

成绩	
----	--

中国矿业大学数学学院

实验报告

课程名称：实用优化算法

实验名称：实验二：共轭梯度法

姓名学号：胡钧耀 06192081

实验时间：2021 年 10 月 24 日

《实用优化算法》实验报告

实验名称：实验二（共轭梯度算法）

1 实验目的

复习理论课学习的共轭梯度算法，运用实验一的一维搜索方法——黄金分割法，掌握 MATLAB 编程实现共轭梯度算法，培养 MATLAB 编程与上机调试能力，加强计算机与数学学科联系。

2 实验内容

$$\min f_1(x_1, x_2) = (1 - x_1)^2 + 2(x_2 - x_1^2)^2, x_0 = (0, 0)^T \quad (1)$$

$$\min f_2(x_1, x_2) = \frac{x_1^4}{4} + \frac{x_2^2}{2} - x_1x_2 + x_1 - x_2, x_0 = (0, 0)^T, x_0' = (1.5, 1)^T \quad (2)$$

3 算法设计

设计如下算法程序结构：

Conjugate_Gradient.m 实现共轭梯度算法。函数输入为：需要求解的函数，需要求解的函数的梯度，起始点。函数输出为：该函数最小值对应自变量值，该函数最小值，迭代次数。golden_section.m 是根据实验 1 的黄金分割算法进行改进后的算法，便于计算出每一次一维搜索的步长。

f1.m 保存问题 1 函数，grad_f1.m 保存问题 1 函数的梯度，f2.m 保存问题 2 函数，grad_f2.m 保存问题 2 函数的梯度，p.m 保存一维搜索的搜索步骤。根据参考文献学习[1]，可设置将函数以及对应函数梯度作为参数直接放入算法函数进行计算。

4 程序代码

Conjugate_Gradient.m

```
function
[x_star, val, k] = Conjugate_Gradient(fun, grad_fun, x_old)
epsilon = 1e-5; n = length(x_old); beta = 0; d = 0; k = 0;
f = feval(fun, x_old); g = feval(grad_fun, x_old);
if norm(g) <= epsilon
    run = 0;
else
    run = 1;
end
while run == 1 && k <= 10000
    d = -g + beta*d;
    alpha = golden_section(fun, x_old, d);
    x_new = x_old + alpha*d;
    g = feval(grad_fun, x_new);
    if norm(g) <= epsilon
        run = 0;
    else
```

```

        if mod(k,n+1) == 0
            beta = 0;
        else
            beta =
                norm(feval(grad_fun,x_new))^2/
                norm(feval(grad_fun,x_old))^2;
        end
        k = k + 1;
        x_old = x_new;
    end
end
x_star = x_old;
val=feval(fun,x_star);

```

golden_section.m

```

function final = golden_section(func, x, d)
epsilon = 1e-5; a = 0; b = 2;
alp1 = a + 0.382*(b-a); alp2 = a + 0.618*(b-a);
while abs(b-a) > epsilon
    p1 = p(func, alp1,x,d);
    p2 = p(func, alp2,x,d);
    if p1 >= p2
        a = alp1;
        alp1 = alp2;
        alp2 = a + 0.618*(b-a);
    else
        b = alp2;
        alp2 = alp1;
        alp1 = a + 0.382*(b-a);
    end
end
final = (a+b) / 2;

```

f1.m

```

function f= f1(x)
f = (1-x(1))^2+2*(x(2)-x(1)^2)^2;
end

```

grad_f1.m

```
function gf = grad_f1(x)
gf = [2*x(1) - 8*x(1)*(- x(1)^2 + x(2)) - 2;
      - 4*x(1)^2 + 4*x(2)];
end
```

f2.m

```
function f= f2(x)
f = 0.25*x(1)^4+0.5*x(2)^2-x(1)*x(2)+x(1)-x(2);
end
```

grad_f2.m

```
function gf = grad_f2(x)
gf = [x(1)^3 - x(2) + 1;
      x(2) - x(1) - 1];
end
```

5 运行结果

命令行输入

```
>> fun = @f1; grad_fun = @grad_f1;
>> [x,y,t]=Conjugate_Gradient(fun,grad_fun,[0;0])
x =    1.0000
      1.0000
y =    6.4380e-11
t =    197

>> fun = @f2; grad_fun = @grad_f2;
>> [x,y,t]=Conjugate_Gradient(fun,grad_fun,[0;0])
x =   -1.0000
      -0.0000
y =   -0.7500
t =    23

>> [x,y,t]=Conjugate_Gradient(fun,grad_fun,[1.5;1])
x =    1.0000
      2.0000
y =   -0.7500
t =    23
```

6 结果分析

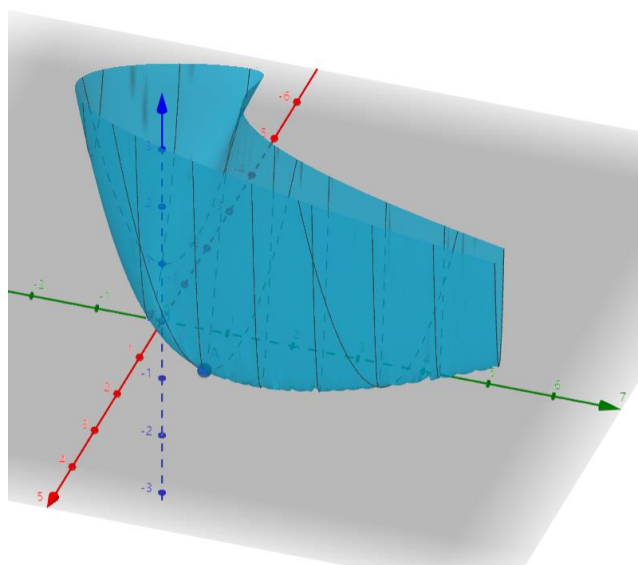


图 1 GEOGEBREA 绘图验证结果（问题一）

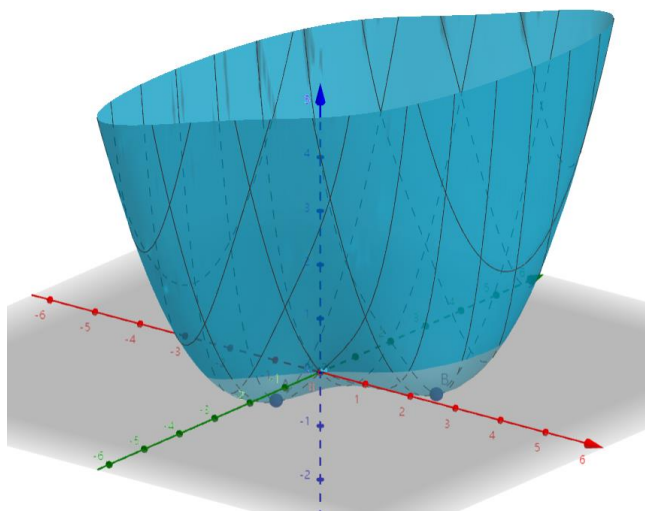


图 2 GEOGEBREA 绘图验证结果（问题二）

对计算结果进行分析，与绘图观察直接得到的结果十分相近。对于问题二，全局有两个极小值，因此需要选取其中的最小值作为全局极小值，最后求解得到两个数值相等。

$$\min f_1 = f_1(1, 1) = 0 \quad (3)$$

$$\min f_2 = \min\{f_2(-1, 0), f_2(1, 2)\} = \min\{-0.75, -0.75\} = -0.75 \quad (4)$$

7 参考文献

- [1] Matlab 中如何将（自定义）函数作为参数传递给另一个函数.VVingerfly.博客园.<https://www.cnblogs.com/VVingerfly/p/4793131.html>