

第 11 讲 微分方程、极值与优化问题的数值求解

(第 8 章 MATLAB 方程数值求解)

目的:

- 一、掌握常微分方程数值求解的方法。
- 二、求函数的极值点和优化问题。

一、掌握常微分方程数值求解的方法。

在 matlab 中, 根据不同的方法, 设计了多个计算常微分方程的函数 (**ode 系列函数**) 分别介绍如下:

- 1、**ode45()**: 高阶(4-5)的显式单步龙格库塔法; (常用方法)
- 2、**ode23()**: 低阶(2-3)的显式单步龙格库塔法;
- 3、**ode113()**: 可变阶(1-13)的多步 PECE 法;
- 4、**ode15s()**: 可变阶(1-5)隐式多步法;
- 5、**ode15i()**: 可变阶(1-5)隐式法; 等等

上面方程, 大致的调用格式均为: **$[x, y] = \text{ode45}(\text{fun}, \text{xspan}, y_0, \text{options})$**

参数说明:

x: 数值解的自变量值;

y: 数值解的因变量值, 如果是二阶微分方程, 那么求解后矩阵 **y** 的第一列 **y(:, 1)** 表示 **y** 的值, 第二列 **y(:, 2)** 表示 **y'** 的值;

xspan: 指定的自变量的取值范围, 初值点开始的区间 $[x_0, T]$, **T** 可小于 **x0**; 也可以指定步长 **h**, 设置为 **x0:h:T**。由于整体误差的关系, 范围一般都不取大。

y0: 初值, 对应于 **x0**, 通常在使用 **ode** 前, 用 **y0=[y1, y2]** 或者 **y0=[y1; y2]** 的方式给出;

options: 其它参数。

fun: 微分方程对应的函数的句柄 (如函数是 **m** 文件, 则 **fun** 是 @ 函数名; 如函数是匿名函数, **fun** 就是函数名)

微分方程对应的函数 fun 的设计说明 (以 **m** 文件为例):

(1) 对于一阶微分方程 $y' = f(x, y)$ ，将最高阶导数 y' 写在左边，其它移到右边

建立的代表微分方程的函数如下，以 $y' = xy$ 为例：

```
function dy=myfun(x, y) %此处 dy 就是微分方程左边的 y' ,  
  
    dy=x*y;  
  
end
```

(2) 对于二阶微分方程 $y'' + p(x)y' + q(x)y = f(x)$ ，

二阶微分方程的初始条件格式是给出 $y(x_0), y'(x_0)$ 的值，微分方程数值解法是从这两个值出发进行迭代。其本质是将微分方程当做方程组进行求解。所以我们需要将二阶微分方程改写为方程组的模式。改写方法如下：

将 $y'' + p(x)y' + q(x)y = f(x)$ 改写为如下方程组：

$$\begin{cases} y' = 0y + 1y' + 0x \\ y'' = -q(x)y - p(x)y' + f(x) \end{cases}$$

如设 $dy = \begin{pmatrix} y' \\ y'' \end{pmatrix}$ ， $Y = \begin{pmatrix} y \\ y' \end{pmatrix}$ 则 $dy(1) = y'$, $dy(2) = y''$ ， $Y(1) = y$, $Y(2) = y'$

于是方程组可写为 $dy(1) = Y(2)$ ； $dy(2) = -q(x)*Y(1) - p(x)*Y(2) + f(x)$

对应的 matlab 函数是：

```
function dy=myfun(x, Y)  
  
dy(1)=Y(2);  
  
dy(2)=-q(x)*Y(1)-p(x)*Y(2)+f(x);  
  
% 上面也可以直接写成 dy=[Y(2); -q*Y(1)-p*Y(2)+f(x)]  
  
end
```

也可以用矩阵来表示 $\begin{pmatrix} y' \\ y'' \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -q & -p \end{pmatrix} \begin{pmatrix} y \\ y' \end{pmatrix} + \begin{pmatrix} 0 \\ f(x) \end{pmatrix}$

对应的 matlab 函数是：

```
function dy=myfun(x,Y)
```

```
dy=zeros(2,1); %给 dy 预分配一个矩阵，它是一个列向量。
```

```
A=[0,1;-q(x),-p(x)];
```

```
z=[Y(1);Y(2)]; %这里的分量 Y(1),Y(2)就是方程组右边中的(y;y'), z 是列向量
```

```
g=[0; f(x)] %注意 g 是列向量。
```

```
dy=A*z+g; %这是原方程所对应的方程组
```

```
end
```

注：（1）在 $dy=myfun(x,Y)$ 中，输入形参中的 Y 不要单纯理解为方程中的单变量 y ，而是要理解为除自变量 x 外的所有输入变量构成的输入向量，例如对 $y'' = -py' - qy + f(x)$ 而言， $dy=myfun(x,Y)$ 中的 Y 要理解为向量 (y, y') ， $Y(1)$ 就是原方程中的 y ， $Y(2)$ 就是原方程中的 y' 。如果实在怕混淆，也可以用其它符号表示形参 Y ，例如写成 $dy=myfun(x,s)$ ，此时 $s(1)$ 就是 y ， $s(2)$ 就是 y' 。

（2） $dy=myfun(x,Y)$ 中，形参自变量 x 不能省略，就算原方程中没有出现 x ，也不能省略。例如，如果方程是 $y'' = 2y' + y$ ，设计对应函数是，也要写成 $dy=myfun(x,Y)$ 。

熟练后也可以不用 m 文件定义，直接用匿名函数定义：

```
dy=@(x,y)[Y(2);-q*Y(1)-p*Y(2)+f(x)]
```

 注意中括号内的分割符用的是**分号(;)。**

设计好微分方程对应的函数后，就可以调用系统 ode 函数来求解该函数了。

```
[x,y]=ode(fun,xspan,y0);
```

其中 $[x,y]$ 是 ode 函数的输出； $xspan$ 是指定的自变量范围，由于整体误差的问题，为保证精度，范围通常取得比较小； $y0$ 是初始的 y, y' 的值。

注意：1、 ode 函数的输出 $[x,y]$ 和微分方程的变量对应关系是：

输出 x ：等于原微分方程的 x ，也就是 $xspan$ 指定的 x 的范围；

输出 y ：

（1）如果是一阶微分方程， $[x,y]$ 中的 y 就是微分方程中的 y ，也就是相对应于 $xspan$ 上的点的 y 值；

(2) 如果是二阶微分方程, $[x, y]$ 中的 y 会是一个两列矩阵, 其中第一列 $y(:, 1)$ 是 $xspan$ 对应的 y 值, 第二列 $y(:, 2)$ 是 $xspan$ 对应的 y' 值。

(3) 在使用 `ode` 函数求解时, 如果不设置输出 $[x, y]$ 选项, 直接用 `>> ode(@myfun, tspan, Y0)`, 那么 `matlab` 会弹出对应数值解的拟合曲线图, 如果设置了输出 $[x, y]$ 那么 `matlab` 不会弹出对应的图, 而是在工作区给出 x 和 y 的值。

注: 算法原因, 在写微分方程函数时不用点运算, 当然用点运算也可以。

练习 1: 求解一阶微分方程 $y' = 2x + y$, $y(0) = 1$

分析: 方程的输出是 y' , 输入是 x, y 。

```
%一阶微分方程, 懒得用 m 文件定义微分方程, 用匿名更简单
>>mydy=@(x,y)(2*x+y); % mydy 就是输出 y', @(x,y)就是输入 x, y
>>xspan=[0,5]; %设置 x 的取值范围, 左端点就是初始条件中的 0
>>y0=1; %在左端点 x=0 处的值
>>[x,y]=ode45(mydy, xspan, y0);
>> plot(x,y); %作图观察数据拟合曲线
```

练习 2: 求解一阶微分方程并绘制曲线 $y' = \cos x + x - y$, $y(0) = 0$

练习 3: 求二阶微分方程的数值解, 并绘制解的曲线

$$\begin{cases} \frac{xd^2y}{dx^2} - 5\frac{dy}{dx} + y = 0 \\ y(1) = 1, y'(1) = 1 \end{cases}$$

分析: 改写原方程为:
$$\begin{cases} y' = 0y + 1y' \\ y'' = -\frac{1}{x}y + \frac{5}{x}y' \\ y(1) = 1, y'(1) = 1 \end{cases}$$

因为初始条件中 $x = 1$, 所以迭代需要从 $x = 1$ 开始的, 因此我们设置的自变量区间也要从 1 开始设置, 比如 `xspan=[1, 5]`。

```
%在 m 文件中用矩阵乘法定义微分方程函数
```

```
function dy=f4(x,y)
```

%dy 就是输出, dy(1)表示 y' , dy(2)表示 y'' ; 对(x,y)中的 y, 其中 y(1)表示 y, y(2)是 y'

```
dy(1)=y(2);
```

```
dy(2)=-y(1)/x+5*y(2)/x;
```

```
dy=[dy(1);dy(2)]; % 系统求解函数需要因变量是列, 所以需将 dy 写为列。
```

```
end
```

%或者如下写对应函数

```
function dy=f4(x,y) %dy(1)和 dy(2)就是  $y'$  和  $y''$ ; y(1), y(2)就是 y 和  $y'$ 。
```

```
dy=[y(2); -y(1)/x+5*y(2)/x]; % 注意中括号内用分号隔开。
```

```
end %输入分成两块, 一块是自变量x, 一块是其它输入y。
```

%在命令行或着脚本中运行如下语句求解上面定义的微分方程

```
>>xspan=[1,5]; y0=[1,1]; %y0 的两个值就是在 x=1 处的初始条件函数值。
```

```
>>[x,y]=ode45(@f4, xspan, y0); %需在 m 文件定义的函数前加@
```

```
>>plot(x, y(:, 1), x, y(:, 2)); %做函数图和导函数图, y(:, 1)是函数值, y(:, 2)是导数值;
```

```
>>legend('y', "y' ");
```

```
>>xlabel('x'); ylabel('y');
```

%也可以在命令行或脚本中用匿名函数方式定义微分方程并求解

```
dy=@(x,y)[ y(2); -y(1)/x+5*y(2)/x]; %中间用分号隔开
```

```
>>xspan=[1,5]; y0=[1,1];
```

```
>>[x,y]=ode45(dy, xspan, y0); %注意这里函数名前不用加@
```

```
>>plot(x, y(:, 1), x, y(:, 2));
```

```
>>legend('y', "y' ");
```

```
>>xlabel('x'); ylabel('y');
```

练习 4: 求解二阶微分方程并绘图 $y'' = y'(1 - y^2) - y$, $y(0) = 0, y'(0) = 1$

练习 5: 求曲线参数方程的微分方程组数值解, 并绘制解的曲线

$$\begin{cases} y'_1 = y_2 y_3 \\ y'_2 = -y_1 y_3 \\ y'_3 = -0.51 y_1 y_2 \\ y_1(0) = 0, y_2(0) = 1, y_3(0) = 1 \end{cases}$$

分析：这里因变量是向量 (y'_1, y'_2, y'_3) ，自变量是 (y_1, y_2, y_3) 与 x 。

```
function dy=fun(x,y)
%用 dy(1),dy(2),dy(3)表示因变向量, y(1),y(2),y(3)表示自变量, 自变量x需单独列出。
dy=[y(2)*y(3); -y(1)*y(3); -0.51*y(1)*y(2)];
end

>>xspan=[0, 8]; y0=[0, 1, 1];
>>[x,y]=ode45(@fun,xspan, y0);
>>plot3(y(:, 1), y(:, 2), y(:, 3)); %做函数图;
>>grid on; %开网格线, 可以观察的更好。
```

二、求函数的最小值和优化问题

特别提醒：matlab 只求函数 f 的最小值；想求最大值，可以求 $-f$ 的最小值。

(一) 无约束优化，函数的最小值问题

1、一元函数在指定区间的最小值（这个算有约束，因为指定了区间）

Matlab 中可以使用 `fminbnd` 函数求一元函数在区间内的最小值，其调用格式为：

`[x, fval]=fminbnd(fun, a, b, options)`

其中：`[x, fval]`是极值点和极值，`fun`是一元函数的函数句柄；`a, b`是求极值区间(`a, b`)；`options`是参数选项，可以通过`op=optimset('属性名', '属性值')`设置迭代容差或者设置是否显示迭代步骤等。`fminbnd`的算法基于黄金分割搜索和抛物线插值方法，它不计算端点处的函数值，也就是它只在开区间内求解，如果端点是极值点，那么`fminbnd`返回的是接近端点的值。

注：算法原因，在写函数方程时记得用点运算。

练习 6：求 $f(x) = \frac{x^3 + \cos x + x \ln x}{e^x}$ 在(0,1)内的最小值。

```
>>f=@(x) (x.^3+cos(x)+x.*log(x))/exp(x);
```

```
>>op=optimset('disp','off'); %不显示迭代步骤, 如想显示步骤使用op=optimset('disp','iter')
>>[x,fmin]=fminbnd(f, 0, 1, op)
x = 0.5223
fmin =0.3974
```

2、多元函数在初值点附近的最小值

Matlab 中使用 **fminsearch** 和 **fminunc** 求多元函数在初始点附近的极值, 分别用于处理多元函数偏导不存在和存在这两种情况下的极值, 如果已知函数表达式, 通常使用 fminunc 函数。

[x, fval]=fminsearch(fun, x0, options)

用单纯型法求解多元函数极值, 当函数不可导时使用, x0 是初始迭代点。

[x, fval]=fminunc(fun, x0, options)

牛顿迭代法求多元函数极值, 当函数可导时使用, x0 是初始迭代点。

虽然函数可导时也可以使用 fminsearch 求解, 但此时不如 fminunc 精确, 所以可导时推荐使用 fminunc。

注: 算法原因, 在写函数方程时记得用点运算。

练习 7: 求 $f(x_1, x_2) = 2x_1^3 + 4x_1x_2^3 - 10x_1x_2 + x_2^2$ 在 [0,0] 附近的最小值点和最小值。

```
>>f=@(x)(2*x(1).^3+4*x(1).*x(2).^3-10*x(1).*x(2)+x(2).^2); %记得用点运算
>>op=optimset('disp','off'); %不显示迭代步骤
>>x0=[0,0];
>> [U,fmin]=fminsearch(f, x0, op) %基于单纯形算法
U = 1.0016 0.8335
fmin = -3.3241
>> [U,fmin]=fminunc(f, x0, op) %因为函数可导, 所以结果比用fminsearch准确
U = 0 0
fmin = 0
```

练习 8: 求函数 $f(x, y) = xe^{-x^2-y^2}$ 在原点附近的最小值。

%在脚本中运行

```
[x,y]=meshgrid(-2:0.1:2); z=x.*exp(-x.^2-y.^2);
```

surf(x,y,z); %原点附近作图，大概观察一下最值位置。

%下面用函数求解最值

```
f=@(x)(x(1).*exp(-x(1).^2-x(2).^2));
```

```
op=optimset('disp','off'); %不显示迭代步骤
```

```
[x1,y1]=fminsearch(f,[0,0], op)
```

```
[x2,y2]=fminunc(f,[0,0], op)
```

练习 9: 求函数 $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ 在 $(-1, 2)$ 附近的最小值。

(二) 有约束优化——约束条件下函数的最小值问题

多变量约束优化问题的标准形式是：

目标：求多元函数 $f(x) = f(x_1, x_2, x_3, \dots, x_n)$ 的最小值

常见对自变量的约束条件类型有：

(1) **线性不等式约束** $A \cdot x \leq b$ ，例如 $\begin{cases} x_1 + 2x_2 \leq 3 \\ 2x_1 - x_2 \leq 2 \end{cases}$ ，等同于 $\begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 3 \\ 2 \end{pmatrix}$

(2) **线性等式约束** $Aeq \cdot x = beq$ ，例如 $\begin{cases} x_1 + x_2 = 0 \\ -x_1 + x_2 = 1 \end{cases}$ ，等同于 $\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

(3) **非线性不等式约束** $C(x) \leq 0$ 和 **非线性等式约束** $Ceq(x) = 0$ ，

例如 $\begin{cases} x_1x_2 + 2x_2x_3 + 3x_1x_2 \leq 18 \\ 2x_1^2 + 3x_2^2 + 5x_3^2 \geq 20 \end{cases}$ 与 $x_1 + x_2 + 3x_3 = 10$

此时需要设计个函数专门用于表示上面两个约束条件，通常如下：

```
function [c, ceq]=NoneLineCon(x)
```

%参数 c 和 ceq 分别为非线性不等式和非线性等式约束，参数名和函数名可以自取

```
c(1)=x(1)*x(2)+2*x(2)*x(3)+3*x(1)*x(2)-18; %可以不用点运算。
```


$c(2) = -2 \times x(1)^2 - 3 \times x(2)^2 - 5 \times x(3)^2 + 20$; %由于求解函数只认 \leq 的条件, 所以记得移项

$ceq = x(1) + x(2) + 3 \times x(3) - 10$;

end

(4) 上下界约束 $lb \leq x \leq ub$, 例如 $x_1, x_2, x_3 \geq 0$ 此时 $lb = [0; 0; 0]$, $ub = []$ 用分号。

Matlab 中用于求解约束条件下函数最小值的函数是 `fmincon` 函数, 其调用格式为:

`[x,fval]=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)`

其中: `fun` 是目标函数、`x0` 是初始迭代点、`A`, `b` 是线性不等式约束、`Aeq`, `beq` 是线性等式约束、`lb`, `ub` 是上下界约束、`nonlcon` 是非线性不等式和等式约束函数(没有时可以缺省)、`options` 是其它设置(可缺省)。

记忆顺序: 目标与初始排第一, 线性条件排第二, 第三需要上下界, 复杂条件排最后。

练习 10、设初始迭代点 $x_0 = [1; 1]$, 求以下问题的最小值

目标: $\min f = 0.4x_2 + x_1^2 + x_2^2 - x_1x_2 + \frac{1}{30}x_1^3$

线性不等式约束条件:
$$\begin{cases} x_1 + 0.5x_2 \geq 0.4 \\ 0.5x_1 + x_2 \geq 0.5 \end{cases}$$

线性等式约束条件: 无

非线性不等式约束条件: 无

非线性等式约束条件: 无

自变量的上下限约束: $x_1 \geq 0, x_2 \geq 0$

分析: 由描述可知 $x_0 = [1; 1]$ $A = \begin{bmatrix} -1 & -0.5 \\ -0.5 & -1 \end{bmatrix}$, $b = \begin{pmatrix} -0.4 \\ -0.5 \end{pmatrix}$, $Aeq = []$, $beq = []$, $lb = [0; 0]$, $ub = []$,

`Nonlcon = []`。

注: 可以不用点运算。

%在脚本中运行

`f=@(x)(0.4*x(2)+x(1)^2+x(2)^2-x(1)*x(2)+x(1)^3/30)`; %目标函数

`x0=[1; 1]`; %这里如用分号表示成列, 则结果就是列。用逗号表示成行, 结果就是行。

```
A=[-1,-0.5;-0.5,-1];
b=[-0.4;-0.5];
Aeq=[];beq=[];
Lb=[0; 0]; Ub=[]; %可以用逗号，但一般统一用分号
[x,y]=fmincon(f,x0,A,b,Aeq,beq,Lb,Ub); %f 本身是句柄，前面不用加@
```

补充知识，梯度法求极值（以二元函数为例）：

设 $z = f(x, y)$ 可微，则

$$\Delta z|_{(x_0, y_0)} = f(x_1, y_1) - f(x_0, y_0) \approx z'_x \Delta x + z'_y \Delta y = (z'_x, z'_y) \cdot (\Delta x, \Delta y)$$

其中 (z'_x, z'_y) 就是函数在 (x_0, y_0) 处的梯度向量， $(\Delta x, \Delta y)$ 为 $(x_0, y_0), (x_1, y_1)$ 构成的向量。上式表明，从 (x_0, y_0) 出发沿着某方向 $(\Delta x, \Delta y)$ 前进，产生的 Δz 是梯度向量与方向向量的内积。

推论：梯度向量的方向指向 z 增大的方向。

证：将行进方向 $(\Delta x, \Delta y)$ 设置来与梯度向量一致，即取 $(\Delta x, \Delta y) = k(z'_x, z'_y), (k > 0)$

此时 $\Delta z = k(z'_x, z'_y) \cdot (z'_x, z'_y) = k[(z'_x)^2 + (z'_y)^2] \geq 0$ ，即梯度的方向是 z 增大的方向。

推论可以用来求函数的极值，在极值处 $z'_x = z'_y = 0$ ，此时 $\Delta z = 0$ 。所以我们可以从某点 (x_0, y_0) 出发，先求出该点的梯度向量 \vec{T} ，然后取 $(\Delta x, \Delta y) = k(z'_x, z'_y)$ （如果想求函数的极大值就取 $k > 0$ ，想求极小值就取 $k < 0$ ）得到新点 $(x_1, y_1) = (x_0 + \Delta x, y_0 + \Delta y)$ ，求出 $\Delta z_1 = f(x_1, y_1) - f(x_0, y_0)$ ，如果 $|\Delta z_1| < tol$ ，说明 $\Delta z_1 \approx 0$ ，从而 (x_1, y_1) 就是近似极值点。如果 $|\Delta z_1| > tol$ ，则将 (x_1, y_1) 作为新的起始点，重复以上步骤直到 $|\Delta z| \approx 0$ 。

练习 11、初始迭代点： $x_0 = [1, 2, 2]$ 求下列优化问题的最优解（目标函数的最小值）

目标： $\min f(x) = -x_1 x_2 x_3$

线性不等式约束条件： 无

线性等式约束条件：无

非线性不等式约束条件：
$$\begin{cases} x_1x_2 + 2x_2x_3 + 3x_1x_3 \leq 18 \\ 2x_1^2 + 3x_2^2 + 5x_3^2 \geq 20 \end{cases}$$

非线性等式约束条件：无

自变量的上下限约束： $x_1, x_2, x_3 \geq 0$

%非线性约束条件函数的结构导致了只能用m文件单独设置

```
function [c, ceq]=cx(x)
c(1)=x(1)*x(2)+2*x(2)*x(3)+3*x(1)*x(3)-18;
c(2)=-2*x(1)^2-3*x(2)^2-5*x(3)^2+20; %注意需用<=表示式
ceq=[];
end
```

%在脚本中运行

```
f=@(x)-x(1)*x(2)*x(3); %目标函数
x0=[1; 2; 2];
A=[]; b=[]; Aeq=[]; beq=[]; lb=[0; 0; 0]; ub=[];
[x, fval]=fmincon(f, x0, A, b, Aeq, beq, lb, ub, @cx)
%目标函数 f 本身是句柄，前面不用加@，但非线性约束函数 cx 不是句柄，前面需要加@
```

练习 12： 设初始迭代点 $x_0 = [1/4; 1/4]$ ，求解以下优化问题

目标： $\min f = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

线性不等式约束条件： 无

线性等式约束条件： 无

非线性不等式约束条件：
$$\left(x_1 - \frac{1}{3}\right)^2 + \left(x_2 - \frac{1}{3}\right)^2 \leq \left(\frac{1}{3}\right)^2$$

非线性等式约束条件： 无

自变量的上下限约束： $0 \leq x_1 \leq 0.5, 0.2 \leq x_2 \leq 0.8$ 。

除 fmincon 这个比较通用的函数外，Matlab 中还有用于求解线性约束条件下线性目标函数最值问题的专用函数 **linprog**，其调用格式是：

[x, fval]=linprog(f, A, b, Aeq, beq, lb, ub, x0, options)

参数含义和 fmincon 相同，不同之处在于：

- (1) 由于目标函数是线性函数，这里的参数 f 是线性函数对应的系数向量
- (2) 迭代点 x0 放在了最后，并且可以缺省。
- (3) 由于约束条件全是线性条件，所以没有非线性约束条件函数。

练习 13：求解线性约束条件下线性目标函数的极值问题

4、目标： $\min f = -5x_1 - 4x_2 - 6x_3$

$$\text{线性不等式约束条件: } \begin{cases} x_1 - x_2 + x_3 \leq 20 \\ 3x_1 + 2x_2 + 4x_3 \leq 42 \\ 3x_1 + 2x_2 \leq 30 \end{cases}$$

线性等式约束条件：无

自变量的上下限约束： $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$ ，即有下限，无上限。

%在脚本中运行

f=[-5,-4,-6]; % 线性目标函数

A=[1, -1, 1; 3, 2, 4; 3, 2, 0]; b=[20; 42; 30];

Aeq=[]; beq=[];

lb=[0; 0; 0]; ub=[]; %统一用分号

[x,y]=linprog(f,A,b,Aeq,beq,lb,ub)

练习 14：某饲养场饲养动物出售，设每头动物每天至少需要 700g 蛋白质、30g 矿物质和 100mg 维生素，现有五种饲料可供选用，各种饲料每千克的营养成分及单价如下表所示，请给出既能满足动物生长营养需要又能使费用最省的五种饲料的搭配方案。

饲料	蛋白质/g	矿物质/g	维生素/mg	价格（元/kg）
1	3	1	0.5	0.2
2	2	0.5	1	0.7

3	1	0.2	0.2	0.4
4	6	2	2	0.3
5	18	0.5	0.8	0.8