第五章作业

曾嘉翊
16307130203

1.用三种方法最小化Rosenbrock函数均能收敛到(1,1), 迭代步数如下.

	(-1.2,1)	(0,0)	(0.5,0.5)	(2,2)	(-1,-1)	平均
DFP	44	24	18	45	30	32.2
BFGS	22	20	16	42	20	24
SR1	57	35	31	15	40	35.6

由数值实验的结果, DFP和BFGS方法不跳过更新通常可以取得更快的迭代速度, 但是在一些参数下会收敛速度很慢(超过设置的最大迭代步数10000步). 在$y' * s$过小时跳过更新也不能完全避免这一情况，最后选择的措施是在$y' * s$过小时将H重置为单位阵. 例如DFP采取跳过更新H的方法以(-1,-1)作为初始点需要908次迭代，而重置H只需要30次迭代. 另外H的初值直接设为单位阵有时比$\frac{y's}{y'y}I$有更好的效果.

```
x0=[-1.2 1
    0 0
    0.5 0.5
    2 2
    -1 -1];
f=@Rosenbrock;
g=@Rosenbrockg;
eps=10^(-5);
for i=1:5
    [x,k]=DFP(x0(i,:)',eps,f,g);
    [x,k]=BFGS(x0(i,:)',eps,f,g);
    [x,k]=SR1(x0(i,:)',eps,f,g);
end
```

```
function y = Rosenbrock(x)
y = (1-x(1))^2+100*(x(2)-x(1)^2)^2;
end

function y = Rosenbrockg(x)
y = [-2*(1-x(1))-400*x(1)*(x(2)-x(1)^2)
    200*(x(2)-x(1)^2)];
end

function [x,k] = DFP(x,eps,f,g)
%DFP 拟牛顿法
%参数: 初始迭代点x, 精度eps, 目标函数f, 梯度g
n=length(x);%获取维数
gk=g(x);%初始化梯度
fk=f(x);%初始化函数值
k=0;%记录迭代次数
H=eye(n,n);%初始化H

% 迭代一步，初始化海森近似阵的逆H
% p=-gk;
% alpha=1;
% while f(x+alpha*p)>fk+10^(-4)*alpha*gk'*p
%     alpha=0.4*alpha;
% end
% s=alpha*p;
% y=g(x+s)-gk;
% H=y'*s/(y'*y)*I;

%开始迭代
while norm(gk)>eps
if k>=10000
    k='maxiterations';%收敛过慢则终止迭代
    break
end
```

```
p=−H*gk;
%步长规则armijo
alpha=1;
while f(x+alpha*p)>fk+10^(−4)*alpha*gk'*p
    alpha=0.42*alpha;
end
s=alpha*p;
x=x+s;
y=g(x)−gk;
 if  y'*s>10^(−10)  %y'* s 过小时重置H
    H=H−H*(y*y')*H/(y'*H*y)+s*s'/(y'*s);%更新H
else
    H=eye(n,n);
end
fk=f(x);
gk=g(x);
k=k+1;
end
end
```

```
function [x,k] = BFGS(x,eps,f,g)
%BFGS 拟牛顿法
%参数: 初始迭代点x, 精度eps, 目标函数f, 梯度g
n=length(x);%获取维数
gk=g(x);%初始化梯度
fk=f(x);%初始化函数值
k=0;%记录迭代次数
I=eye(n,n);
H=I;%初始化H
%开始迭代
while norm(gk)>eps
 if  k>=10000
     k='maxiterations';%收敛过慢则终止迭代
     break
end
p=-H*gk;
%步长规则armijo
alpha=1;
while f(x+alpha*p)>fk+10^(-4)*alpha*gk'*p
     alpha=0.42*alpha;
end
s=alpha*p;
x=x+s;
y=g(x)-gk;
 if  y'*s>10^(-10) %y'* s 过小时重置H
     rho=1/(y'*s);
     H=(I-rho*s*y')*H*(I-rho*y*s')+rho*(s*s');%更新H
else
     H=I;
end
fk=f(x);
gk=g(x);
k=k+1;
end
end
```

```
function [x,k] = SR1(x,eps,f,g)
%SR1 拟牛顿法
%参数: 初始迭代点x, 精度eps, 目标函数f, 梯度g
n=length(x);%获取维数
gk=g(x);%初始化梯度
delta=0.2; %初始半径
eta=0.01; %最小预测比
r=10^(-8);%判断跳过更新的系数
k=0;%记录迭代次数
B=eye(n,n);%初始化H

%开始迭代
while norm(gk)>eps
 if k>=10000
     k='maxiterations';%收敛过慢则终止迭代
     break
 end
 s=CGSteihaug(eps,delta,gk,B);
 y=g(x+s)-gk;
 rho=(f(x)-f(x+s))/(-gk'*s-1/2*s'*B*s);
 if rho>eta %更新x
     x=x+s;
 end
 gk=g(x);
 if rho>3/4 %更新半径
     if norm(s)>0.8*delta
         delta=2*delta;
     end
 elseif rho<0.1
     delta=delta/2;
 end
 if abs(s'*(y-B*s))>=r*norm(s)*norm(y-B*s) %更新B
 B=B+(y-B*s)*(y-B*s)'/((y-B*s)'*s);
 end
 k=k+1;
```

```
end
end

function p = CGSteihaug(eps,delta,r,B)
%CGSTEIHAUG 求解信赖域子问题
%参数: 精度eps, 半径delta, 梯度r, 海森近似阵B
n=length(r);
d=-r;
z=zeros(n,1);
if norm(r)<eps
    p=0;
end
for i=1:n
if d'*B*d<=0
    eq=[d'*d,2*z'*d,z'*z-delta^2];
    root=roots(eq);
    if root(1)>=0
        p=z+root(1)*d;
    else
        p=z+root(2)*d;
    end
    break
end
alpha=r'*r/(d'*B*d);
if norm(z+alpha*d)>=delta
    eq=[d'*d,2*z'*d,z'*z-delta^2];
    root=roots(eq);
    if root(1)>=0
        p=z+root(1)*d;
    else
        p=z+root(2)*d;
    end
    break
end
z=z+alpha*d;
```

```
beta=(r+alpha*B*d)'*(r+alpha*B*d)/(r'*r);
r=r+alpha*B*d;
if norm(r)<eps
    p=z;
    break
end
d=-r+beta*d;
end
end
```

2.用DFP和BFGS方法最小化Powell函数, 结果如下

DFP方法: $x = 10^{-9} * (0.8058, -0.0066, -0.5010, -0.0450)$, 迭代次数10

BFGS方法: $x = 10^{-8} * (0.0472, 0.0003, 0.1127, 0.0117)$, 迭代次数7

```
eps=10^(-5);
x0=[3 -1 0 1]';
f=@Powell;
g=@Powellg;
[x,k]=DFP(x0,eps,f,g);
[x,k]=BFGS(x0,eps,f,g);
function y = Powell(x)
y=(x(1)+10*x(2))^2+5*(x(3)-10*x(4))^2+(x(2)-2*x(3))^2+10*(x(1)-x(4))^2;
end
function y = Powellg(x)
y=[22*x(1)+20*x(2)-20*x(4)
   20*x(1)+202*x(2)-4*x(3)
   18*x(3)-100*x(4)-4*x(2)
   -100*x(3)+1020*x(4)-20*x(1)];
end
```