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)

一　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 $\frac{\sqrt{2}}{2}$.

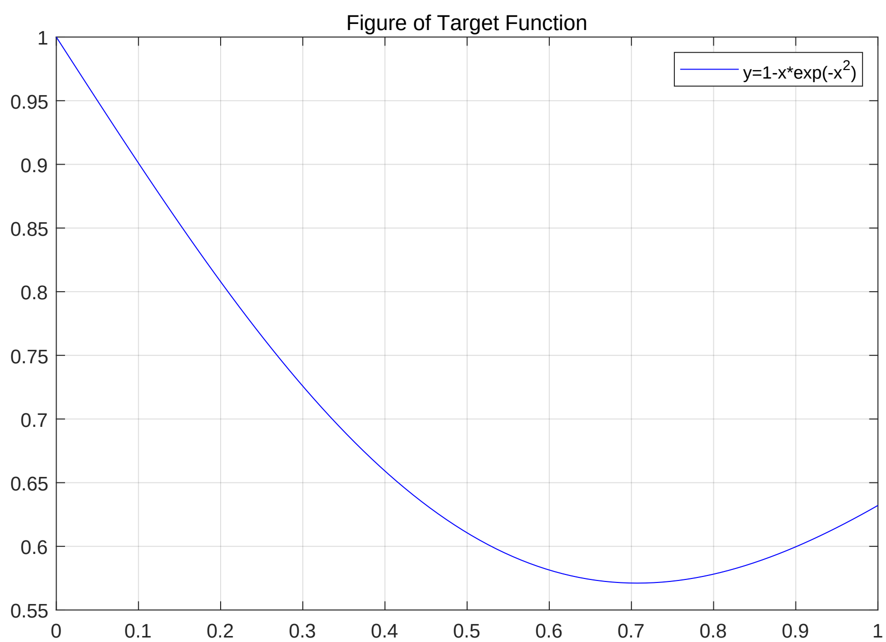


图 1: 目标函数图像

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.

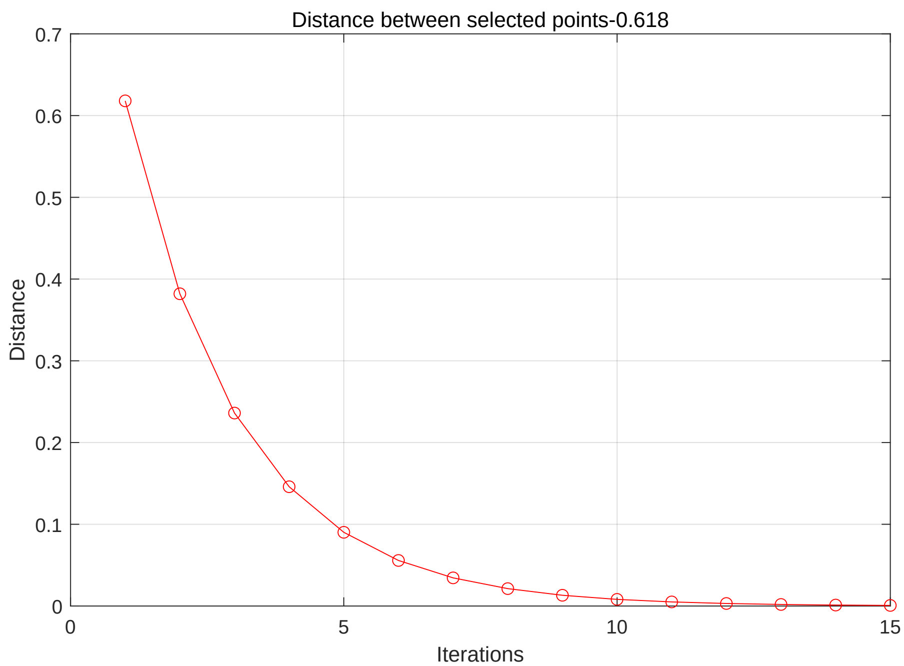


图 2: 0.618 法误差收敛图像

0.618 法 matlab 代码

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

```
            else
            a(k+1)=lambda(k);b(k+1) = b(k);lambda(k+1) = mu(k);
            mu(k+1) = a(k+1)+0.618*(b(k+1)-a(k+1));
            end
        else
            Sign = 2;
            if (mu(k)-a(k)<=epsilon)
                output = lambda(k);
                break;
            else
                a(k+1) = a(k);b(k+1) = mu(k);mu(k+1) = lambda(k);
                lambda(k+1) = a(k+1)+0.382*(b(k+1)-a(k+1));
            end
        end
        kline(k) = k;
        if (Sign==1)
            error(k) = b(k)-lambda(k);
        elseif (Sign==2)
            error(k) = mu(k)-a(k);
        end
        k=k+1;
end
%输出结论
if (b(k)-lambda(k)<=epsilon)
        output = mu(k);
elseif (mu(k)-a(k)<=epsilon)
        output = lambda(k);
end
%加入最后一次迭代的信息
kline(k) = k;
if (Sign==1)
        error(k) = b(k)-lambda(k);
elseif (Sign==2)
        error(k) = mu(k)-a(k);
end
%删除0值
kline(kline==0)=[];
error(error==0)=[];
%绘图
plot(kline, error, '-or');
```

```
grid on;
title ("Distance between selected points-0.618");
xlabel(' Iterations ');
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

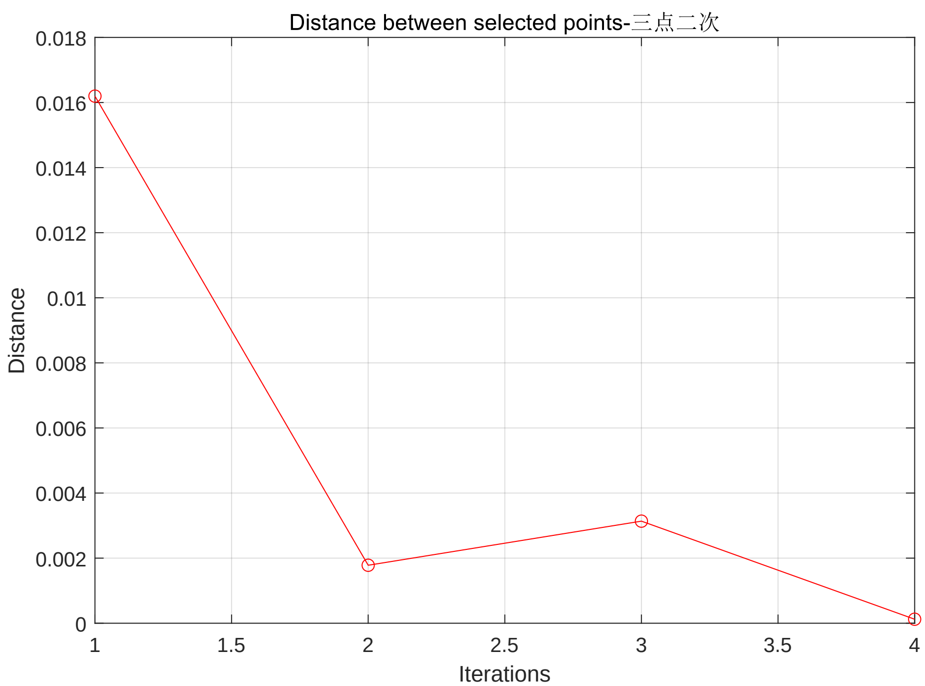


图 3: 三点二次插值法误差收敛图像

三点二次插值法 matlab 代码

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

```
    %以下为画图考虑
    k=1;kline=zeros(1,100,'int8'); error = zeros(1,100,'double');
while(abs(f(x2)-f(x))>=epsilon*f(x2))
    fx=f(x);f1=f(x1);f2=f(x2);f3=f(x3);
    if (x>x2)
        if (fx<=f2)
            x1=x2;x2=x;
        else
            x3=x;
        end
    else
        if (fx<=f2)
            x3=x2;x2=x;
        else
            x1=x;
        end
    end
    x=0.5*((x2^2-x3^2)*f1+(x3^2-x1^2)*f2+(x1^2-x2^2)*f3)/((x2-x3)*f1+ ...
    (x3-x1)*f2+(x1-x2)*f3);
    kline(k)=k;error(k) = abs(f(x2)-f(x));
    if (abs(f(x2)-f(x))<epsilon*f(x2))
        break;
    end
    k=k+1;
end
if (f(x)<f(2))
    output = x;
else
    output = x2;
end
kline(k)=k;error(k) = abs(f(x2)-f(x));
%删除0值
kline(kline==0)=[];
error(error==0)=[];
plot(kline, error, '-or');
grid on;
title("Distance between selected points-三点二次");
xlabel(' Iterations ');
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：

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

其中, $r_i(\boldsymbol{x})$ 由 Watson 函数定义：

$$r_i(\boldsymbol{x})=\sum_{j=2}^{n}(j-1)x_jt_i^{j-2}-(\sum_{j=1}^{n}x_jt_i^{j-1})^2-1$$

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

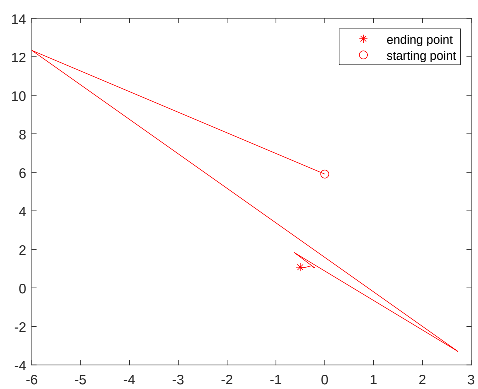
考虑到计算机性能等问题, 下文展示 $n=2$ 时的图像结果与 $n=3$ 时的数值结果.

算法采用 Armijo 准则确定搜索步长,$\rho=0.001$.

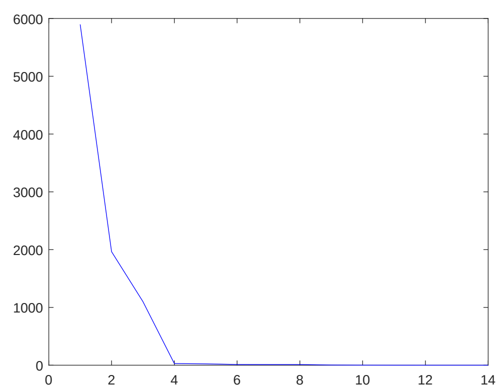
2.2　SR1 法

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



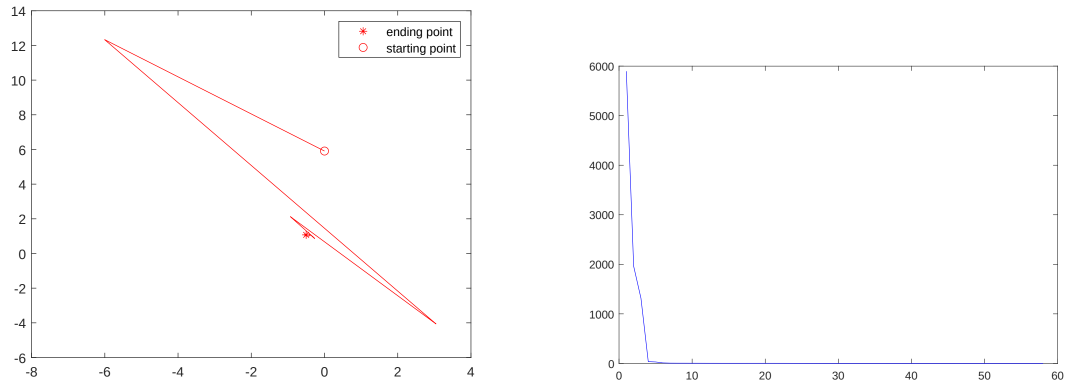
(b) SR1 法梯度数值变化

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.



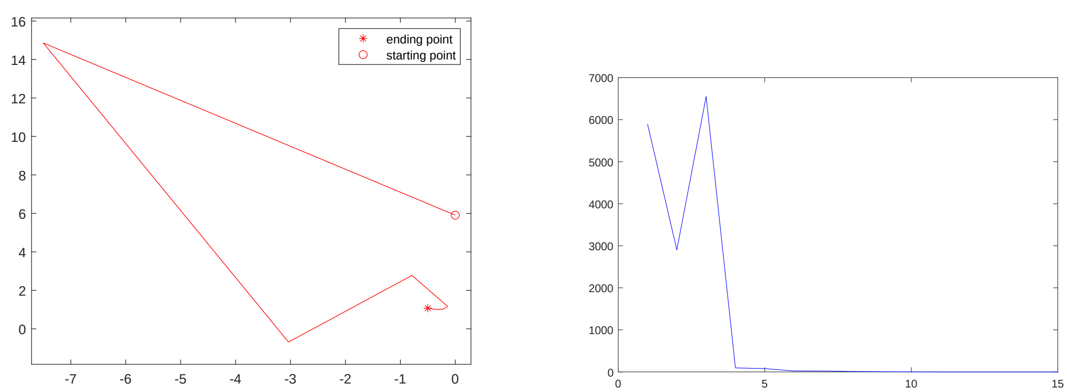
(a) DFP 法迭代路径 (b) DFP 法梯度数值变化

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 BGFS 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

```
n = 3; %维数
method = "DFP"
x0 = zeros(n,1); %初始点
H0 = eye(n);      %初始矩阵
x = x0;
H = H0;
X = sym(zeros(n,1));
e = 1e-5; %精度
step = 0;    %迭代步数
%下面计算目标函数f及其梯度和Hessian矩阵
for i = 1:n
    X(i,1) = sym(['x_',num2str(i)]);
end
f = 0; %函数值
for i = 1:29
    r = 0;
    sum = 0;
    for j = 1:n
        r = r + (j-1)*X(j)*(i/29)^(j-2);
        sum = sum + X(j)*(i/29)^(j-1);
    end
    r = r-sum^2-1;
    f = f + r^2;
end
f = f+X(1)^2+(X(2)-X(1)^2-1)^2;
g = sym(zeros(n,1));
for i = 1:n
    g(i,1) = diff(f,X(i));%函数的偏导数值
end
grad0 = zeros(14,1);%用于在n=2时绘制图像
func = zeros(14,3);
while(1)
    step = step+1;
```

```
    g_val_old = eval(subs(g,X,x));
    d = -H*g_val_old;
    alpha = Armijo(f, X, x, g_val_old, d, 0.9, 0.001);
    x_new = x+alpha*d;%按照Armijo准则更新
    g_val_new = eval(subs(g,X,x_new));
    grad = sqrt(g_val_new'*g_val_new);
    if grad <= e
        break;
    end
    s = x_new - x;
    y = g_val_new - g_val_old;
    H = update(H, s, y, method);
    x = x_new;
    disp("迭代代数："+step, +"梯度的模长为"+grad, +"迭代点为("+ ...
        toString(x_new)+"),函数值为"+eval(subs(f,X,x)))
disp("("+toString(x_new)+")")
%     func(step,:) = [x_new' eval(subs(f,X,x))];
%     grad0(step,1) = grad;
end
disp("------------------------------------------------------------------------
disp("方法为:"+method+";迭代次数："+step+",极小点为("+toString(x_new)+ ...
    "),函数值为"+eval(subs(f,X,x)))
% grad0(step,1) = grad;
% func(step,:) = [x_new' eval(subs(f,X,x))];
% plot(func(:,1),func(:,2),'r')
% hold on
% plot(func(end,1),func(end,2),'r*')
% plot(func(1,1),func(1,2),'ro')
% legend("","ending point","starting point")
% hold off
% plot(grad0,"b")

function H1 = update(H, s, y, method)
%SR1,DFP,BFGS方法
switch method
    case "SR1"
        H1 = H + (s-H*y)*((s-H*y)')/((s-H*y)'*y);
    case "DFP"
        H1 = H + s*s'/(s'*y) - H*y*y'*H/(y'*H*y);
    case "BFGS"
```

```
        H1 = H + (1+(y'*H*y)/(y'*s))*((s*s')/(y'*s))-((s*y'*H+H*y*s')/(y'*s));
end
end

function alpha = Armijo(f, X, x0, g0, d, sigma, rho)
%Armijo准则
alpha = 1;
f0 = subs(f,X,x0);
while(1)
    x = x0+alpha*d;
    fk = subs(f,X,x);
    if fk <= f0 + rho*g0'*d*alpha
        break
    end
    alpha = alpha*sigma;
    if alpha <= 10e-2
        break
    end
end
end
function s=toString(x)
[r,~] = size(x);
s = "";
for i = 1:r
    s = s + num2str(x(i));
    if i ~= r
        s = s+",";
    end
end
end
```