



西安交通大学
XI'AN JIAOTONG UNIVERSITY

最优化方法上机报告

信计 91 闻逊之

学号:2193410365

2022 年 6 月 8 日

(2021-2022 春季学期)

— 作业 4 第一题

1.1 题目阐述

分别用 0.618 法和三点二次插值法求 $\phi(\alpha) = 1 - \alpha e^{-\alpha^2}$ 的极小点, 初始区间取 $[0, 1]$, $\epsilon = 0.001$.

1.2 绘制函数图像

$\phi(\alpha) = 1 - \alpha e^{-\alpha^2}$ 的函数图像如图1所示, 可见函数呈高-低-高形状, 在目标区间内是单峰函数, 满足 0.618 法以及三点二次插值法的应用条件.

容易算出函数的理论极小点为 $\frac{\sqrt{2}}{2}$.

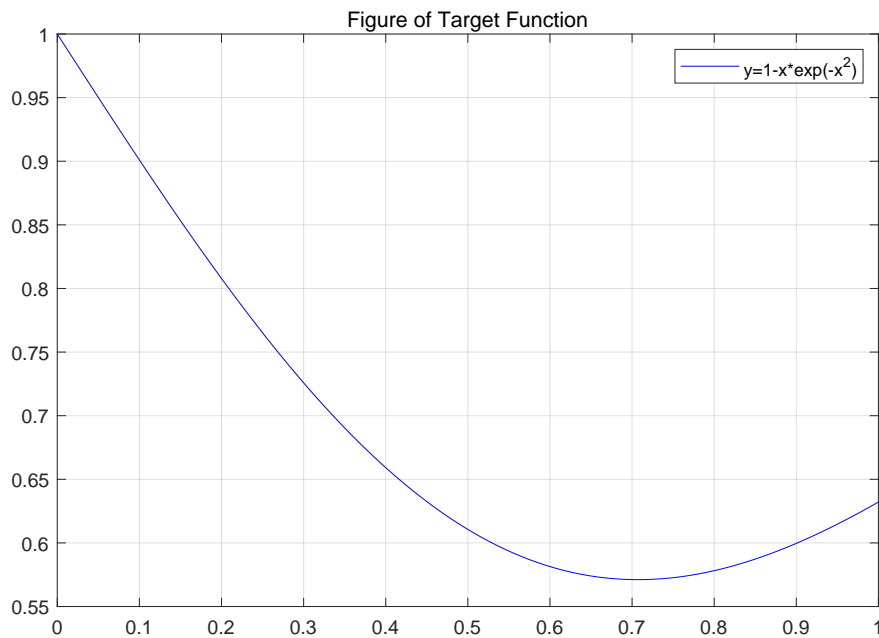


图 1: 目标函数图像

1.3 0.618 法结果及程序

0.618 法求得的结果为 $\alpha^* = 0.7069965$, 其迭代次数与误差距离的图像如图2 所示.

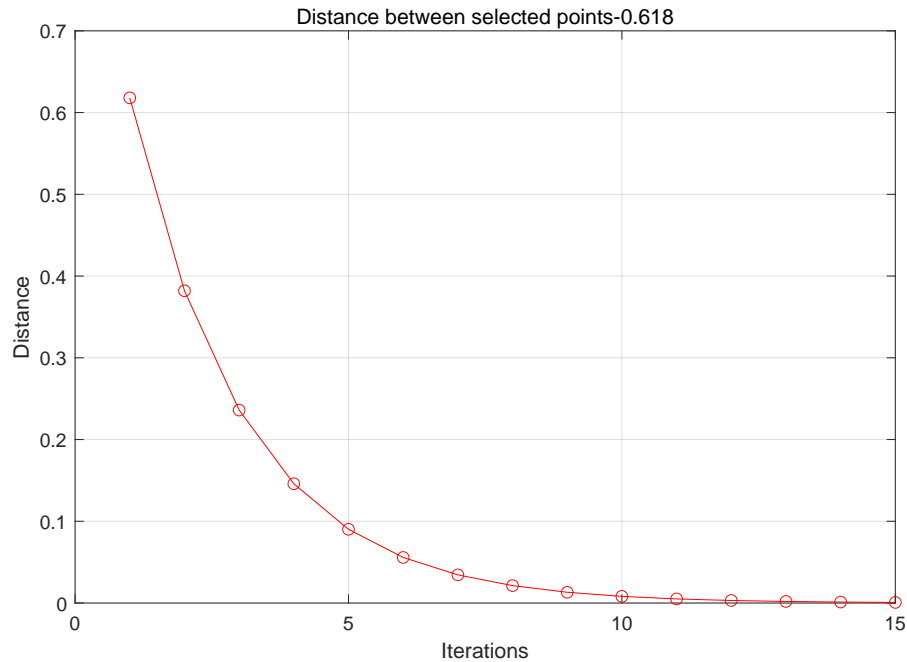


图 2: 0.618 法误差收敛图像

0.618 法 matlab 代码

```
1 function [output] = func_618(f,a0,b0,epsilon)
2 %用0.618法计算函数极小值的函数
3 % 提供函数形式、区间左右端点以及误差
4 lambda = [];mu = [];a = [];b=[];
5 a=[a,a0];b=[b,b0];k=1;
6 lambda = [lambda,a0+0.382*(b0-a0)];mu(end+1) = a0+0.618*(b0-a0);
7 %下一行为画图考. 虑Sign用以标记输出哪个误差
8 kline=zeros(1,100,'int8');error = zeros(1,100,'double');Sign = 0;
9 while(b(k)-lambda(k)>epsilon&&mu(k)-a(k)>epsilon)
10     if (f(lambda(k))>f(mu(k)))
11         Sign = 1;
12         if (b(k)-lambda(k)<=epsilon)
13             output = mu(k);
14             break;
```

```

15     else
16         a(k+1)=lambda(k);b(k+1) = b(k);lambda(k+1) = mu(k);
17         mu(k+1) = a(k+1)+0.618*(b(k+1)-a(k+1));
18     end
19     else
20         Sign = 2;
21         if (mu(k)-a(k)<=epsilon)
22             output = lambda(k);
23             break;
24         else
25             a(k+1) = a(k);b(k+1) = mu(k);mu(k+1) = lambda(k);
26             lambda(k+1) = a(k+1)+0.382*(b(k+1)-a(k+1));
27         end
28     end
29     kline(k) = k;
30     if (Sign==1)
31         error(k) = b(k)-lambda(k);
32     elseif (Sign==2)
33         error(k) = mu(k)-a(k);
34     end
35     k=k+1;
36 end
37 %输出结论
38 if (b(k)-lambda(k)<=epsilon)
39     output = mu(k);
40 elseif (mu(k)-a(k)<=epsilon)
41     output = lambda(k);
42 end
43 %加入最后一次迭代的信息
44 kline(k) = k;
45 if (Sign==1)
46     error(k) = b(k)-lambda(k);
47 elseif (Sign==2)
48     error(k) = mu(k)-a(k);
49 end
50 %删除0值
51 kline(kline==0)=[];
52 error(error==0)=[];
53 %绘图
54 plot(kline,error,'-or');

```

```

55 grid on;
56 title ("Distance between selected points—0.618");
57 xlabel(' Iterations ');
58 ylabel(' Distance ');

```

1.4 三点二次插值法结果及程序

三点二次插值法求得的结果为 $\alpha^* = 0.69559476$, 其迭代次数与误差距离额图像如图3所示

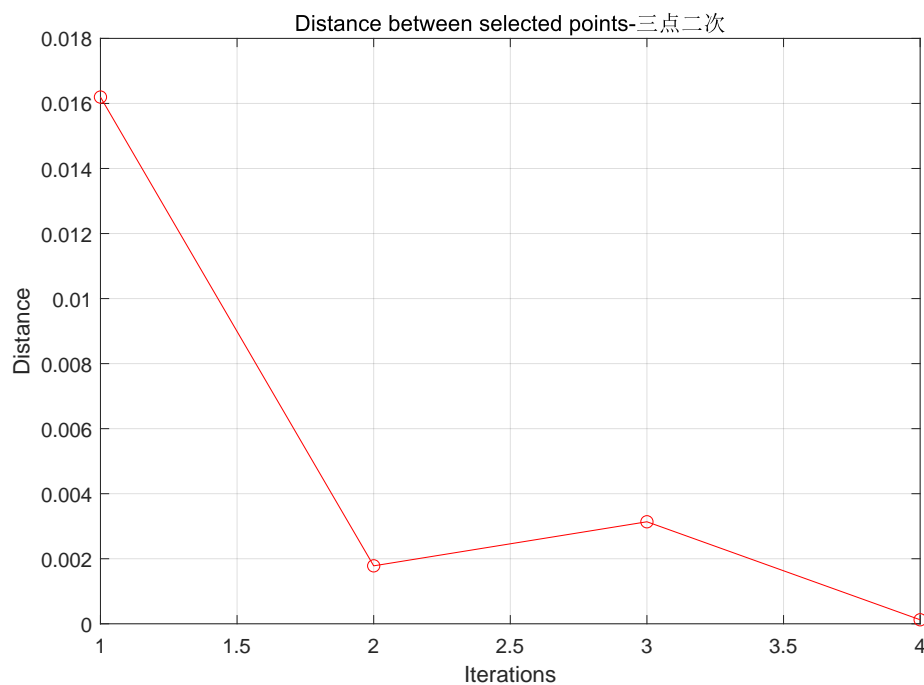


图 3: 三点二次插值法误差收敛图像

三点二次插值法 matlab 代码

```

1 function [output] = func_3_2(f,a0,b0,epsilon)
2 %使用三点二次插值函数求函数极小值点
3 % 输入函数形式、左右端点以及要求误差
4 %(默认函数在此区间内任意点函数值小于在端点的函数值 )
5 x1=a0;x2=(a0+b0)/2;x3=b0;f1=f(x1);f2=f(x2);f3=f(x3);
6 x=0.5*((x2^2-x3^2)*f1+(x3^2-x1^2)*f2+(x1^2-x2^2)*f3)/((x2-x3)*f1+ ...
7 (x3-x1)*f2+(x1-x2)*f3);

```

```

8      %以下为画图考虑
9      k=1;kline=zeros(1,100,'int8'); error = zeros(1,100,'double');
10     while(abs(f(x2)-f(x))>=epsilon*f(x2))
11         fx=f(x);f1=f(x1);f2=f(x2);f3=f(x3);
12         if (x>x2)
13             if (fx<=f2)
14                 x1=x2;x2=x;
15             else
16                 x3=x;
17             end
18         else
19             if (fx<=f2)
20                 x3=x2;x2=x;
21             else
22                 x1=x;
23             end
24         end
25         x=0.5*((x2^2-x3^2)*f1+(x3^2-x1^2)*f2+(x1^2-x2^2)*f3)/((x2-x3)*f1+ ...
26         (x3-x1)*f2+(x1-x2)*f3);
27         kline(k)=k;error(k) = abs(f(x2)-f(x));
28         if (abs(f(x2)-f(x))<epsilon*f(x2))
29             break;
30         end
31         k=k+1;
32     end
33     if (f(x)<f(2))
34         output = x;
35     else
36         output = x2;
37     end
38     kline(k)=k;error(k) = abs(f(x2)-f(x));
39     %删除0值
40     kline(kline==0)=[];
41     error(error==0)=[];
42     plot(kline,error,'-or');
43     grid on;
44     title("Distance between selected points—三点二次");
45     xlabel('Iterations');
46     ylabel('Distance');

```

二 作业 7 第三题

2.1 题目阐述

编写 SR1 方法, DFP 方法和 BFGS 方法程序并提供数值实验结果报告. 考虑最优化问题:

$$\min \sum_{i=1}^m r_i^2(\mathbf{x})$$

其中, $r_i(\mathbf{x})$ 由 Watson 函数定义:

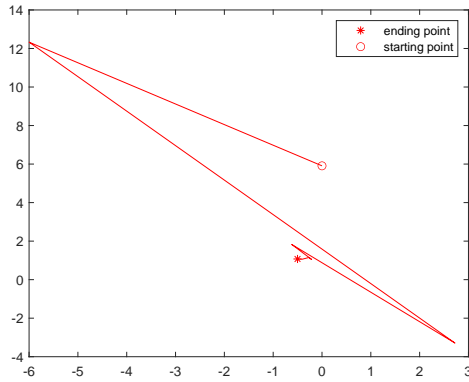
$$r_i(\mathbf{x}) = \sum_{j=2}^n (j-1)x_j t_i^{j-2} - \left(\sum_{j=1}^n x_j t_i^{j-1} \right)^2 - 1$$

其中 $t_i = \frac{i}{29}$ $1 \leq i \leq 29$, $r_{30}(\mathbf{x}) = x_1$, $r_{31} = x_2 - x_1^2 - 1$, $2 \leq n \leq 31$, $m = 31$, 初始值可选为 $x^{(0)} = (0, \dots, 0)^T$

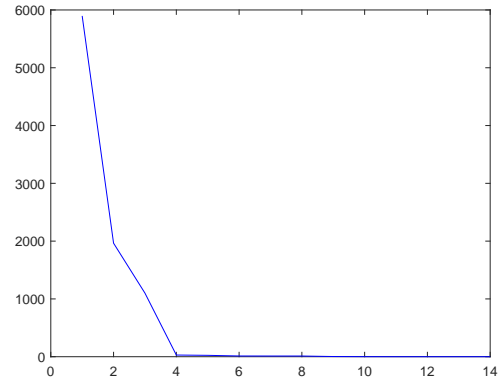
考虑到计算机性能等问题, 下文展示 $n = 2$ 时的图像结果与 $n = 3$ 时的数值结果. 算法采用 Armijo 准则确定搜索步长, $\rho = 0.001$.

2.2 SR1 法

当 $n = 2$ 时, 函数值的更新如图4(a)所示, 其梯度数值的变化如图4(b)所示: 最终迭代次数为 14 次, 极小点为 $(-0.50137, 1.0736)$, 函数值为 0.54661.



(a) SR1 法迭代路径



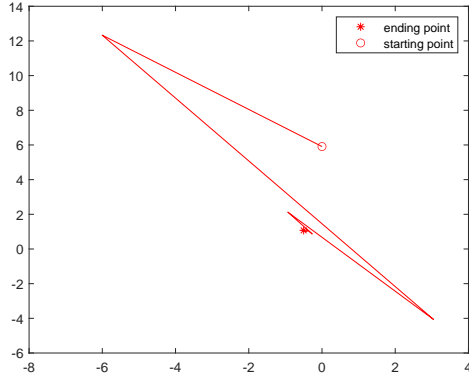
(b) SR1 法梯度数值变化

图 4: SR1 法结果图, $n = 2$

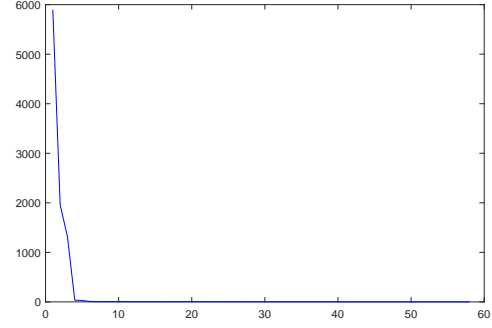
当 $n = 3$ 时, 最终迭代次数为 23 次, 极小点为 $(-0.37573, 0.92779, 0.17164)$, 函数值为 0.4714.

2.3 DFP 法

当 $n = 2$ 时, 函数值的更新如图5(a)所示, 其梯度数值的变化如图5(b)所示: 最终迭代次数为 58 次, 极小点为 $(-0.50137, 1.0736)$, 函数值为 0.54661.



(a) DFP 法迭代路径



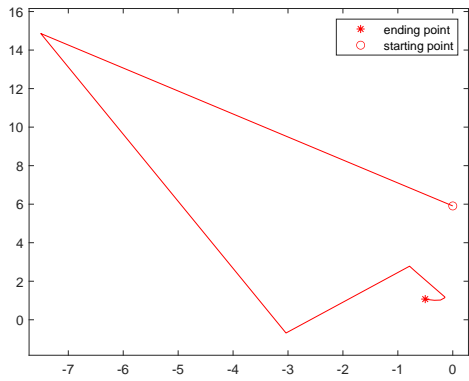
(b) DFP 法梯度数值变化

图 5: DFP 法结果图, $n = 2$

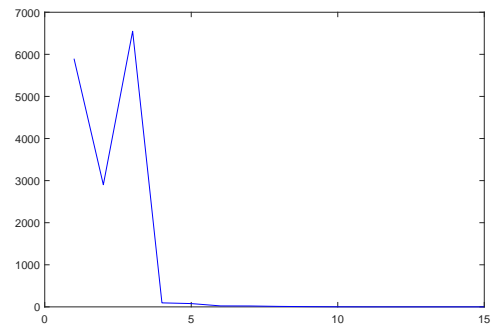
当 $n = 3$ 时, 最终迭代次数为 386 次, 极小点为 $(-0.37573, 0.92779, 0.17164)$, 函数值为 0.4714 .

2.4 BFGS 法

当 $n = 2$ 时, 函数值的更新如图6(a)所示, 其梯度数值的变化如图6(b)所示: 最终迭代次数为 15 次, 极小点为 $(-0.50137, 1.0736)$, 函数值为 0.54661.



(a) BFGS 法迭代路径



(b) BFGS 法梯度数值变化

图 6: BFGS 法结果图, $n = 2$

当 $n = 3$ 时, 最终迭代次数为 52 次, 极小点为 $(-0.37573, 0.92779, 0.17164)$, 函数值为 0.4714 .

2.5 上机程序

SR1,DFP,BFGS 法上机代码

```
1  n = 3; %维数
2  method = "DFP"
3  x0 = zeros(n,1); %初始点
4  H0 = eye(n); %初始矩阵
5  x = x0;
6  H = H0;
7  X = sym(zeros(n,1));
8  e = 1e-5; %精度
9  step = 0; %迭代步数
10 %下面计算目标函数f及其梯度和Hessian矩阵
11 for i = 1:n
12     X(i,1) = sym(['x_', num2str(i)]);
13 end
14 f = 0; %函数值
15 for i = 1:29
16     r = 0;
17     sum = 0;
18     for j = 1:n
19         r = r + (j-1)*X(j)*(i/29)^(j-2);
20         sum = sum + X(j)*(i/29)^(j-1);
21     end
22     r = r - sum^2 - 1;
23     f = f + r^2;
24 end
25 f = f + X(1)^2 + (X(2) - X(1)^2 - 1)^2;
26 g = sym(zeros(n,1));
27 for i = 1:n
28     g(i,1) = diff(f, X(i)); %函数的偏导数值
29 end
30 grad0 = zeros(14,1); %用于在n=2时绘制图像
31 func = zeros(14,3);
32 while(1)
33     step = step+1;
```

```

34     g_val_old = eval(subs(g,X,x));
35     d = -H*g_val_old;
36     alpha = Armijo(f, X, x, g_val_old, d, 0.9, 0.001);
37     x_new = x+alpha*d;%按照Armijo准则更新
38     g_val_new = eval(subs(g,X,x_new));
39     grad = sqrt(g_val_new'*g_val_new);
40     if grad <= e
41         break;
42     end
43     s = x_new - x;
44     y = g_val_new - g_val_old;
45     H = update(H, s, y, method);
46     x = x_new;
47     disp("迭代代数： "+step, +"梯度的模长为"+grad, +"迭代点为("+ ...
48         toString(x_new)+"),函数值为"+eval(subs(f,X,x)))
49 disp("("+toString(x_new)+")")
50 %     func(step,:) = [x_new' eval(subs(f,X,x))];
51 %     grad0(step,1) = grad;
52 end
53 disp("-----")
54 disp("方法为:"+method+";迭代次数： "+step+",极小点为("+toString(x_new)+ ...
55     "),函数值为"+eval(subs(f,X,x)))
56 % grad0(step,1) = grad;
57 % func(step,:) = [x_new' eval(subs(f,X,x))];
58 % plot(func(:,1),func(:,2),'r')
59 % hold on
60 % plot(func(end,1),func(end,2),'r*')
61 % plot(func(1,1),func(1,2),'ro')
62 % legend("", "ending point", "starting point")
63 % hold off
64 % plot(grad0,"b")
65
66 function H1 = update(H, s, y, method)
67 %SR1,DFP,BFGS方法
68 switch method
69     case "SR1"
70         H1 = H + (s-H*y)*((s-H*y)')/((s-H*y)'*y);
71     case "DFP"
72         H1 = H + s*s'/(s'*y) - H*y*y'*H/(y'*H*y);
73     case "BFGS"

```

```

74         H1 = H + (1+(y'*H*y)/(y'*s))*((s*s')/(y'*s))-((s*y'*H+H*y*s')/(y'*s));
75     end
76 end
77
78 function alpha = Armijo(f, X, x0, g0, d, sigma, rho)
79 %Armijo准则
80 alpha = 1;
81 f0 = subs(f,X,x0);
82 while(1)
83     x = x0+alpha*d;
84     fk = subs(f,X,x);
85     if fk <= f0 + rho*g0'*d*alpha
86         break
87     end
88     alpha = alpha*sigma;
89     if alpha <= 10e-2
90         break
91     end
92 end
93 end
94 function s=toString(x)
95 [r,~] = size(x);
96 s = "";
97 for i = 1:r
98     s = s + num2str(x(i));
99     if i ~= r
100         s = s+", ";
101     end
102 end
103 end

```