以下为两个 m 函数，分别是 qpsubp 和 sqpm，其中 qpsubp 用于求解二次规划子问题，sqpm 是用基于拉格朗日函数 Hesse 矩阵的 SQP 方法求解约束优化问题。具体的输入输出值意义已在程序中说明，具体是使用方法也在两段代码的末尾给出。

将以下代码除去示例部分，分别保存为两个.m 文件即可在 matlab 中使用下面给出两端代码供大家参考和交流：

代码 qpsub：

```
function [d,mu,lam,val,k]=qpsubp(dfk,Bk,Ae,hk,Ai,gk)
%功能 ：求解二次规划子问题
%输入: dfk 是 xk 处的梯度，Bk 是第 k 次近似 Hesse 矩阵，Ae,hk 线性等式约束有关的参数
%         Ai,gk 是线性不等式约束的有关参数
%输出: d,val 分别是最优解和最优值,mu,lam 是乘子向量,k 是迭代次数
n=length(dfk); l=length(hk); m=length(gk);
gamma=0.05; epsilon=1.0e-6; rho=0.5; sigma=0.2;
ep0=0.05; mu0=0.05*zeros(l,1); lam0=0.05*zeros(m,1);
d0=ones(n,1); u0=[ep0;zeros(n+l+m,1)];
z0=[ep0;d0;mu0;lam0];
k=0;
z=z0;ep=ep0;d=d0;mu=mu0;lam=lam0;
while(k<=150)
   dh=dah(ep,d,mu,lam,dfk,Bk,Ae,hk,Ai,gk);
   if(norm(dh)<epsilon)
      break;
   end
   A=JacobiH(ep,d,mu,lam,dfk,Bk,Ae,hk,Ai,gk);
   b=beta(ep,d,mu,lam,dfk,Bk,Ae,hk,Ai,gk,gamma)*u0-dh;
   dz=A\b;
   if(l>0&m>0)
      de=dz(1);dd=dz(2:n+1);du=dz(n+2:n+l+1);dl=dz(n+l+2:n+l+m+1);
   end
   if(l==0)
      de=dz(1);dd=dz(2:n+1);dl=dz(n+2:n+m+1);
   end
   if(m==0)
      de=dz(1);dd=dz(2:n+1);du=dz(n+2:n+l+1);
   end
   i=0;
   while(i<=20)
      if(l>0&m>0)
         dh1=dah(ep+rho^i*de,d+rho^i*dd,mu+rho^i*du,lam+rho^i*dl,dfk,Bk,Ae,hk,Ai,gk);
      end
      if(l==0)
         dh1=dah(ep+rho^i*de,d+rho^i*dd,mu,lam+rho^i*dl,dfk,Bk,Ae,hk,Ai,gk);
      end
```

```
        if(m==0)
            dh1=dah(ep+rho^i*de,d+rho^i*dd,mu+rho^i*du,lam,dfk,Bk,Ae,hk,Ai,gk);
        end
        if(norm(dh1)<=(1-sigma*(1-gamma*ep0)*rho^i)*norm(dh))
            mk=i;break;
        end
        i=i+1;
        if(i==20)
            mk=10;
        end
    end
    alpha=rho^mk;
    if(l>0&m>0)
        ep=ep+alpha*de;    d=d+alpha*dd;
        mu=mu+alpha*du; lam=lam+alpha*dl;
    end
    if(l==0)
        ep=ep+alpha*de;    d=d+alpha*dd;
        lam=lam+alpha*dl;
    end
    if(m==0)
        ep=ep+alpha*de;    d=d+alpha*dd;
        mu=mu+alpha*du;
    end
    k=k+1;
end
val=f1(d);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function p=phi(ep,a,b)
p=a+b-sqrt(a^2+b^2+2*ep^2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function dh=dah(ep,d,mu,lam,dfk,Bk,Ae,hk,Ai,gk)
n=length(dfk); l=length(hk); m=length(gk);
dh=zeros(n+l+m+1,1);
dh(1)=ep;
if(l>0&m>0)
    dh(2:n+1)=Bk*d-Ae'*mu-Ai'*lam+dfk;
    dh(n+2:n+l+1)=hk+Ae*d;
    for(i=1:m)
        dh(n+l+1+i)=phi(ep,lam(i),gk(i)+Ai(i,:)*d);
    end
end
if(l==0)
    dh(2:n+1)=Bk*d-Ai'*lam+dfk;
```

```
    for(i=1:m)
        dh(n+1+i)=phi(ep,lam(i),gk(i)+Ai(i,:)*d);
    end
end
if(m==0)
    dh(2:n+1)=Bk*d-Ae'*mu+dfk;
    dh(n+2:n+l+1)=hk+Ae*d;
end
dh=dh(:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function bet=beta(ep,d,mu,lam,dfk,Bk,Ae,hk,Ai,gk,gamma)
dh=dah(ep,d,mu,lam,dfk,Bk,Ae,hk,Ai,gk);
bet=gamma*norm(dh)*min(1,norm(dh));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [dd1,dd2,v1]=ddv(ep,d,lam,Ai,gk)
m=length(gk);
dd1=zeros(m,m); dd2=zeros(m,m); v1=zeros(m,1);
for(i=1:m)
    fm=sqrt(lam(i)^2+(gk(i)+Ai(i,:)*d)^2+2*ep^2);
    dd1(i,i)=1-lam(i)/fm;
    dd2(i,i)=1-(gk(i)+Ai(i,:)*d)/fm;
    v1(i)=-2*ep/fm;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function A=JacobiH(ep,d,mu,lam,dfk,Bk,Ae,hk,Ai,gk)
n=length(dfk); l=length(hk); m=length(gk);
A=zeros(n+l+m+1,n+l+m+1);
[dd1,dd2,v1]=ddv(ep,d,lam,Ai,gk);
if(l>0&m>0)
    A=[1,           zeros(1,n),    zeros(1,l),    zeros(1,m);
       zeros(n,1), Bk,             -Ae',          -Ai';
       zeros(l,1), Ae,             zeros(l,l),    zeros(l,m);
       v1,   dd2*Ai,        zeros(m,l),    dd1];
end
if(l==0)
    A=[1,         zeros(1,n),    zeros(1,m);
       zeros(n,1), Bk,           -Ai';
       v1,   dd2*Ai,        dd1];
end
if(m==0)
    A=[1,         zeros(1,n),    zeros(1,l);
       zeros(n,1), Bk,           -Ae';
       zeros(l,1), Ae,           zeros(l,l)];
end
```

%%%
例 解二次规划问题

min f(x)=x1^2+x1x2+2x2^2-6x1-2x2-12x3

s.t. x1+x2+x3-2=0

x1-2x2+3>=0

x1,x2,x3>=0

```
%%%%%%%%目标函数 f(x)%%%%%%%%%%%%%%
function f=f1(x)      %f1.m
f=x(1)^2+x(1)*x(2)+2*x(2)^2-6*x(1)-2*x(2)-12*x(3);
```

在 matlab 命令窗口依次输入下列命令:

```
dfk=[-6 -2 -12]';
Bk=[2 1 0;1 4 0;0 0 0];
Ae=[1 1 1];
hk=[-2]';
Ai=[1 -2 0;1 0 0;0 1 0;0 0 1];
gk=[3 0 0 0]';
[d,mu,lam,val,k]=qpsubp(dfk,Bk,Ae,hk,Ai,gk)
```

%%%
代码 sqpm

```
function [x,mu,lam,val,k]=sqpm(x0,mu0,lam0)
%功能: 用基于拉格朗日函数 Hesse 矩阵的 SQP 方法求解约束优化问题:
%    min f(x)      s.t.    h_i(x)=0, i=1,...,l
%输入: x0 是初始点，mu0,lam0 是乘子向量的初始值
%输出: x,mu 分别是近似最优点和相应的乘子
%val 是最优值,mh 是约束函数的模,k 是迭代次数
maxk=100;       %最大迭代次数
n=length(x0); l=length(mu0); m=length(lam0);
rho=0.5; eta=0.1; B0=eye(n);
x=x0; mu=mu0; lam=lam0;
Bk=B0; sigma=0.8;
epsilon1=1e-6; epsilon2=1e-5;
[hk,gk]=cons(x); dfk=df1(x);
[Ae,Ai]=dcons(x); Ak=[Ae;Ai];
k=0;
while(k<maxk)
   [dk,mu,lam]=qpsubp(dfk,Bk,Ae,hk,Ai,gk); %求解子问题
   mp1=norm(hk,1)+norm(max(-gk,0),1);
   if(norm(dk,1)<epsilon1)&(mp1<epsilon2)
      break;
   end          %检验终止准则
   deta=0.05;     %罚参数更新
```

```
    tau=max(norm(mu,inf),norm(lam,inf));
    if(sigma*(tau+deta)<1)
        sigma=sigma;
    else
        sigma=1.0/(tau+2*deta);
    end
    im=0;       %Armijo 搜索
    while(im<=20)
        temp=eta*rho^im*dphi1(x,sigma,dk);
        if(phi1(x+rho^im*dk,sigma)-phi1(x,sigma)<temp)
            mk=im;
            break;
        end
        im=im+1;
        if(im==20)
            mk=10;
        end
    end
    alpha=rho^mk; x1=x+alpha*dk;
    [hk,gk]=cons(x1); dfk=df1(x1);
    [Ae,Ai]=dcons(x1); Ak=[Ae;Ai];
    lamu=pinv(Ak)'*dfk;       %计算最小二乘乘子
    if(l>0&m>0)
        mu=lamu(1:l); lam=lamu(l+1:l+m);
    end
    if(l==0)
        mu=[]; lam=lamu;
    end
    if(m==0)
        mu=lamu; lam=[];
    end
    sk=alpha*dk;       %更新矩阵 Bk
    yk=dlax(x1,mu,lam)-dlax(x,mu,lam);
    if(sk'*yk>0.2*sk'*Bk*sk)
        theta=1;
    else
        theta=0.8*sk'*Bk*sk/(sk'*Bk*sk-sk'*yk);
    end
    zk=theta*yk+(1-theta)*Bk*sk;
    Bk=Bk+zk*zk'/(sk'*zk)-(Bk*sk)*(Bk*sk)'/(sk'*Bk*sk);
    x=x1;k=k+1;
end
val=f1(x);
%p=phi1(x,sigma)
```

```
%dd=norm(dk)
%%%%%%%%l1 精确价值函数%%%%%%%%%%%
function p=phi1(x,sigma)
f=f1(x); [h,g]=cons(x); gn=max(-g,0);
l0=length(h); m0=length(g);
if(l0==0)
    p=f+1.0/sigma*norm(gn,1);
end
if(m0==0)
    p=f+1.0/sigma*norm(h,1);
end
if(l0>0&m0>0)
    p=f+1.0/sigma*(norm(h,1)+norm(gn,1));
end
%%%%%%%%%价值函数的方向导数%%%%%%%%
function dp=dphi1(x,sigma,d)
df=df1(x); [h,g]=cons(x); gn=max(-g,0);
l0=length(h); m0=length(g);
if(l0==0)
    dp=df'*d-1.0/sigma*norm(gn,1);
end
if(m0==0)
    dp=df'*d-1.0/sigma*norm(h,1);
end
if(l0>0&m0>0)
    dp=df'*d-1.0/sigma*(norm(h,1)+norm(gn,1));
end
%%%%%%%%%拉格朗日函数 L(x,mu)%%%%%%%
function l=la(x,mu,lam)
f=f1(x);      %调用目标函数文件
[h,g]=cons(x);      %调用约束函数文件
l0=length(h); m0=length(g);
if(l0==0)
    l=f-lam'*g;%%%？ ？ ？ ？ ？ ？ ？ ？ ？ ？ ？ ？ ？ ？ ？ ？ ？ ？
end
if(m0==0)
    l=f-mu'*h;
end
if(l0>0&m0>0)
    l=f-mu'*h-lam'*g;
end
%%%%%%%%%拉格朗日函数的梯度%%%%%%%%
function dl=dlax(x,mu,lam)
df=df1(x);    %调用目标函数梯度文件
```

```
[Ae,Ai]=dcons(x);   %调用约束函数 Jacobi 矩阵文件
[m1,m2]=size(Ai); [l1,l2]=size(Ae);
if(l1==0)
   dl=df-Ai'*lam;
end
if(m1==0)
   dl=df-Ae'*mu;
end
if(l1>0&m1>0)
   dl=df-Ae'*mu-Ai'*lam;
end
```

%%

例 解非线性规划问题

```
   min f(x)=x1^2+x2^2-16x1-10x2
   s.t. -x1^2+6x1-4x2+11>=0
           x1x2-3x2-e^(x1-3)+1>=0
           x1>=0
           x2>=0
```

解 首先编写 4 个 m 文件

```
%%%%%%%%%目标函数 f(x)%%%%%%%%%
function f=f1(x)     %f1.m
f=x(1)^2+x(2)^2-16*x(1)-10*x(2);
%%%%%%%%%目标函数 f(x)的梯度%%%%%%%%%
function df=df1(x)   %df1.m
df=[2*x(1)-16;2*x(2)-10];
%%%%%%%%%约束函数%%%%%%%%%
function [h,g]=cons(x)   %cons.m
h=[];   %无等式约束
g=[-x(1)^2+6*x(1)-4*x(2)+11;...
     x(1)*x(2)-3*x(2)-exp(x(1)-3)+1;x(1);x(2)];   %不等式约束 []>=0 左边部分
%%%%%%%%%约束函数 Jacobi 矩阵%%%%%%%%%
function [dh,dg]=dcons(x)   %dcons.m
dh=[];
dg=[-2*x(1)+6,-4;x(2)-exp(x(1)-3),x(1)-3;1,0;0,1];
```

在 matlab 命令窗口依次输入下列命令:

```
x0=[4 4]';
mu0=[]';%等式个数
lam0=[0 0 0 0]';%不等式个数
[x,mu,lam,val,k]=sqpm(x0,mu0,lam0)
```