

数值分析期末实验

基本信息

姓名：陈启源

学号：2020211946

实验名称：hooke jeeves算法

文件结构

```
1 D:.\n2   └─Numerical_Mathematics_Assignment\n3       └─main\n4           explore.m\n5           func.m\n6           hookejeeves.m\n7           main.m
```

其中explore.m保存了探查部分的代码，func.m保存了需要求解的函数，hookejeeves.m文件中保存了算法的主要代码，main.m中则是求解算法的主程序，调用了hookejeeves.m中的算法来求解func.m中的函数。

代码

explore.m

```
1 function [x] = explore(n, x0, h)\n2 % 输入\n3 % n 变量数量，x0 初始值（向量），func求解函数，h为步长\n4 %\n5 % 输出\n6 % x 最佳点\n7     x = x0;\n8     f0 = func(x);% 得到f(x_0)
```

```

9      for i = 1:n
10         x(i) = x(i) + h(i);
11         ff = func(x); %在一个方向上尝试
12         if ff < f0
13            % 如果这个方向取值更小了，那么继续尝试下一个ei方向，并
            把最小值赋为当前值
14            f0 = ff;
15         else
16            % 否则就尝试相反方向
17            x(i) = x0(i) - h(i);
18            ff = func(x);
19            if ff < f0
20               % 如果相反方向小了，那么就把最小值赋为当前值，继续
               尝试下一个方向
21               f0 = ff;
22            else
23               % 如果两个方向尝试完，都不如x0处小，那么这个方向上
               最小的点就是x0
24               x(i) = x0(i);
25            end
26         end
27     end

```

func.m

```

1  function y=func(x)
2      y = 100*(x(2)-x(1)^2)^2+(1-x(1))^2;
3  end

```

hookejeeves.m

```

1  function [x, best_value, iter] = hookejeeves(n, x0, h, min_h,
    maxiter)
2  % 输入
3  % n 变量数量, x0 初始值（向量）, h 搜索步长, min_h 最小步长
4  % toll 误差, maxiter 最大迭代次数, func求解函数
5  %
6  % 输出
7  % x 最佳点, best_value函数最佳值, iter迭代次数
8      x = x0;
9      best_value = func(x);

```

```

10     iter = 0;
11     while h > min_h
12         [x_new] = explore(n, x, h); % x_new保存探索后的结果
13         if x_new == x
14             h = h/2; % 探索后的结果还是原来的点，说明不优化了，减
少步长
15         else
16             ff = func(x_new); % 探索方向后的点的函数值
17             direction = x_new - x; % 方向
18
19             % 为了更高效，不能只在这个方向上走一步，需要多走几步，
接下来确定步长
20
21             j = 2;
22             ff_new = ff;
23             x_new1 = j * direction + x;
24
25             while(func(x_new1) < ff_new)
26                 % 当再走一个单位方向后的值更小时
27                 ff_new = func(x_new1);
28                 j = j + 1;
29                 x_new1 = j * direction + x; %走j步
30             end
31
32             % 退出循环的时候，j是最小值的下一步，所以需要使用j-1回
到最小值的j
33             x = (j - 1) * direction + x;
34             % 开始新的迭代
35         end
36         iter = iter+1;
37         if maxiter < iter
38             % 大于迭代步数
39             break
40         end
41     end
42     best_value = func(x);
43 end

```

main.m

```
1  clc;
2  clear;
3  n=2;
4  x0=[-1.2;1];
5  h=[0.6;0.5];
6  min_h=1e-4;
7  maxiter=5000;
8  [x,best_value,iter] = hookejееves(n, x0, h, min_h, maxiter)
```

结果

书本参考结果

$$x = (0.9655, 0.9322)$$
$$iter = 935$$

算法求解结果

$$x = (0.9633, 0.9279)$$
$$best_value = 0.0013$$
$$iter = 775$$

```
x =  
  
    0.9633  
    0.9279  
  
best_value =  
  
    0.0013  
  
iter =  
  
    775
```

