Matlab 第 10 次作业参考答案

第一题 $\min f(x)=3x^4+4x^3-12x^2+5$

（1） 对 $f(x)$ 在区间[-3,2]区间中找到最小值点，围绕该点建立一个长度为 1 的单峰区间，使用 Golden Section Search 求解极值问题，容许误差 0.0001，要求给出代码，运行后输出最优解和最优值。

（2） 分别使用 Newton Method、Quasi-Newton Method 求解上述极值问题，初始值取-1.5，$\Delta x=0.001$，容许误差 0.0001，给出代码，运行后输出最优解和最优值。

（3） 使用 Inaccurate Line Search Method 求解上述极值问题，初始值取-3，α_0 取 0.5，ε 取 0.2，容许误差 0.0001，给出代码，运行后输出最优解和最优值。

（4） 比较 Golden Section Search、Newton Method、Inaccurate Line Search Method 的迭代次数，并将三种方法的结果与 Matlab 自带函数的结果进行比较。

```
黄金分割搜索结果为:
最优解x*=-1.999987
最优值f*=-27.000000
迭代次数k=20

牛顿法结果为:
最优解x*=-2.000000
最优值f*=-27.000000
迭代次数k=5

拟牛顿搜索结果为:
最优解x*=-2.000000
最优值f*=-27.000000
迭代次数k=5

非精确线搜索结果为:
搜索方向d=1
最优解x*=-2.000000
最优值f*=-27.000000
迭代次数k=2
```

```
fminbnd结果为:
最优解x*=-2.000016
最优值f*=-27.000000
迭代次数k=10
```

比较：fminbnd 使用的是黄金分割和抛物线插值算法求解一维优化问题，结果与三种方法基本一致，迭代次数比自编黄金分割法少，但比牛顿法和非精确线搜索多。

(1)

Hw10_1.m

```
clc;clear;
f = @(x)3*x.^4+4*x.^3-12*x.^2+5;
figure(1)
```

```
fplot(f,[-3,2]) % [-2.5,-1.5]
Tol = 1e-4;
interval = [-2.5 -1.5];
[x_star,f_star,k] = GSS(f,interval,Tol);
fprintf('黄金分割搜索结果为:\n 最优解 x*=%f\n 最优值 f*=%f\n 迭代次数
k=%d\n',x_star,f_star,k);
```

GSS.m

```
function [x_star,f_star,k] = GSS(f,interval,Tol)
% interval is a initial unimodal interval
a = interval(1);
b = interval(2);
g = (sqrt(5)-1)/2;
k = 0;
while abs(b-a) >= Tol
    if f(a+(1-g)*(b-a)) < f(a+g*(b-a))
        b = a+g*(b-a);
    else
        a = a+(1-g)*(b-a);
    end
    k = k+1;
end
x_star = (a+b)/2;
f_star = f(x_star);
end
```

(2-1)

```
f = @(x)3*x.^4+4*x.^3-12*x.^2+5;
df = @(x)12*x.^3+12*x.^2-24*x;
ddf = @(x)36*x.^2+24*x-24;
Tol = 1e-4;
x0 = -1.5;
[x_star,f_star,k] = NewtonSearch(f,df,ddf,x0,Tol);
fprintf('牛顿法结果为:\n 最优解 x*=%f\n 最优值 f*=%f\n 迭代次数 k=%d\n',x_star,f_star,k)
```

NewtonSearch.m

```
function [x_star,f_star,k] = NewtonSearch(f,df,ddf,x0,Tol)
x1 = x0;
k = 0;
while abs(df(x1)) >= Tol
    x0 = x1;
    x1 = x0-df(x0)/ddf(x0);
    k = k+1;
end
```

```
x_star = x1;
f_star = f(x_star);
end
```

(2-2)

```
f = @(x)3*x.^4+4*x.^3-12*x.^2+5;
df = @(x)12*x.^3+12*x.^2-24*x;
% ddf = @(x)36*x.^2+24*x-24;
Tol = 1e-4;
x0 = -1.5;
dx = 1e-3;
[x_star,f_star,k] = QuasiNtSearch(f,df,dx,x0,Tol);
fprintf('拟牛顿搜索结果为:\n 最优解 x*=%f\n 最优值 f*=%f\n 迭代次数
k=%d\n',x_star,f_star,k)
```

QuasiNtSearch.m

```
function [x_star,f_star,k] = QuasiNtSearch(f,df,dx,x0,Tol)
x1 = x0;
k = 0;
while abs(df(x1)) >= Tol
    x0 = x1;
    x1 = x0-dx*(f(x0+dx)-f(x0-dx))/(2*(f(x0+dx)-2*f(x0)+f(x0-dx)));
    k = k+1;
end
x_star = x1;
f_star = f(x_star);
end
```

(3)

```
clc;clear;
f = @(x)3*x.^4+4*x.^3-12*x.^2+5;
df = @(x)12*x.^3+12*x.^2-24*x;
ddf = @(x)36*x.^2+24*x-24;
x0 = -3;
alp0 = 0.5;
epsilon = 0.2;
Tol = 1e-4;
k = 0;
x1 = x0;
d = 1;
while abs(df(x1)*d) > Tol
    x0 = x1;
%       d = -df(x0)/ddf(x0);
    alp1 = lineSearch(f,df,x0,alp0,epsilon,d);
```

```
        alp0 = alp1;
        x1 = x0+alp1*d;
        k = k+1;
end
x_star = x1;
f_star = f(x_star);
fprintf('非精确线搜索结果为:\n 搜索方向 d=%d\n 最优解 x*=%f\n 最优值 f*=%f\n 迭代次数
k=%d\n',d,x_star,f_star,k)
```

lineSearch.m

```
function alp_star = lineSearch(f,gf,x0,alp0,epsilon,d)
% 搜索当前步的合适步长，AG 准则
while 1
        if f(x0+alp0*d) <= f(x0)+epsilon*alp0*gf(x0)'*d
                alp_star = alp0;
                break;
        else
                alp0 = -gf(x0)'*d/(2*(f(x0+d)-f(x0)-gf(x0)'*d)); % 二次插值线搜索
        end
end
end
```

（4）

```
clc;clear;
f = @(x)3*x.^4+4*x.^3-12*x.^2+5;
[x,fval,exitflag,output] = fminbnd(f,-3,2);
fprintf('fminbnd 结果为:\n 最优解 x*=%f\n 最优值 f*=%f\n 迭代次数
k=%d\n',x,fval,output.iterations);
```

第二题 $\min f(x,y)=100\left(y-x^2\right)^2+(x-1)^2$

（1） 分别使用 Steepest Descent Method、Newton Method 求解极值问题，初值[2;2]，ε=0.2，容许误差 0.0001，给出代码，运行后输出最优解和最优值。

（2） 使用 Quasi-Newton Method 求解上述极值问题，初值[2;2]，H_0为单位阵，容许误差 0.0001，给出代码，运行后输出最优解和最优值，并将结果与 matlab 自带函数的结果进行比较。

（3） 作出目标函数的曲面图，并绘出 Steepest Descent Method、Newton Method、Quasi-Newton Method 的迭代历史。

（1）

```
最速下降法结果为:
最优解为x*=[1.0001115791,1.0002236287]
最优值为f*=0.0000000125
```

```
牛顿法结果为:
最优解为x*=[1.0001112648,1.0002229823]
最优值为f*=0.0000000124
```

（2）

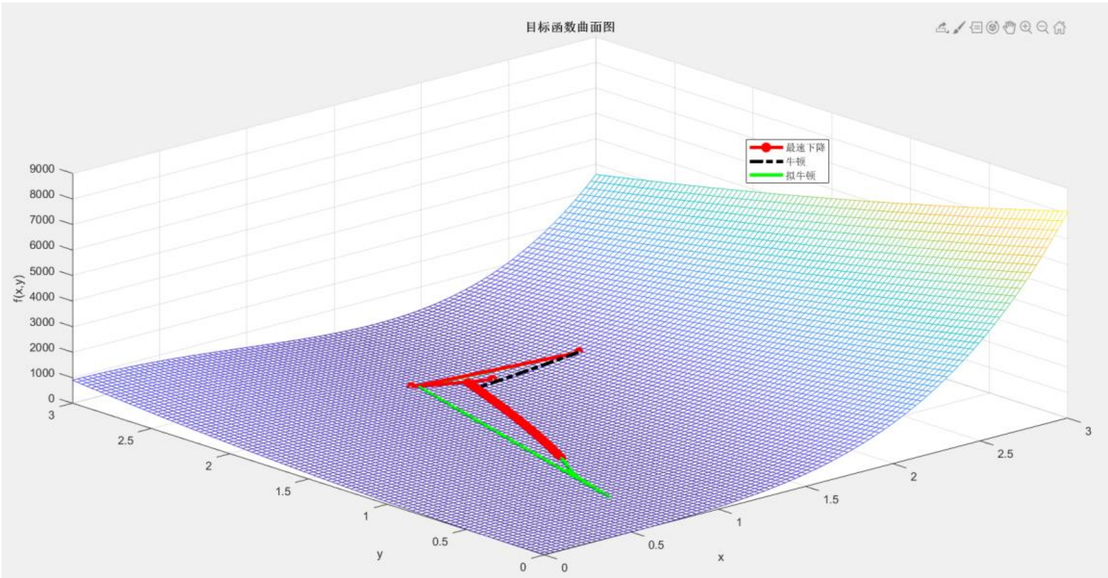
```
拟牛顿法结果为:
最优解为x*=[1.0000108525,1.0000217610]
最优值为f*=0.0000000001
```

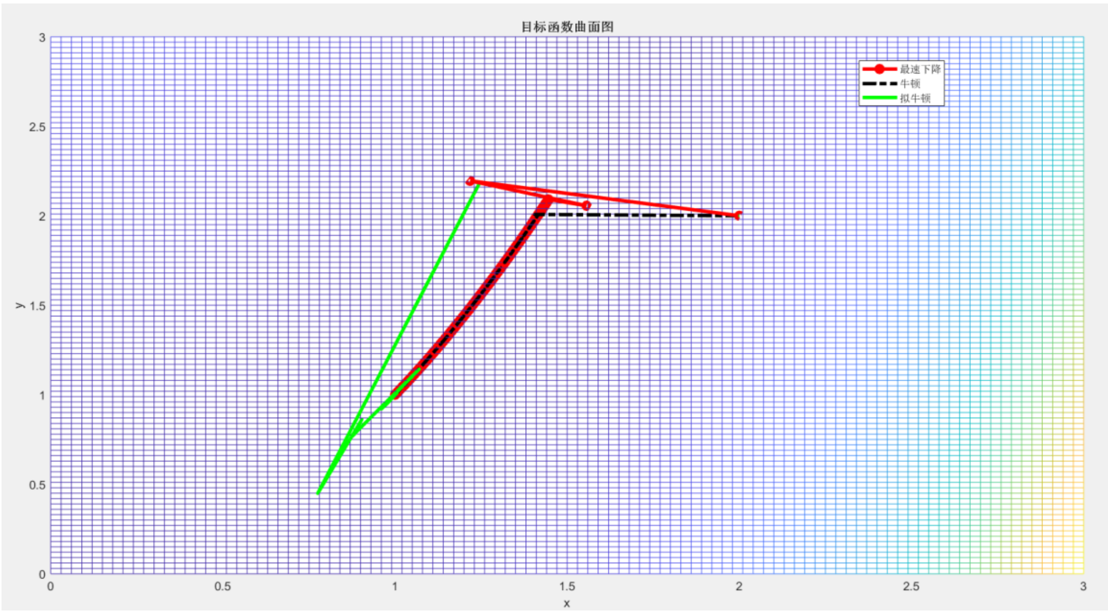
```
fminunc结果为:
最优解为x*=[0.9999979533,0.9999959730]
最优值为f*=0.0000000000
```

比较：

Fminunc 使用的 quasi-newton 法，结果基本相同，但自编的拟牛顿法迭代步数比 fminunc 多。

（3）





clc;clear;

f = @(x)100*(x(2)-x(1)^2)^2+(x(1)-1)^2;

gf = @(x)[-400*x(1)*(x(2)-x(1)^2)+2*(x(1)-1);200*(x(2)-x(1)^2)];

Hf = @(x)[-4*x(2)+12*x(1)^2+2, -4*x(1);-4*x(1), 200];

```
% 最速下降法
x0 = [2;2];
x1 = x0;
f1 = f(x0);
epslion = 0.2;
tol = 1e-4;
w = 0.5;
iter = 1;
while norm(gf(x0))>tol
    alpha = 1;
    dk = -gf(x0);
    while 1 % 线搜索
        if f(x0+alpha*dk)<=f(x0)+epslion*alpha*gf(x0)'*dk
            x0 = x0+alpha*dk;
            break;
        else
            alpha = w*alpha;
        end
    end
    iter = iter+1;
    x1(:,iter) = x0;
    f1(:,iter) = f(x0);
end
iter1 = iter;
fprintf('最速下降法结果为:\n')
fprintf('最优解为 x*=[%.10f,%.10f]\n',x1(1,end),x1(2,end))
fprintf('最优值为 f*=%.10f\n',f1(end))

% 牛顿法
x0 = [2;2];
x2 = x0;
f2 = f(x0);
iter = 1;
while norm(gf(x0))>tol
    alpha = 1;
    dk = -(Hf(x0))\gf(x0);
    while 1
    if f(x0+alpha*dk)<=f(x0)+epslion*alpha*gf(x0)'*dk
        x0 = x0+alpha*dk;
        break;
    else
        alpha = w*alpha;
    end
```

```
        end
        iter = iter+1;
        x2(:,iter) = x0;
        f2(:,iter) = f(x0);
end
iter2 = iter;
fprintf('牛顿法结果为:\n')
fprintf('最优解为 x*=[%.10f,%.10f]\n',x2(1,end),x2(2,end))
fprintf('最优值为 f*=%.10f\n',f2(end))

%  拟牛顿法
x0 = [2;2];
epslion = 0.2;
tol = 1e-4;
w = 0.6;
H = eye(2);
x3 = x0;
f3 = f(x0);
iter = 1;
while norm(gf(x0))>tol
        alpha = 1;
        dk = -H*gf(x0);
        while 1
        if f(x0+alpha*dk)<=f(x0)+epslion*alpha*gf(x0)'*dk
                break;
        else
                alpha = w*alpha;
        end
        end
        sk = alpha*dk;
        yk = gf(x0+sk)-gf(x0);
        x0 = x0 + sk;
        H = H + sk*sk'/(sk'*yk) - H*yk*yk'*H/(yk'*H*yk);
        iter = iter+1;
        x3(:,iter) = x0;
        f3(:,iter) = f(x0);
end
iter3 = iter;
fprintf('拟牛顿法结果为:\n')
fprintf('最优解为 x*=[%.10f,%.10f]\n',x3(1,end),x3(2,end))
fprintf('最优值为 f*=%.10f\n',f3(end))

% fminunc
options = optimoptions('fminunc','OptimalityTolerance',1e-6)
```

```
[xfm,fval,exitflag,output] = fminunc(f,[2;2],options);
fprintf('fminunc 结果为:\n')
fprintf('最优解为 x*=[%.10f,%.10f]\n',xfm(1,end),xfm(2,end))
fprintf('最优值为 f*=%.10f\n',fval(end))

%%  绘图
figure(1);
x = linspace(0,3,101);
y = linspace(0,3,101);
[X Y] = meshgrid(x,y);
Z = 100*(Y-X.^2).^2+(X-1).^2;
fg1 = mesh(X,Y,Z);
xlabel('x');
ylabel('y');
zlabel('f(x,y)');
title('目标函数曲面图');
hold on

figure(1)
fg2 = plot3(x1(1,:),x1(2,:),f1,'-or','LineWidth',3);
fg3 = plot3(x2(1,:),x2(2,:),f2,'-.k','LineWidth',3);
fg4 = plot3(x3(1,:),x3(2,:),f3,'-g','LineWidth',3);
legend([fg2 fg3 fg4],{'最速下降' '牛顿' '拟牛顿'})
```

第三题

$$\begin{aligned} \max \quad & -3x_1 + 2x_2 - x_3 \\ \text{s.t.} \quad & 2x_1 + x_2 - x_3 \leqslant 5 \\ & 4x_1 + 3x_2 + x_3 \geqslant 3 \\ & -x_1 + x_2 + x_3 = 2 \\ & x_1, x_2, x_3 \geqslant 0 \end{aligned}$$

（1） 编程实现单纯形法，求解上述问题。要求给出代码，运行后输出结果。

（2） 使用 linprog 函数求解上述问题。要求给出代码，运行后输出结果。

（1）

```
LP结果为:
最优解为x*=[0.0000000000,2.0000000000,0.0000000000]
最优值为f_max=4.0000000000
```

Hw10_3_1.m

```
fs = [3 -2 1 0 0];
Aeqs = [2 1 -1 1 0;-4 -3 -1 0 1;-1 1 1 0 0];
beqs = [5;-3;2];
xs = LP(fs,Aeqs,beqs);
fvals = fs*xs;
fprintf('LP 结果为:\n')
```

```
fprintf('最优解为 x*=[%.10f,%.10f,%.10f]\n',xs(1),x(2),x(3))
fprintf('最优值为 f_max=%.10f\n',-fvals)
```

LP.m

```
function [x]=LP(f,A,b)
n=length(b);%the number of equations
m=size(A,2);%the number of variables

s=combntns(1:m,n);
mark=0;
for i=1:size(s,2)
    x=A(:,s(i,:));
    x=rref(x);
    if ~isempty(find(x(n,:)~=0))
        ColB=s(i,:);% the index of cols of B
        B=A(:,ColB);% array B
        [Atemp,btemp]=BasicEye(A,b,ColB);
        if isempty(find(btemp<0))
            mark=1;
            break
        end
    end
end
if mark==0
    error('There is no solution.')
end
A=Atemp;
b=btemp;
x=LPf(ColB,B,n,m,f,A,b);
end

function [x]=LPf(ColB,B,n,m,f,A,b)

for i=1:n
    index=find(A(:,ColB(i))~=0);
    for j=1:length(index)
        if index(j)>=i
            index1=index(j);
            break
        end
    end
    b(index1)=b(index1)/A(index1,ColB(i));
    A(index1,:)=A(index1,:)/A(index1,ColB(i));
    for j=1:n
```

```
            if j~=index1
                if A(j,ColB(i))~=0
                    b(j)=b(j)-b(index1)*A(j,ColB(i));
                    A(j,:)=A(j,:)-A(index1,:)*A(j,ColB(i));
                end
            end
        end
        if index1~=i
            tempA=A(i,:);
            tempb=b(i);
            A(i,:)=A(index1,:);
            b(i)=b(index1);
            A(index1,:)=tempA;
            b(index1)=tempb;
        end
    end
    XB=zeros(m,1);
    XB(ColB)=b;
    r=[];% nonbasis r
    rindex=[];% nonbasis index
    for i=1:m
        if isempty(find(ColB==i))
            rindex=[rindex i];
        end
    end
    for i=1:m-n
        intemp=rindex(i);
        temp=f(intemp)-f(ColB)*A(:,intemp);
        r=[r temp];
    end
    if isempty(find(r<0))
        x=XB;
    else
        [~,in]=min(r);
        in=rindex(in(1));%basis in
        [bA,order]=sort(b./A(:,in));
        out=[];
        for j=1:n
            if bA(j)>0&bA(j)~=Inf&bA(j)~=-Inf
                out=order(j);
                break
            end
        end
        if isempty(out)
```

```
            error('bA<0');
        end
        ColB(out)=in;
        B=A(:,ColB);
        x=LPf(ColB,B,n,m,f,A,b);
end
end

function [A,b]=BasicEye(A,b,ColB)
n=length(b);
for i=1:n
        index=find(A(:,ColB(i))~=0);
        for j=1:length(index)
            if index(j)>=i
                index1=index(j);
                break
            end
        end
        b(index1)=b(index1)/A(index1,ColB(i));
        A(index1,:)=A(index1,:)/A(index1,ColB(i));
        for j=1:n
            if j~=index1
                if A(j,ColB(i))~=0
                    b(j)=b(j)-b(index1)*A(j,ColB(i));
                    A(j,:)=A(j,:)-A(index1,:)*A(j,ColB(i));
                end
            end
        end
        if index1~=i
            tempA=A(i,:);
            tempb=b(i);
            A(i,:)=A(index1,:);
            b(i)=b(index1);
            A(index1,:)=tempA;
            b(index1)=tempb;
        end
end
end
```

（2）

```
linprog结果为:
最优解为x*=[0.0000000000,2.0000000000,0.0000000000]
最优值为f_max=4.0000000000
```

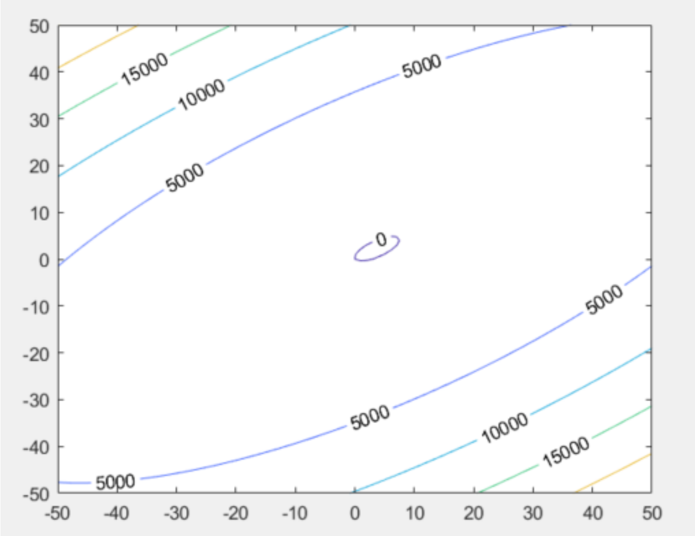
Hw10_3_2.m

```
clc;clear;
f=[3 -2 1];
A=[2 1 -1; -4 -3 -1];
b=[5;-3];
Aeq=[-1 1 1];
beq=[2];
LB=[0 0 0];
LU=[Inf Inf Inf];
[x,fval,exitflag,output] =linprog(f,A,b,Aeq,beq,LB,LU);
fprintf('linprog 结果为:\n')
fprintf('最优解为 x*=[%.10f,%.10f,%.10f]\n',x(1),x(2),x(3))
fprintf('最优值为 f_max=%.10f\n',-fval)
```

第四题

$$\begin{aligned} \min\ & 2x_1^2 - 4x_1x_2 + 4x_2^2 - 6x_1 - 3x_2 \\ \text{s.t.}\ & x_1 + x_2 \le 3 \\ & 4x_1 + x_2 \le 9 \\ & x_1, x_2 \ge 0 \end{aligned}$$

使用不同的初值，调用 Matlab 自带的 fmincon 和 ga 函数求解上述非线性规划问题并比较之。




```
初值为:x0=[0.0000000000,0.0000000000]
fmincon结果为:
最优解为x*=[1.9499999715,1.0500000202]
最优值为f*=-11.0249999801
```

ga结果为:
最优解为x*=[1.9490359086,1.0519640150]
最优值为f*=-11.0273749542

比较：Fmincon 依赖于初值的选择，初值选择不好，则容易陷入局部最优；而遗传算法有概率能够达到全局最优。

Hw10_4.m

```
clc;clear;
f1 = @(x)2*x(1).^2-4*x(1)*x(2)+4*x(2).^2-6*x(1)-3*x(2);
f2 = @(x,y)2*x.^2-4*x.*y+4*y.^2-6*x-3*y;
x1 = linspace(-50,50,101);
x2 = linspace(-50,50,101);
[X1,X2] = meshgrid(x1,x2);
Z = f2(X1,X2);
figure
contour(X1,X2,Z,'ShowText','on')

clc;
LB = [0;0];
UB = [Inf;Inf];
A = [1 1;4 1];
b = [3;9];
x0 = [0;0];
[x,fval,exitflag,output] = fmincon(f1,x0,A,b,[],[],LB,UB);
fprintf('初值为:x0=[%.10f,%.10f]\n',x0(1),x0(2))
fprintf('fmincon 结果为:\n')
fprintf('最优解为 x*=[%.10f,%.10f]\n',x(1),x(2));
fprintf('最优值为 f*=%.10f\n',fval)
%
[xg,fvalg,exitflagg,outputg] = ga(f1,2,A,b,[],[],LB,UB);
fprintf('ga 结果为:\n')
fprintf('最优解为 x*=[%.10f,%.10f]\n',xg(1),xg(2))
fprintf('最优值为 f*=%.10f\n',fvalg)
```