第一次课后练习

2018302120169 傅宇千

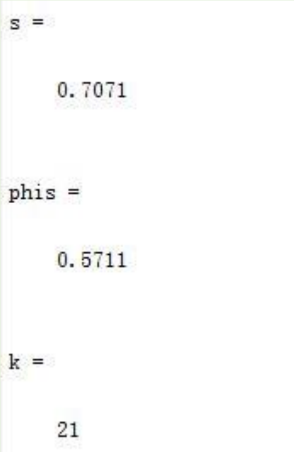
本次实验采用 MATLAB 编程实现

1. 编程实现用黄金分割法求函数$\phi(\alpha)=1-\alpha e^{-\alpha^2}$的极小值，取初始区间为[0,1]， ε =0.01.

   该程序如图：

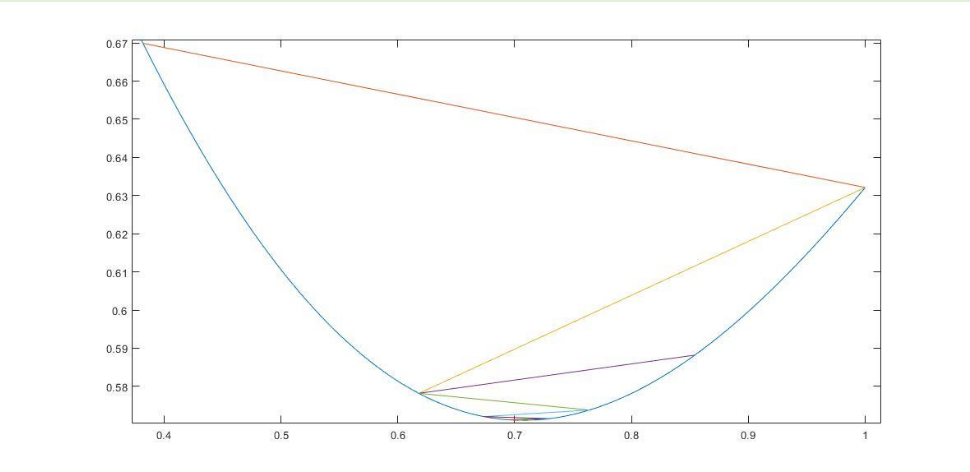
```
function [s,phis,k,G,E]=golds(phi,a,b,delta,epsilon)
%输入: phi是目标函数, a, b 是搜索区间的两个端点
% delta, epsilon分别是自变量和函数值的容许误差
% 输出: s, phis分别是近似极小点和极小值, G是n x 4矩阵,
% 其第k行分别是a,p,q,b的第k次迭代值[ak,pk,qk,bk],
% E=[ds,dphi], 分别是s和phis的误差限.
x=0:0.0001:1;
y=feval(phi,x);
plot(x,y);
hold on;
t=(sqrt(5)-1)/2;%缩短率
h=b-a;%初始区间长度
phia=feval(phi,a); phib=feval(phi,b);%计算此时两端点函数值

p=a+(1-t)*h; q=a+t*h;
phip=feval(phi,p); phiq=feval(phi,q);

k=1;
G(k,:)=[a, p, q, b];

while(abs(phib-phia)>epsilon)||(h>delta)
    if(phip<phiq)
        b=q; phib=phiq; q=p; phiq=phip;
        h=b-a; p=a+(1-t)*h; phip=feval(phi,p);
    else
        a=p; phia=phip; p=q; phip=phiq;
        h=b-a; q=a+t*h; phiq=feval(phi,q);
    end
    k=k+1;
    G(k,:)=[a, p, q, b];
    %text(a,phia,'o');
    %text(b,phib,'o');
    plot([a;b],[phia,phib]);
    hold on;
end
ds=abs(b-a); dphi=abs(phib-phia);
if(phip<=phiq)
    s=p; phis=phip;
else
    s=q; phis=phiq;
end
text(s,phis,'*','color','g');
E=[ds,dphi];
```

运行截图：



```
s =

    0.7071


phis =

    0.5711


k =

    21
```

可视化结果：



2. 编程实现用 Armijio 搜索算法计算第 k 次的搜索步长 αk，目标函数为：

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

给定：$x_k = (1, -1)^T, d_k = (1,1)^T$，$\rho = 0.6$。

该程序如下图：

```
% 目标函数
function f=fun(x)
f=100*(x(1)^2-x(2))^2+(x(1)-1)^2;
% 目标函数梯度
function gf=gfun(x)
gf=[400*x(1)*(x(1)^2-x(2))+2*(x(1)-1), -200*(x(1)^2-x(2))]';
function Armijio(xk,dk)
rho=0.6;%题目中rho=0.6
beta=0.5;%取beta为0.5
m=0; mmax=70;%实验证明只需69次即可收敛
if(fun(xk+dk)<=fun(xk)+rho*gfun(xk)'*dk)
    alpha=1;
else
    while (m<=mmax)
        if(fun(xk+beta*rho^m*dk)<=fun(xk)+beta*rho^(m+1)*gfun(xk)'*dk)
            mk=m;
            break;
        end
        m=m+1;
    end
    alpha=beta*rho^mk;
end

alpha
beta
xk1=xk+alpha*dk
fk=fun(xk)
fk1=fun(xk1)
```

在运行过程中，我发现在题中由于题设梯度与实际所需下降梯度差距过大，所以计算出 alpha 很小，将梯度改为(1，3)'后即可得出较为合适的结果。

下面贴出原题的梯度方向运行结果和改变题设梯度方向后的运行结果：

```
>> xk=[1,-1]'; dk=[1,1]'; Armijio(xk,dk)

alpha =

   2.4627e-16


beta =

    0.5000


xk1 =

    1.0000
   -1.0000


fk =

   400


fk1 =

   400
```

原题设下运行结果

```
>> xk=[1,-1]'; dk=[1,3]'; Armijio(xk,dk)

alpha =

    0.3000


beta =

    0.5000


xk1 =

    1.3000
   -0.1000


fk =

   400


fk1 =

  320.5000
```

改变题设后运行结果