代数方程的求解

一元方程的图解法

思路： 用 ezplot() 函数绘隐函数 $f(x)=0$ 曲线找零点。

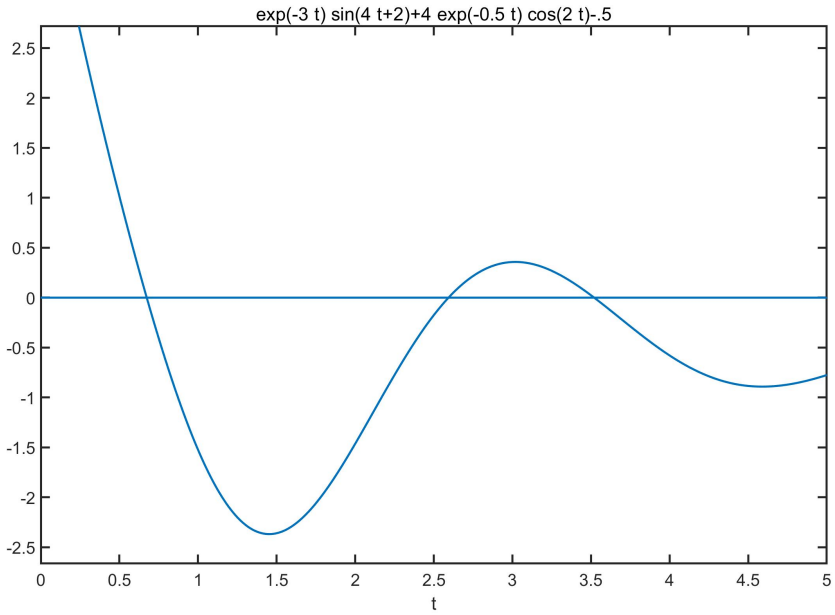
例 (6-1)

解 $\mathrm{e}^{-3t}\sin(4t+2)+4\mathrm{e}^{-0.5t}\cos(2t)=0.5$。

MATLAB 代码：

```
figure
ezplot('exp(-3*t)*sin(4*t+2)
       +4*exp(-0.5*t)*cos(2*t)
       -.5',[0 5])
line([0,5],[0,0])
```

输出：



三个解，约为 0.670、2.59、3.51.

二元方程的图解法

思路：绘制在同一张图上看交点。

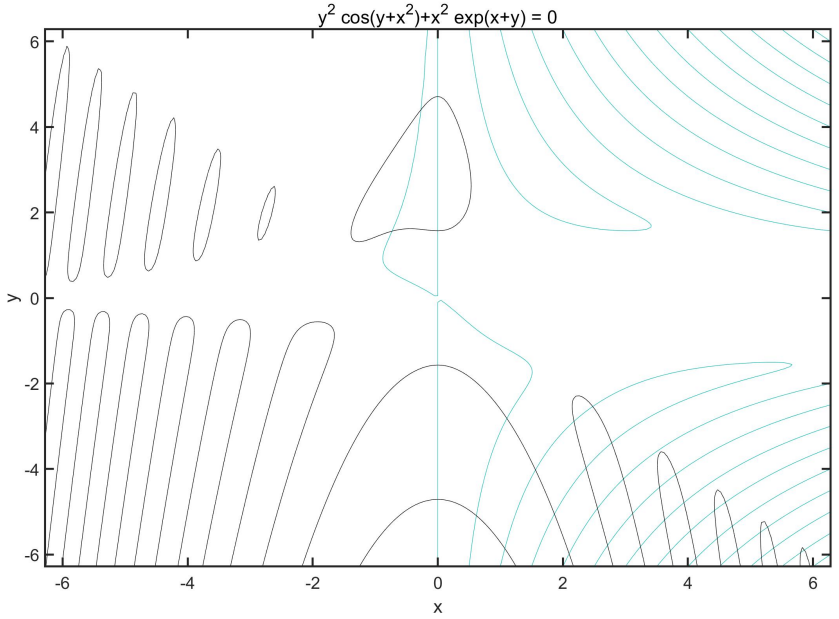
例 (6-2)

解
$$\begin{cases} x^2 e^{-xy^2/2} + e^{-x/2}\sin(xy) = 0 \\ x^2\cos(x+y^2) + y^2 e^{x+y} = 0 \end{cases}$$

MATLAB 代码：

```
figure
ezplot('x^2*exp(-x*y^2/2)+exp
       (-x/2)*sin(x*y)')
hold on
h1 = ezplot('y^2*cos(y+x^2)+x
       ^2*exp(x+y)');
set(h1,'Color','k')
hold off
```

输出：



局限：仅适用一元、二元方程。仅能近似得实数根。

多项式型方程的准解析解法

用 solve 函数，语法：

```
S = solve(eqn1, eqn2,... ,eqnn)%最简调用方式
[x,...] = solve(eqn1, eqn2,... ,eqnn)%直接得出根
[x,...] = solve(eqn1, eqn2,... ,eqnn, 'x,...')%同上, 并指定变量
```

例 (6-4)

解 $\begin{cases} x^2 + y^2 - 1 = 0 \\ 0.75x^3 - y + 0.9 = 0 \end{cases}$

MATLAB 代码:

```
syms x y
[x,y]=solve(x^2+y^2-1==0,0.75*x^3-y+0.9==0)
x=double(x), y=double(y)
[eval('x.^2+y.^2-1') eval('0.75*x.^3-y+0.9')]%解的验算
```

输出：

```
x = -0.9817 + 0.0000i
     0.3570 + 0.0000i
    -0.5540 - 0.3547i
    -0.5540 + 0.3547i
     0.8663 - 1.2154i
     0.8663 + 1.2154i
y = 0.1904 + 0.0000i
     0.9341 + 0.0000i
     0.9293 - 0.2114i
     0.9293 + 0.2114i
    -1.4916 - 0.7059i
    -1.4916 + 0.7059i
```

一般非线性方程数值解

用 fsolve() 函数，语法：

```
x = fsolve(fun,x0)%最简单求解语句
[x,f,flag,out] = fsolve(fun,x0,opt,p1,p2,...)%一般求解格式
opt = optimset%获得默认的常用变量
opt.TolX=1e-10; set(opt,'TolX',1e-10)%修改参数
```

例 (6-8)

解 Lambert 函数，其解 w 满足 $we^w = x$。

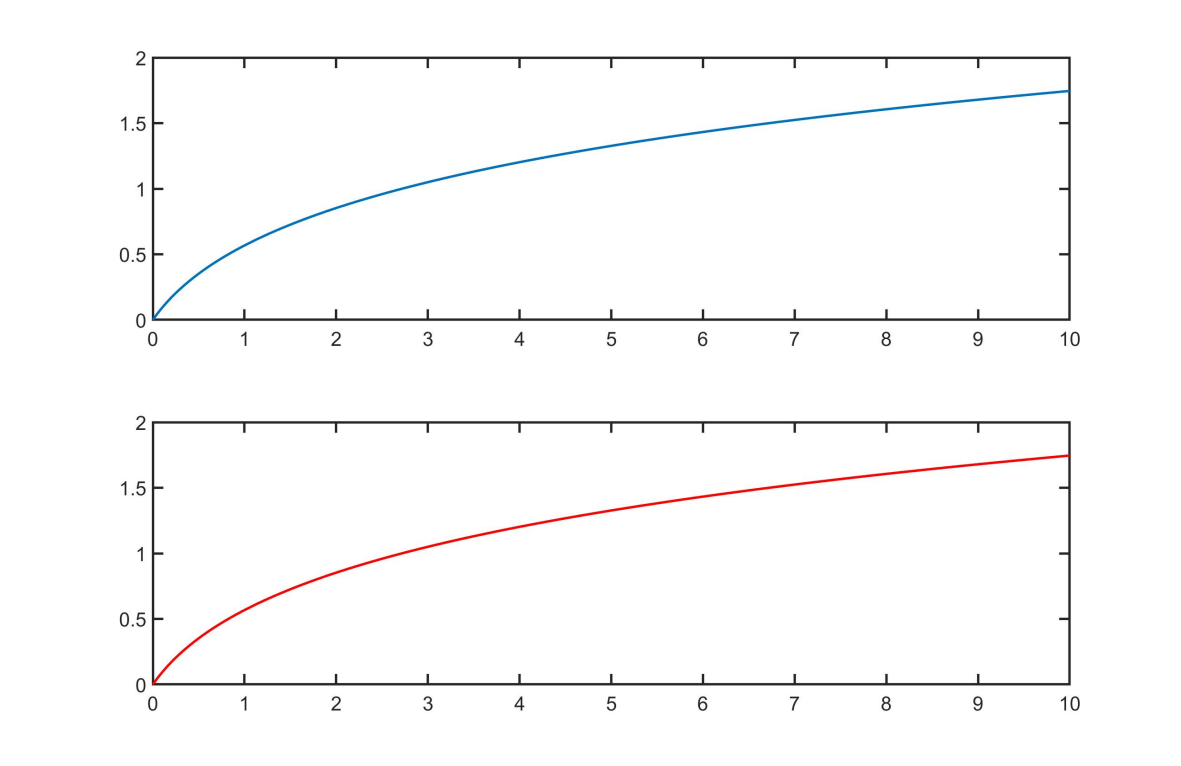
MATLAB 代码:

```
figure
y=[]; xx=0:.05:10; x0=0; h=optimset; h.
      Display='off';
for x = xx
y1 = fsolve(@(w) w.*exp(w)-x,x0,h); x0
      =y1; y=[y,y1];
end
```

```
subplot(2,1,1);
plot(xx,y);
y0 = lambertw(xx);
subplot(2,1,2);
plot(xx,y0,'r');
```

一般非线性方程数值解

输出：



蓝线：fsolve() 的解，红线：软件自带该问题求值函数。

无约束最优化问题求解

解析解法与图解法

无约束最优化问题提法：$\min_{x} f(\mathbf{x})$，优化变量 $\mathbf{x} = [x_1, \ldots, x_n]^T$，目标函数 $f(\cdot)$。

解析解法：最优点出现在一阶导数为零的点上，可列

$$\left.\frac{\partial f}{\partial x_1}\right|_{\mathbf{x}=\mathbf{x}^*} = 0, \left.\frac{\partial f}{\partial x_2}\right|_{\mathbf{x}=\mathbf{x}^*} 0, \cdots, \left.\frac{\partial f}{\partial x_n}\right|_{\mathbf{x}=\mathbf{x}^*} = 0$$

$$\Longrightarrow \begin{cases} \text{高数：Hessian 矩阵。} \\ \textbf{图解法}：\text{画曲线判断极值点。} \checkmark \end{cases}$$

局限：一元、二元函数适用，三元或多元函数无法画图表示。

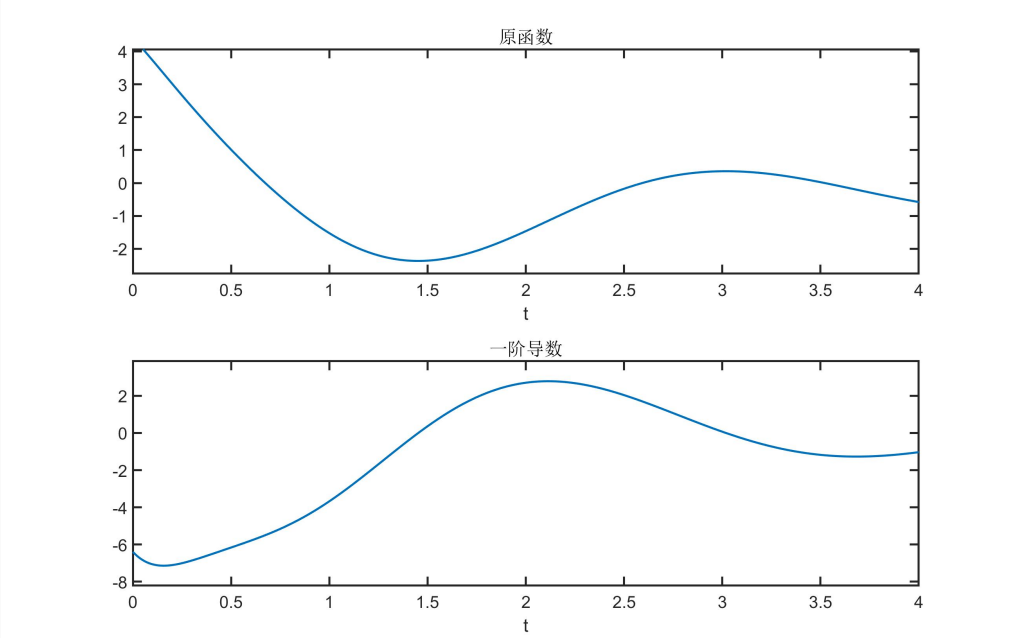
解析解法与图解法

例 (6-11)
讨论例 6-1 函数最优性。

MATLAB 代码：

```
syms t
y = exp(-3*t)*sin(4*t+2)+4*
      exp(-0.5*t)*cos(2*t)-.5;
y1 = diff(y,t)
t0 = solve(y1)
b = subs(diff(y1),t,t0)
```

输出：



可见在 [0,4] 内约 1.46 处，最小值约-2.37。

基于 MATLAB 的数值解法

数学原理：单纯形 (Simplex) 法。(Def: N dim., N+1 vertices interconnect)

思路：调用 fminsearch() 或者 fminunc() 函数。

调用语法，fminunc() 为例：

```
x=fminunc(Fun, x0) %最简求解语句
[x,f,flag,out]=fminunc(Fun, x0, opt, p1, p2,...) %一般求解格式
```

例 (6-12)

求 $z = f(x, y) = (x^2 - 2x)\,\mathrm{e}^{-x^2-y^2-xy}$ 最小值。

MATLAB 代码：

```
ff2=@(x)(x(1)^2-2*x(1))*exp(-x(1)^2-x(2)^2-x(1)*x(2));
ff2_0 = optimset;
ff2_0.Display='iter';
result1 = fminsearch(ff2,[0; 0],ff2_0)
result2 = fminunc(ff2,[0; 0],ff2_0)
```

fminsearch 输出

```
Iteration Func-count min f(x) Procedure
0 1 0
1 3 -0.000499937 initial simplex
2 4 -0.000499937 reflect
3 6 -0.00149944 expand
4 7 -0.00149944 reflect
......
71 135 -0.641424 contract inside
72 137 -0.641424 contract outside
......
优化已终止:当前的 x 满足使用 1.000000e-04 的 OPTIONS.TolX 的终止条件，F(X)
        满足使用 1.000000e-04 的 OPTIONS.TolFun 的收敛条件
......
result1 =
0.6111
-0.3056
```

fminunc 输出

```
Iteration Func-count f(x) Step-size First-order optimality
0 3 0 2
1 6 -0.367879 0.5 0.736
......
6 24 -0.641424 1 0.000619
7 27 -0.641424 1 1.8e-06
......
Optimization completed because the size of the gradient is less than the value of
      the optimality tolerance.
......
result2 =
0.6110
-0.3055
```

全局最优解与局部最优解

概念：导数为 0 的点包括局部和全局最优点。得到的最优点依赖于初始选取。⇐ 可以用遗传算法试不同初值

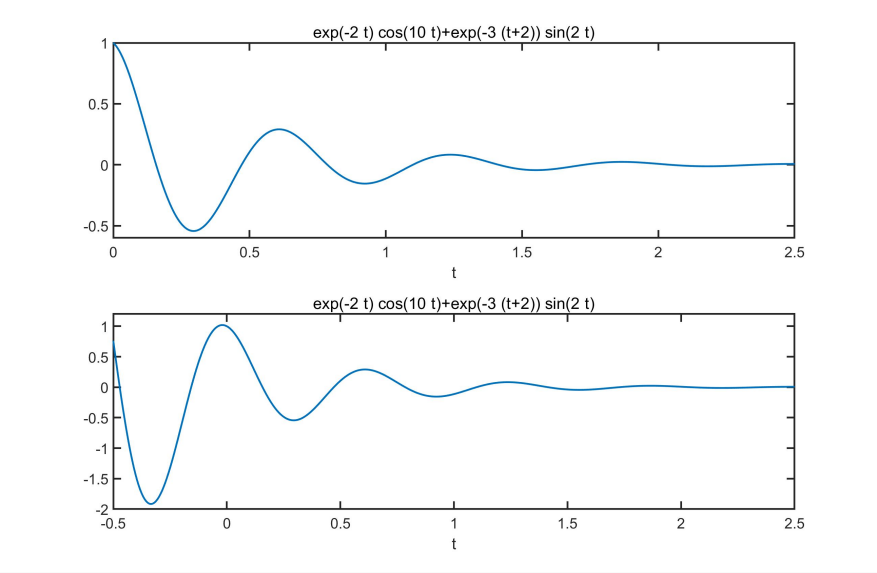
例 (6-13)

$y(t) = \mathrm{e}^{-2t}\cos 10t + \mathrm{e}^{-3t-6}\sin 2t, t \geqslant 0$ 局部最小值与全局最小值。

MATLAB 代码：

```
ff3=@(t)exp(-2*t).*cos(10*t)+
      exp(-3*(t+2)).*sin(2*t);
[t3_1,f3_1] = fminsearch(ff3,1)
[t3_2,f3_2] = fminsearch(ff3,.1)
```

输出：



[t3_1, f3_1] =[0.9228, -0.1547] 初值 t=1

[t3_2, f3_2] =[0.2945, -0.5436] 初值

利用梯度求解最优化问题

思路：引入目标函数梯度，加快计算速度，改进搜索精度，但影响计算速度。即设置 optimset.GradObj='on' 。

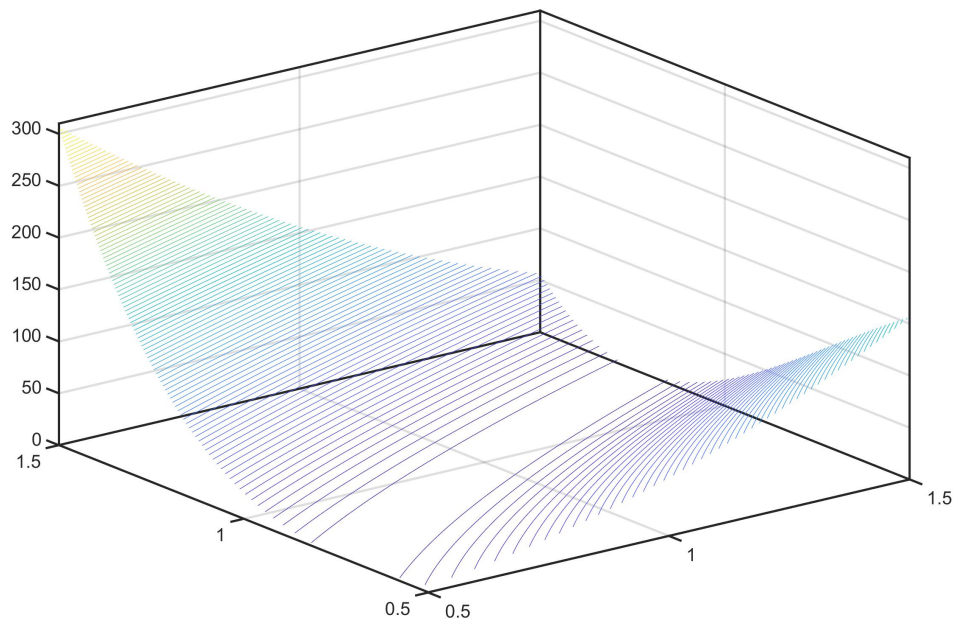
例 (6-14)

求解 Rosenbrock 函数 $f(x_1, x_2) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2$ 无约束最优化问题。

MATLAB 代码：

```
ff4=@(x)100*(x(2)-x(1)^2)^2+(1-x(1))^2;
ff4_0 = optimset; ff4_0.Display='iter'; ff4_0.TolX = 1e-10; ff4_0.TolFun= 1e
    -20;
x4 = fminunc(ff4,[0; 0],ff4_0)
syms x1 x2
ff4_1=100*(x2-x1^2)^2+(1-x1)^2;
J = jacobian(ff4_1,[x1, x2]);
```

利用梯度求解最优化问题



Rosenbrock 函数，易知最小值 (1,1)。

MATLAB 代码（续）:

```
ff4_0.GradObj='on';
x4_modi = fminunc(@ff4_modi
    ,[0; 0],ff4_0)
function [y,Gy] = ff4_modi(x)
y = 100*(x(2)-x(1)^2)^2+(1-x
    (1))^2;
Gy = [2*x(1) - 400*x(1)*(- x
    (1)^2 + x(2)) - 2; - 200*
    x(1)^2 + 200*x(2)];
end
```

利用梯度求解最优化问题

```
Iteration Func-count f(x) Step-size First-order optimality
0 3 1 2
1 12 0.771192 0.0817341 5.34
......
20 81 1.94742e-11 1 1.06e-06
x4 = (1.0000,1.0000)

Iteration Func-count f(x) Step-size First-order optimality
0 1 1 2
1 4 0.771191 0.0817342 5.34
......
22 29 1.16799e-23 1 1.26e-10
23 30 2.04561e-28 1 2.41e-13
x4_modi = (1.0000,1.0000)
```

注：实际情况常见带变量边界约束的最优化问题，如 $\min_{\mathbf{x}\ s.t.\ \mathbf{x_m}\leqslant\mathbf{x}\leqslant\mathbf{x_M}}$，可以调用 fminsearchbnd() 函数。

有约束最优化问题的计算机求解

约束条件与可行解区域

有约束最优化问题提法： $\min\limits_{\mathbf{x}\ s.t.\ \mathrm{G}(\mathbf{x})\leqslant\mathbf{0}} f(\mathbf{x})$。满足 $\mathrm{G}(\mathbf{x})\leqslant\mathbf{0}$ 的范围称为可行解区域 (feasible region)。

例 (6-15)

二元最优化问题

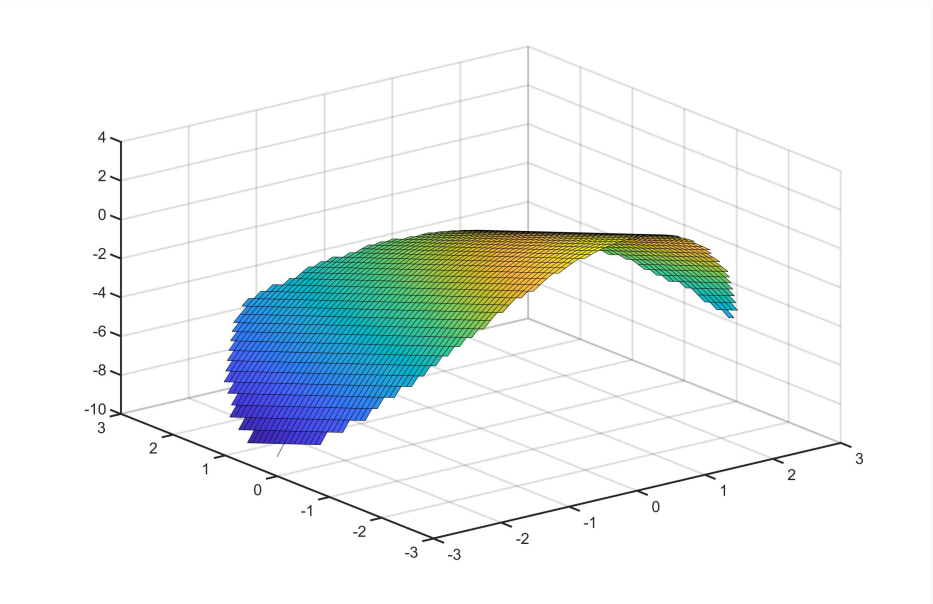
$$\max \quad (-x_1^2 - x_2)$$

$$\begin{cases} 9 \geqslant x_1^2 + x_2^2 \\ x_1 + x_2 \leqslant 1 \end{cases}$$

MATLAB 代码：

```
syms x1 x2
[x1,x2]=meshgrid(−3:.1:3);
z = −x1.^2 − x2;
i = find(x1.^2 + x2.^2 > 9); z(i) = NaN;
i = find(x1 + x2 > 1); z(i) = NaN;
surf(x1,x2,z)
```

输出：



max(z(:)) = 3, 在 (-3,0) 处。

线性规划问题的计算机求解

线性规划问题提法：

$$\min \quad \mathbf{f}^T\mathbf{x}$$
$$\mathbf{x} \text{ s.t. } \begin{cases} \mathbf{A}\mathbf{x} \leqslant \mathbf{B} \\ \mathbf{A}_{\text{eq}}\mathbf{x} = \mathbf{B}_{\text{eq}} \\ \mathbf{x}_m \leqslant \mathbf{x} \leqslant \mathbf{x}_M \end{cases},$$

线性不等式约束：$\mathbf{A}\mathbf{x} \leqslant \mathbf{B}$，线性等式约束：$\mathbf{A}_{\text{eq}}\mathbf{x} = \mathbf{B}_{\text{eq}}$，上界变量 $\mathbf{x}_M$，下界变量 $\mathbf{x}_m$。

linprog() 函数调用：

```
[x,f_opt,flag,c] = linprog(f, A, B, A_eq, B_eq, x_m, x_M, x_0, OPT, p1, p2,...)
```

线性规划问题的计算机求解

例 (6-17)

$$\begin{aligned} \min \quad & (-2x_1 - x_2 - 4x_3 - 3x_4 - x_5) \\ \text{x s.t.} \quad & \begin{cases} 2x_2 + x_3 + 4x_4 + 2x_5 \leqslant 54 \\ 3x_1 + 4x_2 + 5x_3 - x_4 - x_5 \leqslant 62 \\ x_1, x_2 \geqslant 0, , x_3 \geqslant 3.32, x_4 \geqslant 0.678, x_5 \geqslant 2.57 \end{cases} \end{aligned}$$

MATLAB 代码:

```
f = [−2;−1;−4;−3;−1]; A = [0 2 1 4 2; 3
      4 5 −1 −1]; B = [54; 62];
Ae=[]; Be=[]; xm=[0 0 3.32 .678 2.57];
ff5_0 = optimset; ff5_0.LargeScale = '
      off'; ff5_0.Display = 'iter';
ff5_0.TolX = 1e−15; ff5_0.TolFun = 1e
      −10; ff5_0.TolCon = 1e−9;
[x, f_opt, key, c] = linprog(f, A, B, Ae,
      Be, xm, [], [], ff5_0)
```

输出:

```
Iter = 2, Time = 0.013 .....

x =
19.7850
0
3.3200
11.3850
2.5700

f_opt = −89.5750
```

二次型规划的求解

二次型规划问题提法：

$$\min \quad \left(\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x}+\mathbf{f}^{\mathrm{T}}\mathbf{x}\right)$$

$$\mathbf{x}\ \text{s.t.} \begin{cases} \mathbf{A}\mathbf{x} \leqslant \mathbf{B} \\ \mathbf{A}_{\mathrm{eq}}\mathbf{x} = \mathbf{B}_{\mathrm{eq}} \\ \mathbf{x}_m \leqslant \mathbf{x} \leqslant \mathbf{x}_M \end{cases}$$

quadprog() 函数调用：

```
[x,f_opt,flag,c]=quadprog(H,f,A,B,A_eq,B_eq,x_m,x_M,x_0,OPT,p1,p2,...)
```

例 (6-18)

$$\min \quad \left[(x_1-1)^2+(x_2,+2)^2+(x_3-3)^2+(x_4-4)^2\right]$$

$$\mathbf{x}\ \text{s.t.} \begin{cases} x_1+x_2+x_3+x_4 \leqslant 5 \\ 3x_1+3x_2+2x_3+x_4 \leq 10 \\ x_1,x_2,x_3,x_4 \geqslant 0 \end{cases}$$

输出：

```
Iter = 5..... x =(0.0000,0.6667,1.6667,2.6667); f_opt =−23.6667
```

一般非线性规划问题的求解

一般非线性规划问题提法

$$\min \quad f(\mathbf{x})$$

$$\mathbf{x}\ \text{s.t.} \begin{cases} \mathbf{A}\mathbf{x} \leqslant \mathbf{B} & , \mathbf{A}_{\text{eq}}\mathbf{x} = \mathbf{B}_{\text{eq}} \\ \mathbf{x}_m \leqslant \mathbf{x} \leqslant \mathbf{x}_M & , \mathbf{C}(\mathbf{x}) \leqslant 0 \\ \mathbf{C}_{eq}(\mathbf{x}) = 0 & . \end{cases}$$

fmincon() 函数调用：

```
[x,f_opt,flag,c]=fmincon(F,x0,A,B,A_eq,B_eq,x_m,x_M,CF,OPT,p1,p2,...)
%全局最优解尝试，则调fmincon_global()函数，规则同上。
```

例 (6-19)

$$\min \quad [1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3]$$

$$\mathbf{x}\ \text{s.t.} \begin{cases} x_1^2 + x_2^2 + x_3^2 - 25 = 0 \\ 8x_1 + 14x_2 + 7x_3 - 56 = 0 \\ x_1, x_2, x_3 \geqslant 0 \end{cases}$$

一般非线性规划问题的求解

MATLAB 代码

```
ff8_0 = optimset; ff8_0.LargeScale = 'off'; ff8_0.Display = 'iter';
ff8_0.TolX = 1e-15; ff8_0.TolFun = 1e-30; ff8_0.TolCon = 1e-20;
x0=ones(3,1); xm=zeros(3,1); xM=[]; A=[]; B=[]; Aeq=[]; Beq=[];
[x,f_opt,flag,c] = fmincon(@opt_fun1, x0, A, B, Aeq, Beq, xm, xM, @opt_con1,
      ff8_0)
```

输出

```
x =
3.5121
0.2170
3.5522

f_opt =
961.7152

iterations: 16
```

其他

学习心得 i

一　这是本系统具体的书，可以了解，明白 MATLAB 能解决的问题，有实际需要时再详细翻阅。

二　MATLAB 集成很多科学计算算法，也方便调用。我联想到了 FFTW、LAPACK 等，让用户避免重复造轮子。但我认为需要对算法有些了解，定性明白它的长处和不足。