西安電子科技大學

题　　目：_____

学　　院：_____

专　　业：_____

姓　　名：_____

学　　号：_____

简述：

一维搜索又称线性搜索(Line Search),就是指单变量函数的最优化,它是多变量函数最优化的基础,是求解无约束非线性规划问题的基本方法之一。

在多变量函数的最优化中,迭代格式 $x_{k+1}=x_k+a_kd_k$ 其关键就是构造搜索方向 d_k 和步长因子 a_k

设 $\Phi(a)=f(x_k+ad_k)$,这样从凡出发,沿搜索方向 d_k,确定步长因子 a_k,使 $\Phi(a)<\Phi(0)$ 的问题就是关于步长因子 a 的一维搜索问题。其主要结构可作如下概括:首先确定包含问题最优解的搜索区间，然后采用某种分割技术或插值方法缩小这个区间，进行搜索求解。

一维搜索通常分为精确的和不精确的两类。如果求得 a_k 使目标函数沿方向 d_k 达到

极小,即使得

$$f(x_k+a_kd_k)=\min f(x_k+a_kd_k) \qquad (a>0)$$

则称这样的一维搜索为最优一维搜索,或**精确一维搜索**,a_k 叫**最优步长因子**;

如果选取 a_k 使目标函数 f 得到可接受的下降量,即使得下降量 $f(x_k)-f(x_k+a_kd_k)>0$ 是用户可接受的,则称这样的一维搜索为近似一维搜索,或**不精确**一维搜索, 或可接受一维

搜索。

由于在实际计算中, 一般做不到精确的一维搜索, 实际上

也没有必要做到这一点,因为精确的一维搜索需要付出较高的代价,而对加速收敛作用不大,因此花费计算量较少的不精确一维搜索方法受到了广泛的重视和欢迎。

精确一维搜索,作为一种理想的状态,虽然在实际计算中被采用的概率较之不精确一维搜索要小,但有关精确一维搜索技术的研究历史悠久成果相当丰富,方法众多,其理论体系也相对比较完备,对其进行进一步的研究仍有着重要的理论意义和现实意义。通常我们根据算法中有无使用导数的情况,将精确一维搜索算法分为两大类:一类是不用函数导数的方法,这其中就包括二分法(又称作对分法或中点法)、0.618 法(黄金分割脚、Fibonacci 法(分数法)、割线法、成功一失败法等;另一类是使用函数导数的方法,包括经典的 Newton 法、抛物线法以及各种插值类方法等。

实验目的:

对问题 $\min f(x)=3x^4-4x^3-12x^2$ 用黄金法分割法和牛顿法求解最优解。

实验环境:

JAVA 编程，MATLA 绘图

实验内容与步骤:

一、Newton 切线法:

Newton 切线法迭代步骤

(1) 确定初始搜索区间$[a,b]$，要求$\varphi'(a)<0,\varphi'(b)>0.$

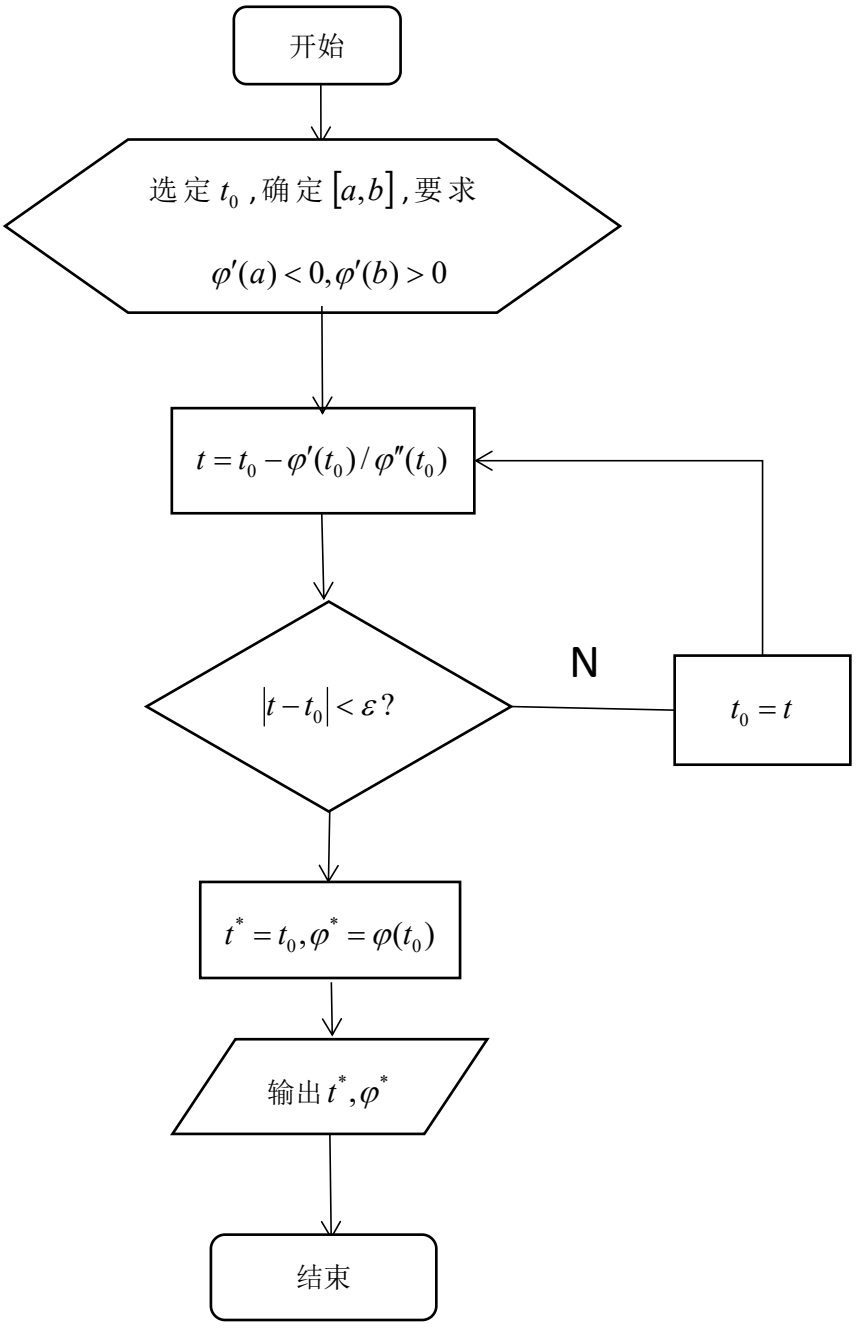
(2) 选定t_0.

(3) 计算$t=t_0-\varphi'(t_0)/\varphi''(t_0)$.

(4) 若$|t-t_0|\geq\varepsilon$,则$t_0=t$，转(3);否则转(5)

(5) 打印t, $\varphi(t)$,结束.

Newton 切线法的计算流程图如下：



代码实现：

```
public class Newton {
    public static void Newton(double a,double b,double t0,double e){
        double t = t0;
        int count=0;
        do{
            t0=t;//第一次 t0 就是本身
            count++;
            t=t0-y1(t0)/y2(t0);
        }while(Math.abs(t-t0)>=e);
        System.out.println("最优解 t*: "+t0+"    对应的最优值 y(t*):
"+y(t0));
        System.out.println("迭代次数: "+count);
    }
    public static double y(double x){
        return 3*Math.pow(x, 4)-4*Math.pow(x, 3)-12*Math.pow(x, 2);
        //y=3x^4-4x^3-12x^2
    }
    public static double y1(double x){
        return 12*Math.pow(x, 3)-12*Math.pow(x, 2)-24;
        //y'=12x^3-12x^2-24x
    }
    public static double y2(double x){
        return 36*Math.pow(x, 2)-24*x-24;
        //y"=36x^2-24x-24
    }
    public static void main(String[] args) {
        Newton(1,3,2.9,0.0001);
    }
}
```

二、黄金分割法:

黄金分割法迭代步骤：

已知$\varphi(t)$，常数$\beta=0.382$，终止限 ε .

(1) 确定$\varphi(t)$的搜索区间$[a,b]$.

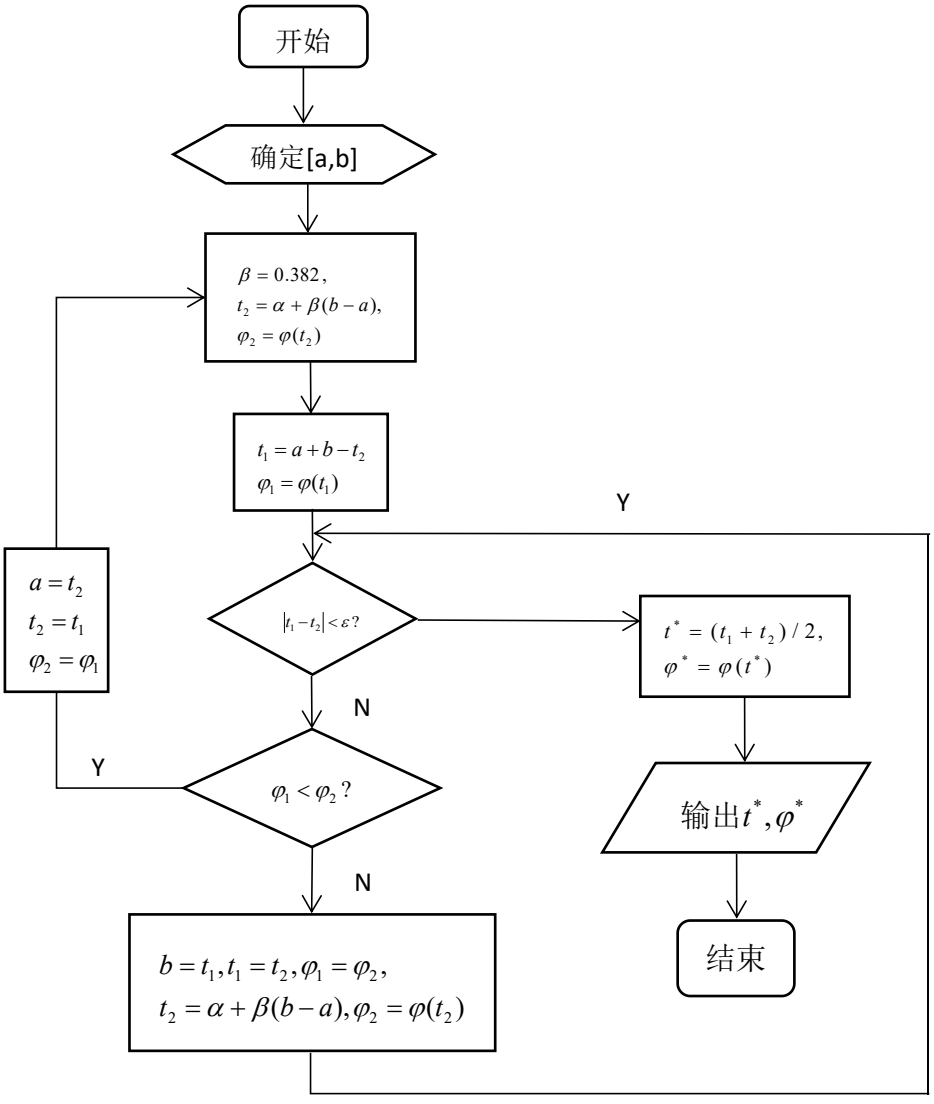
(2) 计算$t_2=\alpha+\beta(b-a),\varphi_2=\varphi(t_2).$

(3) 计算$t_1=a+b-t_2,\varphi_1=\varphi(t_1).$

(4)　若$|t_1-t_2|<\varepsilon$,则打印$t^*=\frac{t_1+t_2}{2}$,结束，否则转(5).

(5)　判别是否满足$\varphi_1<\varphi_2$:若满足,则置$a=t_2,t_2=t_1,\varphi_2=\varphi_1$,然后转(3);否则，置$b=t_1,t_1=t_2,\varphi_1=\varphi_2,t_2=\alpha+\beta(b-a),\varphi_2=\varphi(t_2)$,然后转(4).

黄金分割法流程图如下：



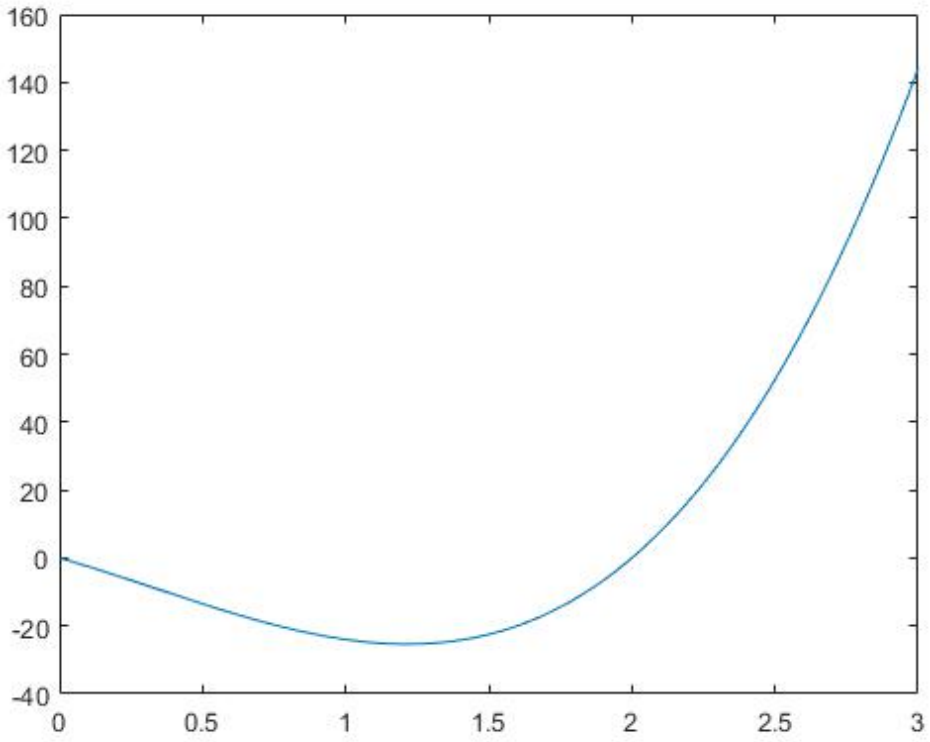
代码实现：

```
public class GoldSearch {
    public static void GoldSearch(double a, double b, double e){
        //1>>已经确定 f(t)的初始搜多区间[a,b]
        double B = ((3-Math.sqrt(5))/2);//beta 的值
        double t1,t2,f1,f2;
        t2 = a + B*(b-a);//2>>计算 t2,f2
        f2 = fun(t2);
        t1 = a+b-t2;    //3>>计算 t1,f1
        f1 = fun(f2);
        int count = 0;
        while(true){
            if(Math.abs(t1-t2)<e){
                double t = (t1+t2)/2;
                System.out.println("最优点 t*="+t);
                System.out.println("此点对应的函数值："+fun(t));
                System.out.println("迭代步数"+count);
                break;
            }else if(f1<f2){
                a=t2;t2=t1;f2=f1;
                t1 = a+b-t2;    //3>>计算 t1,f1
                f1 = fun(f2);
                count++;
            }else{
                b=t1;t1=t2;f1=f2;
                t2=a+B*(b-a);f2=fun(t2);
                count++;
            }
        }
    }
    private static double fun(double t) {
        return (double) (3*Math.pow(t,4)-4*Math.pow(t,
3)-12*Math.pow(t, 2));
    }
    public static void main(String[] args){
        GoldSearch(2,3,0.0001);
    }
}
```

MATLAB 函数图：

```
% y='3*x.^4-4*x.^3-12*x.^2';
% [min_x,min_y]=fminbnd(y,0,5);
% fplot(y,[0 3])

f='12*x^3 - 12*x^2 - 24*x';
[min_x,min_y]=fminbnd(f,0,5);
fplot(f,[0 3])
```



总结：

一维线性搜索中牛顿法误差比较大，黄金分割法误差比较小。在完成报告时学会了如何利用网络去查找资料整理，通过查看别人的博客然后慢慢的去理解。不懂得时候和同学讨论。大作业是我对一维线性搜索有了更加深刻的理解。