1-betahat;
plot(powhat,'b*-')
```

[↓ Example](#)

2.2.2 MCMC with matlab

以煤矿灾难数据为例, 考虑模型

$$\begin{aligned} Y_i &\sim \text{Poisson}(\theta), & i = 1, \dots, k \\ Y_i &\sim \text{Poisson}(\lambda), & i = k + 1, \dots, n \end{aligned}$$

使用Bayes方法, 取先验

$$\theta \sim G(a_1, b_1); \lambda \sim G(a_2, b_2); b_1 \sim G(c_1, d_1); b_2 \sim G(c_2, d_2)$$

故得到条件分布

$$\theta|Y, \lambda, b_1, b_2, k \sim G(a_1 + \sum_{i=1}^k Y_i, b_1 + k);$$

$$\lambda|Y, \theta, b_1, b_2, k \sim G(a_2 + \sum_{i=k+1}^n Y_i, b_2 + n - k);$$

$$b_1|Y, \theta, \lambda, b_2, k \sim G(a_1 + c_1, \theta + d_1);$$

$$b_2|Y, \theta, \lambda, b_1, k \sim G(a_2 + c_2, \lambda + d_2);$$

$$f(k|Y, \theta, \lambda, b_1, b_2) = \frac{L(Y; k, \theta, \lambda)}{\sum_{j=1}^n L(Y; j, \theta, \lambda)}$$

$$L(Y; k, \theta, \lambda) = \exp(k(\lambda - \theta))(\theta/\lambda)^{\sum_{i=1}^k Y_i}$$

使用Gibbs算法, 对 k, θ, λ 产生MC链, 代码如下:

```
% y contains number of disasters.
y=[4,5,4,1,0,4,3,4,0,6,3,3,4,0,2,6,3,3,5,4,5,3,...
1,4,4,1,5,5,3,4,2,5,2,2,3,4,2,1,3,2,2,1,1,1,1,...
3,0,0,1,0,1,1,0,0,3,1,0,3,2,2,0,1,1,1,0,1,0,1,...
0,0,0,2,1,0,0,0,1,1,0,2,3,3,1,1,2,1,1,1,1,2,4, ...
2,0,0,0,1,4,0,0,0,1,0,0,0,0,0,1,0,0,1,0,1];
% year contains the year.
year=1:112+1850;

n = length(y);
m = 1100; % number in chain
a1 = 0.5; a2 = 0.5; c1 = 0; c2 = 0; d1 = 1; d2 = 1;
theta = zeros(1,m);
lambda = zeros(1,m);
```

[↑Code](#)

```

k = zeros(1,n);
% Holds probabilities for k.
like = zeros(1,n);
% Get starting points.
k(1) = unidrnd(n,1,1);
% Note that k will indicate an index to the year
% that corresponds to a hypothesized change-point.
theta(1) = 1; lambda(1) = 1; b1 = 1; b2 = 1;
% Start the Gibbs Sampler.
for i = 2:m
    kk = k(i-1);
    % Get parameters for generating theta
    t = a1 + sum(y(1:kk));
    lam = kk + b1;
    % Generate the variate for theta.
    theta(i) = gamrnd(t,1/lam,1,1);
    % Get parameters for generating lambda.
    t = a2 + sum(y) - sum(y(1:kk));
    lam = n-kk+b2;
    % Generate the variate for lambda.
    lambda(i) = gamrnd(t,1/lam,1,1);

```

```

% Generate the parameters b1 and b2.
b1 = gamrnd(a1+c1,1/(theta(i)+d1),1,1);
b2 = gamrnd(a2+c2,1/(lambda(i)+d2),1,1);
% Now get the probabilities for k.
for j = 1:n
    like(j) = exp((lambda(i)-theta(i))*j)*(theta(i)/lambda(i))sum(y(1:j));
end
like = like/sum(like);
% Now sample the variate for k
k(i) = cssample(1:n,like,1);
end
% convergence plots
subplot(3,1,1)
plot(theta)
ylabel('Theta')
subplot(3,1,2)
plot(lambda)
ylabel('Lambda')
subplot(3,1,3)
plot(k)
ylabel('Change-Point k')

```

其中 *cssample* 函数为从一个离散分布中产生随机数, 源代码如下

```
function rs = cssample(x,p,n)
% CSSAMPLE Random sample from an arbitrary discrete/finite distribution.
%
% R = CSSAMPLE(X,P,N) This function will take an arbitrary discrete.
% finite distribution and return a random sample from it.
% The domain of the function is X. These are the values that the random
% variable can assume. The probability associated with each one is given in the
% vector P. The number of variates that will be generated is N.

% W. L. and A. R. Martinez, 9/15/01
% Computational Statistics Toolbox

if length(x) ~= length(p)
    error('The size of the input vectors do not match.')
    return
end
```

```
% sort just in case they are not in order
[xs,ind]=sort(x);
ps = p(ind); % sort these in the same order as x
% Get the cdf
F = cumsum(ps);
% Find all of the required variates
for i=1:n
    u = rand(1,1);
    if u<= F(1)
        rs(i) = x(1);
    elseif u > F(end-1)
        rs(i) = x(end);
    else
        ind = find(u <= F);
        rs(i) = xs(ind(1));
    end
end
```

[↓Code](#)

Chapter 3

Symbolic computation with matlab

matlab可以进行符号运算. 创建符号变量的函数为`sym`和`syms`.

3.1 Creating Symbolic Variables and Expressions

> `sym`

函数`sym`用于创建单个符号变量, 使用格式为

```
sym('字符串或者表达式')
```

[↑Code](#)

[↓Code](#)

例如

[↑Example](#)


```
x = sym('x')
a = sym('alpha')
rho = sym('(1 + sqrt(5))/2')
f = rho^2 - rho - 1
```

[↓Example](#)

➤ `syms`

函数`syms`用于创建多个符号变量, 使用格式为

```
syms 变量1 变量2 ...
```

[↑Code](#)

[↓Code](#)

例如

```
syms a b c x;
f = sym('a*x^2 + b*x + c');
```

[↑Example](#)

[↓Example](#)

3.2 Calculus

> Differentiation 例如

```
syms x;
f = sin(5*x);
diff(f)
g = exp(x)*cos(x);
diff(g,2) % take the second derivative of g
c = sym('5');
diff(c)
syms s t % Derivatives of Expressions with Several Variables
f = sin(s*t);
diff(f,t)
```

[↑Example](#)

[↓Example](#)

> Limits 例如

[↑Example](#)

```
syms h n x
limit((cos(x+h) - cos(x))/h,h,0)
limit( (1 + x/n)^n,n,inf )
limit(x/abs(x),x,0,'left') %One-Sided Limits
```

[↓Example](#)

➤ **Integration** 例如

```
syms a positive;
syms x;
f = exp(-a*x^2);
int(f,x,-inf,inf)
```

[↑Example](#)

[↓Example](#)

➤ **Symbolic Summation** 例如求级数和 $1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots$ 和几何级数和 $1 + x + x^2 + \dots$:

[↑Example](#)

```
syms x k
s1 = symsum(1/k^2,1,inf)
s2 = symsum(x^k,k,0,inf)
```

[↓Example](#)

➤ Solving Equations 例如

```
syms a b c x
S = a*x^2 + b*x + c;
solve(S)
b = solve(S,b) % solve for b
```

[↑Example](#)

[↓Example](#)

Chapter 4

Optimization

最小化的优化函数有 `fminbnd`, `fminsearch`, `fminunc`, `fmincon`, `fgoalattain`, `fminimax`, `lsqcurvefit`, 以及 `lsqnonlin` 等等.

4.1 Unconstrained Minimization Example

函数 `fminunc` 用以寻求无限制条件的目标函数的最小值. 例如求解

$$\min_x f(x) = e^{x_1} (4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$$

Step 1: 创建 M-文件

```
function f = objfun(x)
f = exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
```

[↑ Example](#)

[↓Example](#)

Step 2: 使用无限制条件的优化算法:

[↑Example](#)

```
x0 = [-1,1];    % Starting guess
options = optimset('LargeScale','off');
[x,fval,exitflag,output] = fminunc(@objfun,x0,options)
```

[↓Example](#)

4.2 Nonlinear Inequality Constrained Example

如果有非线性不等式限制条件, 则可以使用函数**fmincon**. 例如对前面的例子, 若限制条件为

$$x_1 x_2 - x_1 - x_2 \leq -1.5; x_1 x_2 \geq -10;$$

则根据**fmincon**的使用规则,

Step 1: 对建立一个M文件计算限制条件:

```
function [c, ceq] = confun(x)
% Nonlinear inequality constraints
c = [1.5 + x(1)*x(2) - x(1) - x(2);
     -x(1)*x(2) - 10];
% Nonlinear equality constraints
ceq = [];
```

[↑Example](#)

[↓Example](#)

Step 2: 使用优化算法:

```
x0 = [-1,1]; % Make a starting guess at the solution
options = optimset('LargeScale','off');
[x, fval] = ...
fmincon(@objfun,x0,[],[],[],[],[],[],@confun,options)
```

[↑Example](#)

[↓Example](#)