数值最优化算法及应用

张鑫航

23010050

国防科技大学

版本：1.0

更新：2024 年 6 月 7 日

使用LaTeX生成。

目录

1 基于 Armjio 步长的梯度下降法

1.1 问题

利用基于 Armjio 步长梯度下降法求解以下问题：

$$f(\boldsymbol{x}) = 100 \times (x_2 - x_1^2)^2 + (1 - x_1)^2 \tag{1}$$

1.2 方法

最优步长的缺陷：

- 最优步长的初衷是在每一迭代步使目标函数下降量达到最大. 但计算量太大，而且不好求.
- 最优化问题关注全局最优值点。集中于某个方向上的线搜索似乎没有必要。

设 $\boldsymbol{d}_k$ 为 $\boldsymbol{x}_k$ 的下降方向，即 $\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{g}_k < 0$，对 $\sigma \in (0,1)$，只要 $\alpha > 0$ 充分小，有

$$f(\boldsymbol{x}_k + \alpha\boldsymbol{d}_k) = f(\boldsymbol{x}_k) + \alpha\boldsymbol{g}_k\boldsymbol{d}_k + o(\alpha) < 0$$

步长选取：先取一个较大的步长, 看目标函数是否有满意的下降量。否则依次按比例压缩，直至满足要求。又称进退试探法。

定义 1.1 (Armijo 步长规则) 取常数 $\beta > 0, \sigma, \gamma \in (0,1)$。步长 $\alpha_k = \beta\gamma^{m_k}$，其中 m_k 为 $0, 1, \cdots$ 中满足下式的最小值

$$f(\boldsymbol{x}_k + \beta\gamma^{m_k}\boldsymbol{d}_k) \leqslant f_k + \sigma\beta\gamma^{m_k}\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k$$

如 $\alpha_k < \beta$，则

$$\begin{aligned} f(\boldsymbol{x}_k + \beta\gamma^{m_k}\boldsymbol{d}_k) &\leqslant f(\boldsymbol{x}_k) + \sigma\beta\gamma^{m_k}\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k \\ f(\boldsymbol{x}_k + \beta\gamma^{m_k - 1}\boldsymbol{d}_k) &> f(\boldsymbol{x}_k) + \sigma\beta\gamma^{m_k-1}\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k \end{aligned}$$

将 Armijo 步长规则写为伪代码见算法 1.1。

算法 1.1: Armijo 步长规则

Input: 当前迭代点 $\boldsymbol{x}$, 下降方向 $\boldsymbol{d}$, 优化函数 f, 梯度函数 ∇f

Output: Armijo 步长 α

$m_k = 0, \beta > 0, \sigma, \gamma \in (0,1)$
while $f(\boldsymbol{x} + \beta\gamma^{m_k}\boldsymbol{d}) > f(\boldsymbol{x}) + \sigma\beta\gamma^{m_k}\nabla f(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{d}$ **do**
    if $f(\boldsymbol{x} + \beta\gamma^{m_k}\boldsymbol{d}) \leqslant f(\boldsymbol{x}) + \sigma\beta\gamma^{m_k}\nabla f(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{d}$ **then**
        break
    else
        $m_k \leftarrow m_k + 1$
    end
end
$\alpha \leftarrow \beta\gamma^{m_k}$
return α

1.3 结果

运行 “main.m” 得到迭代序列如图 1。

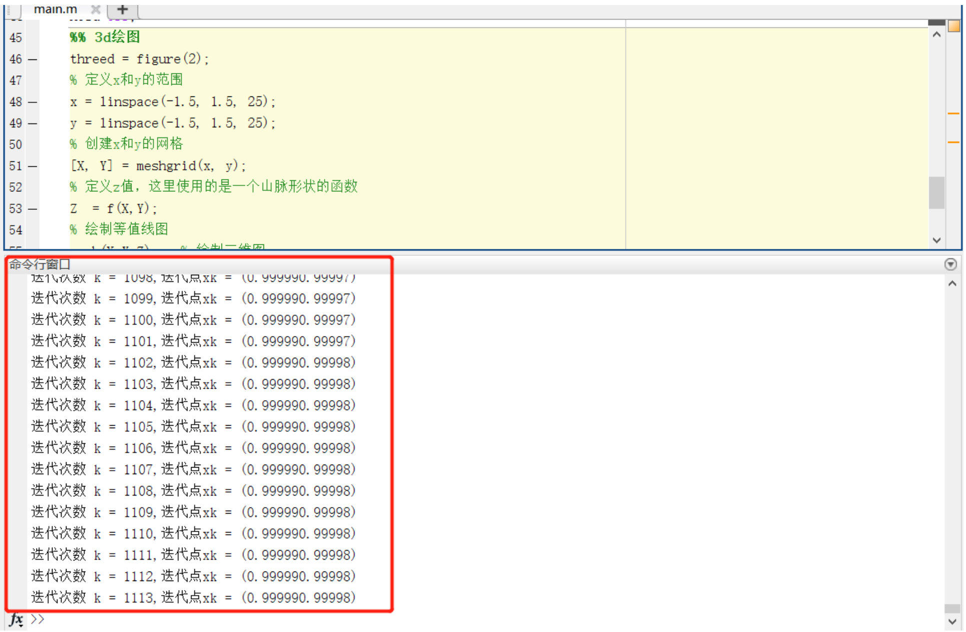
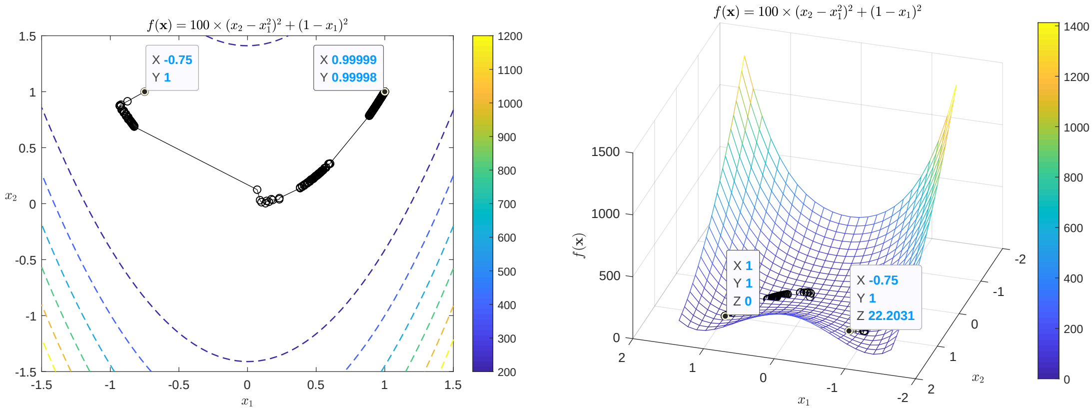


图 1: 运行 ‘main.m’ 得到迭代序列

算法求解问题 (1) 得到的数值最优解 $\boldsymbol{x}^*$

$$\boldsymbol{x}^* = (0.999990, 0.99998)^{\mathrm{T}}$$

算法迭代过程如图 2。



(a) 基于 Armjio 步长的梯度下降法 2d 迭代　(b) 基于 Armjio 步长的梯度下降法 3d 迭代

图 2: 基于 Armjio 步长的梯度下降法

1.4 代码

Armjio 步长和主函数见代码 1和 2。

Listing 1: Armjio 步长代码

```
function [alpha_armijo] = armijo(x,d,f0,f1)
% x是每次循环中得到的函数的解，d是下降方向
% f0是原目标函数， f1是目标函数的一次导数
alpha = 1; % a>0
sigma = 0.4; % 取值范围(0,1)越大越慢
gamma = 0.5; % 取值范围(0,0.5)越大越快
j = 1; %循环标志
    while (j>0)
        x_new = x+alpha.*d;
        if f0(x_new(1),x_new(2))<=f0(x(1),x(2))+sigma*alpha.*f1(x(1),x(2))'*d %检验条件
            j = 0; % 循环break
            alpha_armijo = alpha;
        else
            alpha = alpha*gamma; %缩小alpha, 进入下一次循环
        end
    end
end
```

Listing 2: 主函数代码

```
%%
```

```
clc;
clear;
%% 函数
syms x1 x2;
%这是一个Rosenbrock函数，用来测试最佳化演算法性能的非凸函数，也被称为香蕉函数。该函数的特点是只有一个全局最低点。
f = 100*(x2-x1^2)^2+(1-x1)^2;
f = matlabFunction(f); % 将符号表达式转换为函数句柄或文件
%% 绘图
% 画出函数的水平集，然后hold on，待会会在这个图上显示出每一个迭代步长和迭代结果
figure(1); clf; fc = fcontour(f,[-1.5 1.5 -1.5 1.5]); axis equal; hold on
fc.LineWidth = 1;
fc.LineStyle = '--';
fc.LevelList = [1200 1000 800 600 400 200 0];
colorbar;
xlabel('$x_1$', 'Interpreter', 'latex', 'fontsize', 10.5);
ylabel('$x_2$', 'Interpreter', 'latex', 'fontsize', 10.5, 'rotation', pi/2);
title('$f(\mathbf{x}) = 100\times (x_2-x_1^2)^2+(1-x_1)^2$', 'Interpreter', 'latex', 'fontsize', 10.5);
%% 初始设置
f_sym = sym(f); %将函数句柄转换回符号表达式
F_td = matlabFunction(gradient(f_sym));% 求一次导数
x0 = [-3/4 1]'; % 给一个收敛的起始点
eps = 1e-5; % 设可容误差为一个很接近0的数
k=0; % 设k=0，用来记录迭代次数
f_td = F_td(x0(1),x0(2)); %计算一次导数在起始点的值
%% 开始迭代
while norm(f_td) > eps   %设置一个条件为一次导数的值大于可容误差的循环
    d= - f_td; % 计算下降方向（最速下降法的定义）
    [alpha_armijo] = armijo(x0,d,f,F_td); %嵌套Armijo线搜法
    xnew = x0 + alpha_armijo*d; %在起始点上通过Armijo得到的步长迭代出新的点。
     % 在上面hold on的水平集里面画出新的迭代结果（点）。
    plot([x0(1) xnew(1)],[x0(2) xnew(2)],'ko-')
    refresh
```

```
    x0 = xnew; %用迭代点覆盖起始点，每次循环就覆盖上一个点
    f_td = F_td(x0(1),x0(2)); %计算出函数的一次导在迭代点上面的值，用于循环
        条件的判断
    disp(['迭代次数 k = ',num2str(k), ',迭代点xk = (', num2str(x0(1)),
        num2str(x0(2)), ')']);
    k = k+1; %迭代次数+1
end
x = x0; %最后循环结束得到的最终迭代点就是函数的最优点
result = f(x(1),x(2)); %函数在该最优点的值（局部最小值）
```

2 有约束优化（二次规划问题）

2.1 问题

利用有效集方法求解下述问题

$$
\begin{aligned}
&\min f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x} + \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\
&\mathrm{s.t.}\, \boldsymbol{A}_e\boldsymbol{x} - \boldsymbol{b}_e = \boldsymbol{0} \\
&\qquad \boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i \geqslant \boldsymbol{0}
\end{aligned}
$$

其中

$$
\boldsymbol{H} = \begin{bmatrix} 2 & -1 \\ -1 & 4 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 1 \\ 10 \end{bmatrix}
$$
$$
\boldsymbol{A}_e = [], \quad \boldsymbol{b}_e = []
$$
$$
\boldsymbol{A}_i = \begin{bmatrix} -3 & -2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{b}_i = \begin{bmatrix} -6 \\ 0 \\ 0 \end{bmatrix}
$$

或者写为

$$
\begin{aligned}
&\min\, f(\boldsymbol{x}) = x_1^2 + 2x_2^2 - x_1x_2 + x_1 + 10x_2 \\
&\mathrm{s.t.}\, -3x_1 - 2x_2 \geqslant -6 \\
&\qquad x_1 \geqslant 0 \\
&\qquad x_2 \geqslant 0
\end{aligned} \tag{2}
$$

其约束的有效域为如图 3。

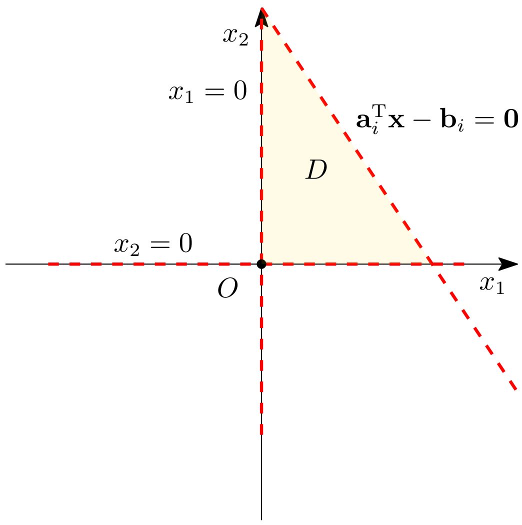


图 3: 问题 (2) 的有效域

2.2 算法

算法过程如图 4。

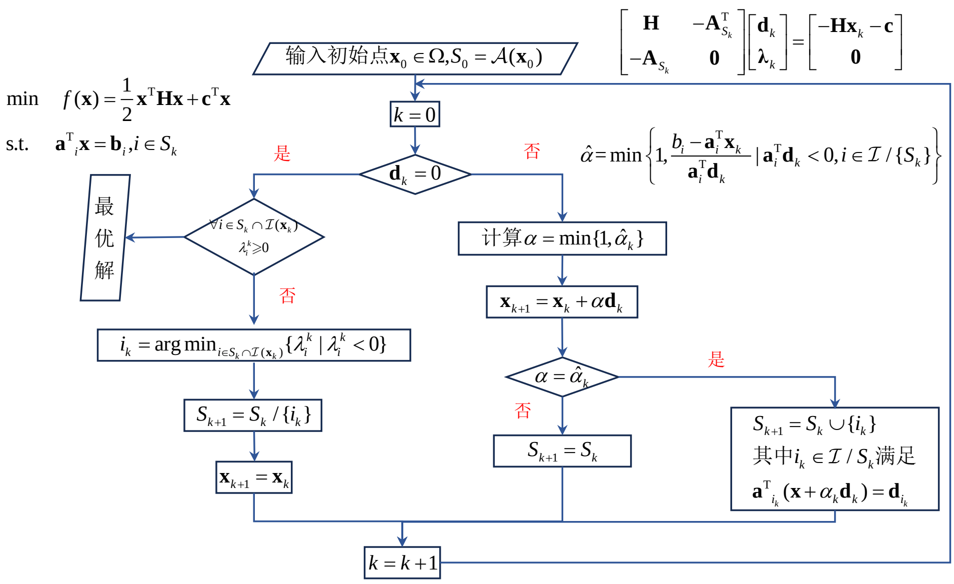


图 4: 有效集方法流程图

2.3 结果

运行 “active_set.m” 得到迭代序列如图 5。

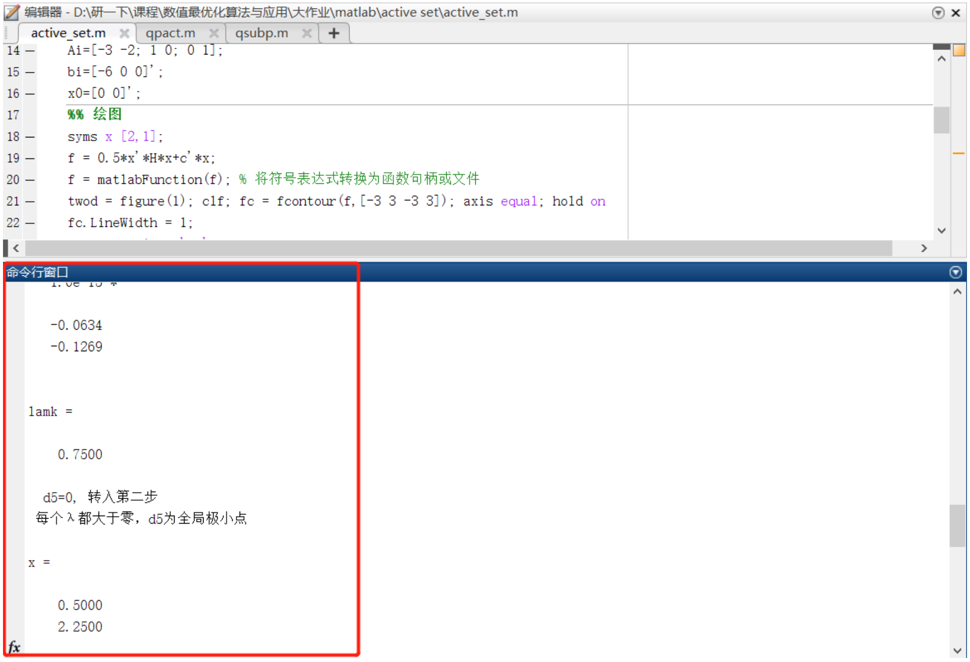
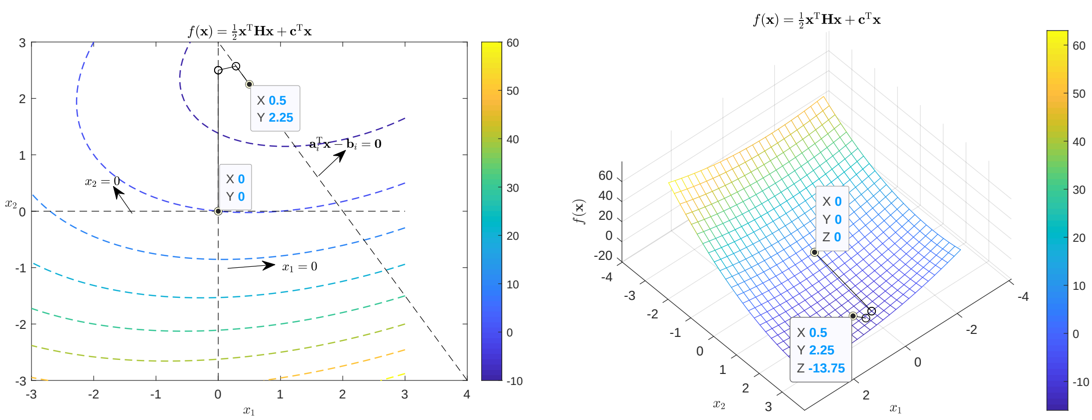


图 5: 运行 'active_set.m' 得到迭代序列

算法得到的数值最优解 $\boldsymbol{x}^*$

$$\boldsymbol{x}^* = (0.5000, 2.2500)^{\mathrm{T}}$$

算法迭代过程如图 6。



(a) 基于有效集方法的梯度下降法 2d 迭代　(b) 基于有效集方法的梯度下降法 3d 迭代

图 6: 有效集方法

2.4 代码

有效集方法求解、子问题求解和主函数分别见代码 3、4和 5。

Listing 3: 有效集方法代码

```
function [x,lamk,exitflag,output, threedx1, threedx2]=qpact(H,c,Ae,be,Ai,bi
    ,x0)
% 功能: 用有效集方法解一般约束二次规划问题:
%   min f(x)=0.5*x'*H*x+c'*x,
%   s.t.a'_i*x-b_i=0,(i=1,...,l),
%       a'_i*x-b_i>=0,(i=l+1,...,m)
% 输入: x0是初始点, H, c分别是目标函数二次型矩阵和向量;
%   Ae=(a_1,...,a_l)',  be=(b_1,...,b_l)';
%   Ai=(a_{l+1},...,a_m), bi=(b_{l+1},...,b_m)'.
% 输出:  x是最优解,  lambda是对应的乘子向量; output是结构
% 变量, 输出极小值f(x), 迭代次数k等信息, exitflag是算法终止类型

    % 绘图
    threedx1 = [x0(1)]; threedx2 = [x0(2)];

    % 初始化
    %epsilon 是一个很小的数, 用于辅助不等式判断,
    %如在进行 >= 的判断时, 往往是 x>=b+epsilon, err也是一个类似的量
    epsilon=1.0e-9; err=1.0e-6;
    k=0; x=x0;  %k为迭代次数, x0为初始点
    n=length(x);  kmax=1.0e3;   %kmax最大迭代次数
    ne=length(be); ni=length(bi); lamk=zeros(ne+ni,1);
    index=ones(ni,1);
    for (i=1:ni)
        if(Ai(i,:)*x>bi(i)+epsilon), index(i)=0; end
    end
    %算法主程序
    while (k<=kmax)
        disp(['迭代次数 k = ',num2str(k), ',迭代点x',num2str(k) ,'= (',
            num2str(x(1)), ',' ,num2str(x(2)), ')']);
        %求解子问题
        Aee=[ ];
        if(ne>0),  Aee=Ae; end
        for(j=1:ni)         %不等式约束的个数
            if(index(j)>0), Aee=[Aee; Ai(j,:)]; end    %将不等式约束和等式
                约束的稀疏矩阵合并
```

```
        end
    %   min f(x)=0.5*x'*H*x+c'*x,
    %    s.t.  a'_i*x-b_i=0,(i=1,...,l, l+1,...,m),
    %    将不等式约束改为等式约束求解子问题。
        gk=H*x+c;
        [m1,n1] = size(Aee);
        [dk,lamk]=qsubp(H,gk,Aee,zeros(m1,1));       %计算出极小点dk和拉格朗日乘子向量lamk
        disp('  求解子问题');
        dk
        lamk
        if(norm(dk)<=err)                            %dk为0的时候转入第二步
            disp(['  d',num2str(k),'=0, ','转入第二步']);
            y=0.0;
            if(length(lamk)>ne)
                [y,jk]=min(lamk(ne+1:length(lamk))); %确定有效集中lambda的最小元素。
            end
            if(y>=0)
                exitflag=0;                          %如果每个lambda都大于零，则dk为全局极小点，
                disp([' 每个 都大于零，','d',num2str(k),'为全局极小点']);
            else                                     %否则减去lamda对应的有效集元素，形成新的有效集
                exitflag=1;
                for(i=1:ni)
                    if(index(i) && (ne+sum(index(1:i)))==jk)   %如果lambda对应的有效集位置为jk，且索引为1，则将索引置0
                        disp(['  更新有效集S',num2str(k),',去掉i',num2str(k),'=',num2str(i),'和x']);
                        xnew = x;
                        index(i)=0; break;           %确保在之后的计算中，不在计算当前不等式约束。
                    end
                end
            end
%            k=k+1;
```

```
%             disp(['迭代次数 k = ',num2str(k), ',迭代点x',num2str(k) ,'=
    (', num2str(x(1)), ',' ,num2str(x(2)), ')']);
        else                                            %如果dk不等于0，转入第
            三步
            disp(['  d',num2str(k),' 0， ','转入第三步，确定步长 ']);
            exitflag=1;                                 %确定步长alpha
            %求步长
            alpha=1.0; tm=1.0;
            for(i=1:ni)
                if((index(i)==0)&(Ai(i,:)*dk<0))
                    tm1=(bi(i)-Ai(i,:)*x)/(Ai(i,:)*dk);  %alpha的计算见第三
                        步的具体公式
                    if(tm1<tm)
                        tm=tm1; ti=i;
                    end
                end
            end
            disp('  更新有效集和x');
            alpha=min(alpha,tm);                           %选取最小的alpha,
                一般为边界点对应的alpha
            xnew = x+alpha*dk;                              %确定新的x位置。
            %修正有效集
            if(tm<1), index(ti)=1; end
        end
        if(exitflag==0), break; end
        plot([x(1) xnew(1)],[x(2) xnew(2)],'ko-');
        refresh;
        x = xnew;
        threedx1 = [threedx1;x(1)]; threedx2 = [threedx2;x(2)];
        k=k+1;
    end

    output.fval=0.5*x'*H*x+c'*x;
    output.iter=k;
end
```

Listing 4: 子问题代码

```
function [x,lambda]=qsubp(H,c,Ae,be)
%求解子问题即，求解一个等式约束的二次规划，
    ginvH=pinv(H);
    [m,n]=size(Ae);
    if (m>0)
        rb = Ae*ginvH*c + be;
        lambda = pinv(Ae*ginvH*Ae')*rb;
        x = ginvH*(Ae'*lambda-c);
    else
        x = -ginvH*c;
        lambda = zeros(m,1);
    end
end
```

Listing 5: 主函数代码

```
%%
clc;
clear;
%% 问题
%输入： x0是初始点，H，c分别是目标函数二次型矩阵和向量；
%   Ae=(a_1,...,a_l)',  be=(b_1,...,b_l)';
%   Ai=(a_{l+1},...,a_m), bi=(b_{l+1},...,b_m)'.
%   min f(x)=0.5*x'*H*x+c'*x,
%   s.t. a'_i*x-b_i=0,(i=1,...,l),
%        a'_i*x-b_i>=0,(i=l+1,...,m)
H=[2 -1; -1 4];
c=[-1 -10]';
Ae=[ ]; be=[ ];
Ai=[-3 -2; 1 0; 0 1];
bi=[-6 0 0]';
x0=[0 0]';
%% 绘图
syms x [2,1];
f = 0.5*x'*H*x+c'*x;
f = matlabFunction(f); % 将符号表达式转换为函数句柄或文件
twod = figure(1); clf; fc = fcontour(f,[-3 3 -3 3]); axis equal; hold on
fc.LineWidth = 1;
```

```matlab
fc.LineStyle = '--';
colorbar;
xlabel('$x_1$', 'Interpreter', 'latex', 'fontsize', 10.5);
ylabel('$x_2$', 'Interpreter', 'latex', 'fontsize', 10.5, 'rotation', pi/2);
title('$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x}+\mathbf{c}^{\mathrm{T}}\mathbf{x}$',...
    'Interpreter', 'latex', 'fontsize', 10.5);
%% 求解
[x,lambda,exitflag,output,threedx1, threedx2]=qpact(H,c,Ae,be,Ai,bi,x0)
hold off;
%% 3d绘图
threed = figure(2);
% 定义x和y的范围
x = linspace(-3, 3, 25);
y = linspace(-3, 3, 25);
% 创建x和y的网格
[X, Y] = meshgrid(x, y);
% 定义z值，这里使用的是一个山脉形状的函数
Z  = f(X,Y);
threedf = f(threedx1, threedx2);
% 绘制等值线图
mesh(X,Y,Z);   % 绘制三维图
hold on;
axis square;  % 坐标轴之间的尺度相等
plot3(threedx1, threedx2, threedf, 'ko-');
xlabel('$x_1$', 'Interpreter', 'latex', 'fontsize', 10.5);
ylabel('$x_2$', 'Interpreter', 'latex', 'fontsize', 10.5, 'rotation', pi/2);
zlabel('$f(\mathbf{x})$', 'Interpreter', 'latex', 'fontsize', 10.5);
title('$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x}+\mathbf{c}^{\mathrm{T}}\mathbf{x}$',...
    'Interpreter', 'latex', 'fontsize', 10.5);
```

3 手工例题

3.1 有效集方法

课程名称:

姓名: 张鑫航

学号: 23010050

用有效集方法求解

$$\min \ f(x) = x_1^2 + x_2^2 - 2x_1 - 4x_2$$

$$\text{s.t.} \quad -x_1 - x_2 + 1 \geq 0 \quad \cdots \quad 1$$

$$x_1 \geq 0 \quad \cdots \quad 2$$

$$x_2 \geq 0 \quad \cdots \quad 3$$

解:

第1步迭代:

取初始点: $x^{(0)} = (0,0)^T$, $S_0 = A(x^{(0)}) = \{2,3\}$ 约束2,3取等.

求解等式约束优化子问题:

$$\min \ (0+d_1)^2 + (0+d_2)^2 - 2(0+d_1) - 4(0+d_2)$$

$$= d_1^2 + d_2^2 - 2d_1 - 4d_2$$

$$\text{s.t.} \quad d_1 = 0$$

$$d_2 = 0$$

KKT条件:
$$\begin{cases} 2d_1 - 2 - \lambda_1 = 0 \\ 2d_2 - 4 - \lambda_2 = 0 \\ d_1 = 0 \\ d_2 = 0 \end{cases}$$
得 $d^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ $\lambda^{(0)} = \begin{pmatrix} -2 \\ -4 \end{pmatrix}$

因为 $d^{(0)} = 0$, 新的迭代点为 $x^{(1)} = x^{(0)} = (0,0)^T$

因为 $\lambda_i < 0$, 修正有效集 $S_1 = S_0 \setminus \{3\} = A_0 \setminus \{3\} = \{2\}$.

第2次迭代:

求解子问题

$$\min \ (0+d_1)^2 + (0+d_2)^2 - 2(0+d_1) - 4(0+d_2)$$

$$= d_1^2 + d_2^2 - 2d_1 - 4d_2$$

$$\text{s.t.} \quad d_1 = 0$$

KKT条件:
$$\begin{cases} 2d_1 - 2 - \lambda_1 = 0 \\ 2d_2 - 4 = 0 \\ d_1 = 0 \end{cases}$$
得 $d^{(1)} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$, $\lambda^{(1)} = -2$

共 页, 第 页

因为 $d^{(1)}\neq 0$，转第三步，确定步长 α，　$I/\{S_1\}=\{1,3\}$.

$$\alpha_1=\min\left\{1,\ \frac{b_i-a_i^T x^{(1)}}{a_i^T d^{(1)}}\ \middle|\ i\in\{1,3\},\ a_i^T d^{(1)}<0\right\}$$

$$=\min\left\{1,\ \frac{-1-(-1,-1)\cdot(0,0)^T}{(-1,-1)\cdot(0,2)^T},\ \cancel{\frac{0-(0,1)\cdot(0,0)^T}{(0,1)\cdot(0,2)^T}}\right\}$$

因为 $(0,1)\cdot(0,2)^T>0$

$$=\min\left\{1,\ \frac{1}{2}\right\}=\frac{1}{2}\text{，因为 }\alpha_1=\hat{\alpha}_1=\frac{1}{2}$$

更新 x

$$x^{(2)}=x^{(1)}+\alpha_1 d^{(1)}=(0,0)^T+\frac{1}{2}(0,2)^T=(0,1)^T$$

从 $I/S_1=\{1,3\}$ 中取

$a_1^T x^{(2)}=(-1,-1)\cdot(0,1)^T=-1=b_1$　知，取 $i_k=1$.

$a_3^T x^{(2)}=(0,1)\cdot(0,1)^T=1\neq b_3$

更新 $S_2=S_1\cup\{1\}=\{1,2\}$.

进入第3次迭代

求解子问题 $\min\ (0+d_1)^2+(1+d_2)^2-2d_1-4(d_1+1)$

$$=d_1^2+d_2^2-2d_1-2d_2+C$$

$$\text{s.t.}\ \ -d_1-d_2=0$$

$$d_1=0$$

KKT条件

$$\begin{cases}2d_1-2+\lambda_1=0\\ 2d_2-2+\lambda_1-\lambda_2=0\\ d_1+d_2=0\\ d_1=0\end{cases}\quad\text{得 } d^{(2)}=\begin{pmatrix}0\\0\end{pmatrix},\ \lambda^{(2)}=\begin{pmatrix}2\\0\end{pmatrix}$$

由 $d^{(2)}=0$，且对于 $\forall i\in S_2\cap I(x^{(2)})=\{1,2\}$，$\lambda_i^{(2)}\geq 0$，算法停止.

原问题最优解和最优 Lagrange 乘子分别为

$$x^*=x^{(2)}=\begin{pmatrix}0\\1\end{pmatrix},\qquad \lambda^*=(2,0,0)^T$$

$f(x^*)=0^2+1-0-4=-3$

例 3.1 求解二次规划问题☆☆☆☆☆

$$
\begin{aligned}
\min \quad & Q(\boldsymbol{x}) = x_1^2 + x_2^2 - 2x_1 - 4x_2 \\
\text{s.t.} \quad & -x_1 - x_2 + 1 \geqslant 0 \\
& x_1, x_2 \geqslant 0
\end{aligned} \tag{3}
$$

解 $k=0$:

取初始点

$$
\boldsymbol{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},\ S_0 = \mathcal{A}(\boldsymbol{x}^{(0)}) = \{2, 3\}
$$

求解等式约束优化子问题

$$
\begin{aligned}
\min \quad & d_1^2 + d_2^2 - 2d_1 - 4d_2 \\
\text{s.t.} \quad & d_1 = 0 \\
& d_2 = 0
\end{aligned}
$$

KKT 条件

$$
\begin{cases}
2d_1 - 2 - \lambda_1 = 0 \\
2d_2 - 4 - \lambda_2 = 0 \\
d_1 = 0 \\
d_2 = 0
\end{cases}
$$

得最优解和相应的 Lagrange 乘子

$$
\boldsymbol{d}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},\ \boldsymbol{\lambda}^{(0)} = \begin{pmatrix} -2 \\ -4 \end{pmatrix}
$$

因为 $\boldsymbol{d}^{(0)} = \mathbf{0}$，新的迭代点为

$$
\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}
$$

因为 $\lambda_i < 0$，修正指标集

$$
S_1 = S_0 / \{3\} = \mathcal{A}_0 / \{3\} = \{2\}
$$

$k=1$:

求解子问题

$$
\begin{aligned}
\min \quad & d_1^2 + d_2^2 - 2d_1 - 4d_2 \\
\text{s.t.} \quad & d_1 = 0
\end{aligned}
$$

KKT 条件

$$
\begin{cases}
2d_1 - 2 - \lambda_1 = 0 \\
2d_2 - 4 = 0 \\
d_1 = 0
\end{cases}
$$

得最优解

$$\boldsymbol{d}^{(1)}=\begin{pmatrix}0\\2\end{pmatrix},\ \boldsymbol{\lambda}^{(1)}=\begin{pmatrix}-2\end{pmatrix}$$

因为 $\boldsymbol{d}^{(1)}\neq\mathbf{0}$，转第三步，计算步长 α，其中 $i\in\mathcal{I}/S_1=\{1,2,3\}\,/\,\{2\}=\{1,3\}$
因为

$$\boldsymbol{a}_1^{\mathrm{T}}\boldsymbol{d}^{(1)}=\begin{pmatrix}-1 & -1\end{pmatrix}\cdot\begin{pmatrix}0\\2\end{pmatrix}=-2$$

$$\boldsymbol{a}_2^{\mathrm{T}}\boldsymbol{d}^{(1)}=\begin{pmatrix}0 & 1\end{pmatrix}\cdot\begin{pmatrix}0\\2\end{pmatrix}=2>0$$

$$\begin{aligned}\alpha_1&=\min\left\{1,\frac{b_i-\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}^{(1)}}{\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{d}^{(1)}}\mid i=1,3,\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{d}^{(1)}<0\right\}\\&=\min\left\{1,\frac{-1-\begin{pmatrix}-1 & -1\end{pmatrix}\cdot\begin{pmatrix}0\\0\end{pmatrix}}{-2}\right\}=\frac{1}{2}\end{aligned}$$

从 $\mathcal{I}/S_1=\{1,3\}$ 中取

$$\boldsymbol{a}_1^{\mathrm{T}}\boldsymbol{x}^{(2)}=\begin{pmatrix}-1 & -1\end{pmatrix}\cdot\begin{pmatrix}0\\1\end{pmatrix}=-1=\boldsymbol{b}_1$$

$$\boldsymbol{a}_3^{\mathrm{T}}\boldsymbol{x}^{(2)}=\begin{pmatrix}0 & 1\end{pmatrix}\cdot\begin{pmatrix}0\\1\end{pmatrix}=-1\neq\boldsymbol{b}_3=0$$

不等式积极约束 $i=1$ 令

$$\boldsymbol{x}^{(2)}=\boldsymbol{x}^{(1)}+\alpha_1\boldsymbol{d}^{(1)}=\begin{pmatrix}0\\1\end{pmatrix},\ S_2=S_1\cup\{1\}=\{1,2\}$$

$k=3$:

求解子问题

$$\begin{aligned}\min\quad & d_1^2+d_2^2-2d_1-2d_2\\ \text{s.t.}\quad & -d_1-d_2=0\\ & d_1=0\end{aligned}$$

KKT 条件

$$\begin{cases}2d_1-2+\lambda_1=0\\2d_2-2+\lambda_1-\lambda_2=0\\d_1+d_2=0\\d_1=0\end{cases}$$

得最优解

$$\boldsymbol{d}^{(2)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \boldsymbol{\lambda}^{(2)} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

由 $\boldsymbol{d}^{(2)} = 0$，且对于 $\forall i \in S_k \cap \mathcal{I}(\boldsymbol{x}^{(2)}) = \{1, 2\}, \lambda_i^{(2)} \geqslant 0$ 算法停止。原问题最优解和最优 Lagrange 乘子分别为：

$$\boldsymbol{x}^* = \boldsymbol{x}^{(2)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \boldsymbol{\lambda}^* = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}$$

$$f(\boldsymbol{x}^*) = -3$$

3.2 共轭梯度法

$\frac{\partial f}{\partial x_1}=2x_1$, $\frac{\partial f}{\partial x_2}=x_2$, $\frac{\partial f}{\partial x_3}=x_3$

姓 名：张鑫航

课程名称：

学 号：23010050

利用共轭梯度法求 $\min f(x)=x_1^2+\frac{1}{2}x_2^2+\frac{1}{2}x_3^2$ 或者方程 $Ax=b$，$A=\begin{bmatrix}2&&\\&1&\\&&1\end{bmatrix}$ $b=\begin{bmatrix}0\\0\\0\end{bmatrix}$

解：$k=0$：

取初始点 $x_0=(1,1,1)^T$，$g_0=(2,1,1)^T$，$\beta_{-1}=0$，$d_0=-g_0=(-2,-1,-1)^T$

$$f(x_0+\alpha d_0)=f(1-2\alpha,1-\alpha,1-\alpha)=(1-2\alpha)^2+\frac{1}{2}(1-\alpha)^2+\frac{1}{2}(1-\alpha)^2$$

$$=(4+\frac{1}{2}+\frac{1}{2})\alpha^2+(-4-1-1)\alpha+c$$

$$\alpha=\arg\min f(x_0+\alpha d_0)=-\frac{-4-1-1}{2(4+\frac{1}{2}+\frac{1}{2})}=\frac{3}{5}$$

$k=1$：

$x_1=x_0+\alpha_0 d_0=(1-\frac{3}{5}\times 2,1-\frac{3}{5},1-\frac{3}{5})^T=\frac{1}{5}(-1,2,2)^T$，$g_1=\frac{2}{5}(-1,1,1)^T$

$\beta_0=\frac{g_1^Tg_1}{g_0^Tg_0}=\frac{\frac{4}{25}(1+1+1)}{4+1+1}=\frac{2}{25}$，$d_1=-g_1+\beta_0 d_0=(\frac{2}{5}+\frac{2}{25}(-2),-\frac{2}{5}+\frac{2}{25}(-1),-\frac{2}{5}+\frac{2}{25}(-1))^T$

$$=\frac{6}{25}(1,-2,-2)^T$$

$$f(x_1+\alpha d_1)=f(-\frac{1}{5}+\frac{6}{25}\alpha,\frac{2}{5}+(\frac{-12}{25})\alpha,\frac{2}{5}-\frac{12}{25}\alpha)^T=(-\frac{1}{5}+\frac{6}{25}\alpha)^2+\frac{1}{2}(\frac{2}{5}-\frac{12}{25}\alpha)^2+\frac{1}{2}(\frac{2}{5}-\frac{12}{25}\alpha)^2$$

$$=\frac{1}{25^2}(-5+6\alpha)^2+\frac{1}{2\cdot 25^2}(10-12\alpha)^2+\frac{1}{2\cdot 25^2}(10-12\alpha)^2$$

$$=\frac{1}{25^2}[(36+72+72)\alpha^2+(-60-120-120)\alpha+c]$$

$$\alpha=\arg\min f(x_1+\alpha d_1)=-\frac{-60-120-120}{2(36+72+72)}=\frac{5}{6}$$

$k=2$；$x_2=x_1+\alpha d_1=(-\frac{1}{5}+\frac{5}{6}\cdot\frac{6}{25},\frac{2}{5}+\frac{5}{6}\cdot\frac{-12}{25},\frac{2}{5}+\frac{5}{6}\frac{(-12)}{25})^T=(0,0,0)^T$

$g_2=(0,0,0)^T$.

因为 $\|g_2\|=0$，终止

$x^*=(0,0,0)^T$，$\min f(x)=0$

共 页，第 页

例 3.2 利用共轭梯度法求 $\boldsymbol{Ax}=\boldsymbol{b}$ 或者说，利用共轭梯度法求 $\min x_1^2+\dfrac{1}{2}x_2^2+\dfrac{1}{2}x_3^2$ ★★★★★

其中，

$$\boldsymbol{A}=\begin{bmatrix}2&0&0\\0&1&0\\0&0&1\end{bmatrix},\quad \boldsymbol{b}=\begin{pmatrix}0\\0\\0\end{pmatrix}$$

解 取初始点 $\boldsymbol{x}_0=(1,1,1)^{\mathrm{T}}$，迭代过程：

$$\frac{\partial f}{\partial x_1}=2x_1,\ \frac{\partial f}{\partial x_2}=x_2,\ \frac{\partial f}{\partial x_3}=2x_3,$$

1. $\boldsymbol{x}_0=(1,1,1)^{\mathrm{T}}$，$\boldsymbol{g}_0=(2,1,1)^{\mathrm{T}}$，$\beta_{-1}=0$，$\boldsymbol{d}_0=-\boldsymbol{g}_0=(-2,-1,-1)^{\mathrm{T}}$.

$$\begin{aligned}f(\boldsymbol{x}_0+\alpha\boldsymbol{d}_0)\ &=f((1-2\alpha),\tfrac{1}{2}(1-\alpha),\tfrac{1}{2}(1-\alpha))\\&=(1-2\alpha)^2+\tfrac{1}{2}(1-\alpha)^2+\tfrac{1}{2}(1-\alpha)^2\\&=(4+\tfrac{1}{2}+\tfrac{1}{2})\alpha^2+(-4-1-1)\alpha+c\end{aligned}$$

$$\alpha_0=\arg\min f(\boldsymbol{x}_0+\alpha_0\boldsymbol{d}_0)=-\frac{-4-1-1}{2(4+\frac{1}{2}+\frac{1}{2})}=\frac{3}{5}$$

2. $\boldsymbol{x}_1=\boldsymbol{x}_0+\alpha_0\boldsymbol{d}_0=\frac{1}{5}(-1,2,2)^{\mathrm{T}}$，$\boldsymbol{g}_1=\frac{2}{5}(-1,1,1)^{\mathrm{T}}$，$\beta_0=\frac{\boldsymbol{g}_1^{\mathrm{T}}\boldsymbol{g}_1}{\boldsymbol{g}_0^{\mathrm{T}}\boldsymbol{g}_0}=\frac{2}{25}$

$$\boldsymbol{d}_1=-\boldsymbol{g}_1+\beta_0\boldsymbol{d}_0=-\frac{6}{25}(1,-2,-2)^{\mathrm{T}}$$

$$\begin{aligned}f(\boldsymbol{x}_1+\alpha\boldsymbol{d}_1)\ &=f((-\tfrac{1}{5}+\tfrac{6}{25}\alpha),\tfrac{1}{2}(\tfrac{2}{5}-\tfrac{12}{25}\alpha),\tfrac{1}{2}(\tfrac{2}{5}-\tfrac{12}{25}\alpha))\\&=\tfrac{1}{25^2}\left[(36+72+72)\alpha^2+(-60-120-120)\alpha+c\right]\end{aligned}$$

$$\alpha_1=\arg\min f(\boldsymbol{x}_1+\alpha_1\boldsymbol{d}_1)=-\frac{-60-120-120}{2(36+72+72)}=\frac{5}{6}$$

3. $\boldsymbol{x}_2=\boldsymbol{x}_1+\alpha_1\boldsymbol{d}_1=\boldsymbol{0}$，$\|\boldsymbol{g}_2\|=0$，终止。

知道 $\boldsymbol{x}^*=(0,0,0)^{\mathrm{T}}$, $\min f(\boldsymbol{x})=0$，迭代过程如下:

k	$\boldsymbol{x}_k$	$\boldsymbol{g}_k$	β_{k-1}	$\boldsymbol{d}_k$	α_k
0	$(1,1,1)^{\mathrm{T}}$	$(2,1,1)^{\mathrm{T}}$	0	$-(2,1,1)^{\mathrm{T}}$	$\frac{3}{5}$
1	$\frac{1}{5}(-1,2,2)^{\mathrm{T}}$	$\frac{1}{5}(-2,2,2)^{\mathrm{T}}$	$\frac{2}{25}$	$-\frac{6}{25}(1,-2,-2)^{\mathrm{T}}$	$\frac{5}{6}$
2	$(0,0,0)^{\mathrm{T}}$	$(0,0,0)^{\mathrm{T}}$			

3.3 DFP

用DPP方法求解

$$\min f(x) = 2x_1^2 + x_2^2 - 4x_1 + 2$$

解：易知：$x^* = (1,0)^T$，$\frac{\partial f}{\partial x_1} = 4x_1 - 4$，$\frac{\partial f}{\partial x_2} = 2x_2$

取初始值 $x_0 = (2,1)^T$，$H_0 = I$

$k=0$:

$x_0 = (2,1)^T$，$g_0 = (4,2)^T$，$d_0 = -H_0 g_0 = (-4,-2)^T$

$f(x_0 + \alpha d_0) = f(2-4\alpha, 1-2\alpha) = 2(2-4\alpha)^2 + (1-2\alpha)^2 - 4(2-4\alpha) + 2$

$= (32+4)\alpha^2 + (-32-4+16)\alpha + c$

$\alpha_0 = \arg\min f(x_0 + \alpha_0 d_0) = \frac{-32-4+16}{-2(32+4)} = \frac{5}{18}$

$k=1$:

$x_1 = x_0 + \alpha_0 d_0 = (2 + \frac{5}{18}(-4), 1 + \frac{5}{18}(-2))^T = (\frac{8}{9}, \frac{4}{9})^T$

$g_1 = (-\frac{4}{9}, \frac{8}{9})^T$，$s_0 = x_1 - x_0 = \alpha_0 d_0 = (-\frac{10}{9}, -\frac{5}{9})^T$

$y_0 = g_1 - g_0 = (-\frac{4}{9} - 4, \frac{8}{9} - 2)^T = (-\frac{40}{9}, -\frac{10}{9})^T$

$$H_1 = H_0 + \frac{s_0 s_0^T}{s_0^T y_0} - \frac{H_0 y y^T H_0}{y_0^T H_0 y_0} = \frac{1}{306}\begin{bmatrix} 86 & -38 \\ -38 & 305 \end{bmatrix}$$

$d_1 = -H_1 g_1 = \frac{12}{51}\begin{pmatrix} 1 \\ -4 \end{pmatrix}$

$f(x_1 + \alpha d_1) = f(\frac{8}{9} + \frac{12}{51}\alpha, \frac{4}{9} - \frac{12\times 4}{51}\alpha)$

$\alpha_1 = \arg\min f(x_1 + \alpha_1 d_1) = \frac{17}{36}$

$k=2$:

$x_2 = x_1 + \alpha_1 d_1 = (1,0)^T$ 终止

较大的矩阵计算:

$k=1$时, $s_0^T y_0 = \frac{10}{9}\cdot(-\frac{40}{9}) - \frac{5}{9}\cdot(-\frac{10}{9}) = \frac{400+50}{81} = \frac{450}{81}$

$s_0 s_0^T = (-\frac{10}{9}, -\frac{5}{9})^T \cdot (-\frac{10}{9}, -\frac{5}{9}) = \begin{pmatrix} \frac{100}{81} & \frac{50}{81} \\ \frac{50}{81} & \frac{25}{81} \end{pmatrix}$

$y_0^T H_0 y_0 = y_0^T y_0 = \frac{1600+100}{81} = \frac{1700}{81}$

$H_0 y_0 y_0^T H_0 = y_0 y_0^T = (-\frac{40}{9}, -\frac{10}{9})^T \cdot (-\frac{40}{9}, -\frac{10}{9}) = \begin{pmatrix} \frac{1600}{81} & \frac{400}{81} \\ \frac{400}{81} & \frac{100}{81} \end{pmatrix}$

$$H_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{81}{450}\cdot\frac{25}{81}\begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix} - \frac{81}{1700}\cdot\frac{100}{81}\begin{pmatrix} 16 & 4 \\ 4 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{1}{18}\begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix} - \frac{1}{17}\begin{pmatrix} 16 & 4 \\ 4 & 1 \end{pmatrix} = \frac{1}{306}\begin{pmatrix} 306+4\times17-16\times18 & 2\times17-4\times18 \\ 2\times17-4\times18 & 306+17-18 \end{pmatrix}$$

$$= \frac{1}{306}\begin{pmatrix} 86 & -38 \\ -38 & 305 \end{pmatrix}$$

$$d_1 = -H_1 g_1 = \frac{1}{306}\cdot\frac{1}{9}\begin{pmatrix} 86 & -38 \\ -38 & 305 \end{pmatrix}\begin{pmatrix} -4 \\ 8 \end{pmatrix} = \frac{1}{306\times9}\begin{pmatrix} -4\times86+8\times(-38) \\ -4\times(-38)+8\times305 \end{pmatrix}$$

$$= \frac{12}{51}(1, -4)^T$$

精确线搜索 $f(x_1+\alpha d_1) = 2(\frac{8}{9}+\frac{12}{51}\alpha)^2 + (\frac{4}{9}-\frac{48}{51}\alpha)^2 - 4(\frac{8}{9}+\frac{12}{51}\alpha) + 2$

$$= (\frac{12^2\cdot2}{51^2} + \frac{48^2}{51^2})\alpha^2 + (2\cdot2\cdot\frac{8}{9}\cdot\frac{12}{51} - 2\cdot\frac{4}{9}\cdot\frac{48}{51} - 4\cdot\frac{12}{51})\alpha + c$$

$$\alpha_1 = \arg\min f(x_1+\alpha d_1) = -\frac{(2\cdot2\cdot\frac{8}{9}\cdot\frac{12}{51} - 2\cdot\frac{4}{9}\cdot\frac{48}{51} - 4\cdot\frac{12}{51})}{2(\frac{12^2\cdot2}{51^2}+\frac{48^2}{51^2})} = \frac{17}{36}$$

$$x_2 = x_1 + \alpha_1 d_1 = (\frac{8}{9}+\frac{17}{36}\cdot\frac{12}{51}, \frac{4}{9}+\frac{17}{36}\cdot(-\frac{12\times4}{51}))^T$$

$$= (1, 0)^T$$