

《最优化方法》课程的实践环节 2  
选题一（课本习题）

解

题

报

告

SY1606220

王志宏

## 目录

1. 习题 5.6 .....	5
1.1. 算法流程图 .....	5
1.2. 重要表达式 .....	6
1.3. 计算结果 .....	6
1.4. 说明和分析 .....	8
2. 习题 5.7 .....	10
2.1. 算法流程图 .....	10
2.2. 重要表达式 .....	10
2.3. 计算结果 .....	11
2.4. 说明和分析 .....	14
3. 习题 5.8 .....	16
3.1. 算法流程图 .....	16
3.2. 重要表达式 .....	17
3.3. 参数设置 .....	18
3.4. 计算结果 .....	18
3.4.1 $\mu=1$ , 无线搜索 .....	18
3.4.2 $\mu=1$ , 有线搜索 .....	19
3.4.3 $\mu=0.1$ , 无线搜索 .....	20
3.4.4 $\mu=0.1$ , 有线搜索 .....	21
3.5. 说明和分析 .....	22
4. 习题 5.9 .....	24

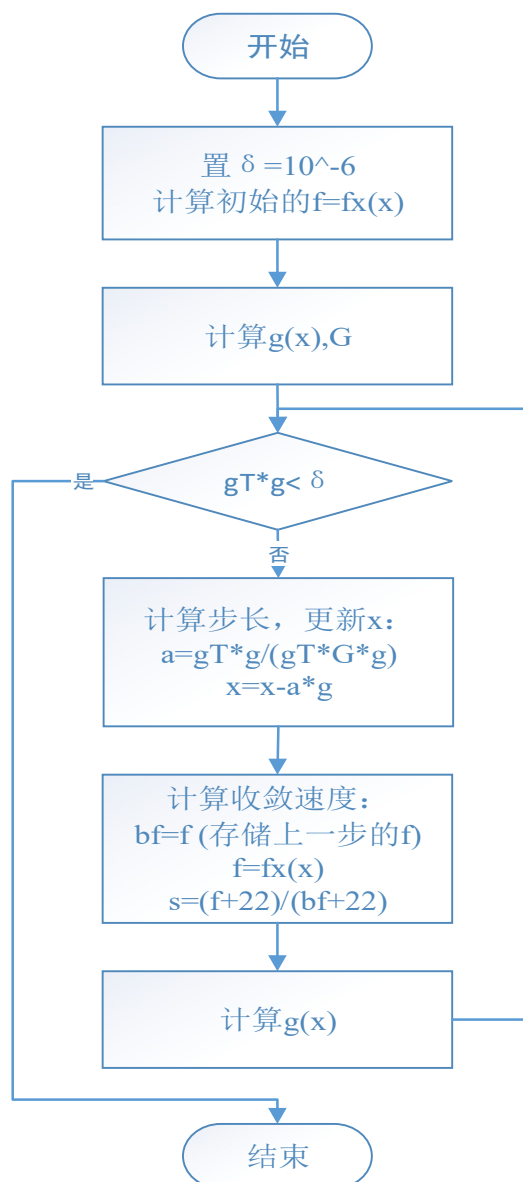
4.1. 算法流程图 .....	24
4.2. 重要表达式 .....	25
4.3. 参数设置 .....	25
4.4. 计算结果 .....	26
4.5. 说明和分析 .....	28
5. 习题 5.19 .....	30
5.1. 算法流程图 .....	30
5.2. 参数选择 .....	31
5.3. 重要表达式 .....	31
5.4. 计算结果 .....	31
5.5. 说明和分析 .....	32
5.4.1 $n=5$ .....	33
5.4.2 $n=8$ .....	33
5.4.3 $n=12$ .....	34
5.4.4 $n=20$ .....	35
6. 习题 5.27 .....	37
6.1. 题目说明 .....	37
6.2. 算法流程图 .....	38
6.3. 重要表达式 .....	39
6.4. 参数设置 .....	40
6.4.1 初始点的选择 .....	40
6.4.2 $\gamma$ 的选择 .....	40

6.5. 计算结果.....	41
6.6. 说明和分析 .....	42
7. 习题 6.4.....	43
7.1. 算法流程图 .....	43
7.2. 重要表达式 .....	45
7.3. 计算结果.....	46
7.4. 说明和分析 .....	47
7.5.1 子问题最多迭代两次: .....	47
7.5.2 逼近度 $\rho$ 的计算.....	48

# 1. 习题 5.6

## 1.1. 算法流程图

最速下降法的算法流程图如下图所示，因为题中 Hessian 阵  $G$  为常矩阵，因此不需要每步都再计算  $G$



## 1.2. 重要表达式

梯度公式：

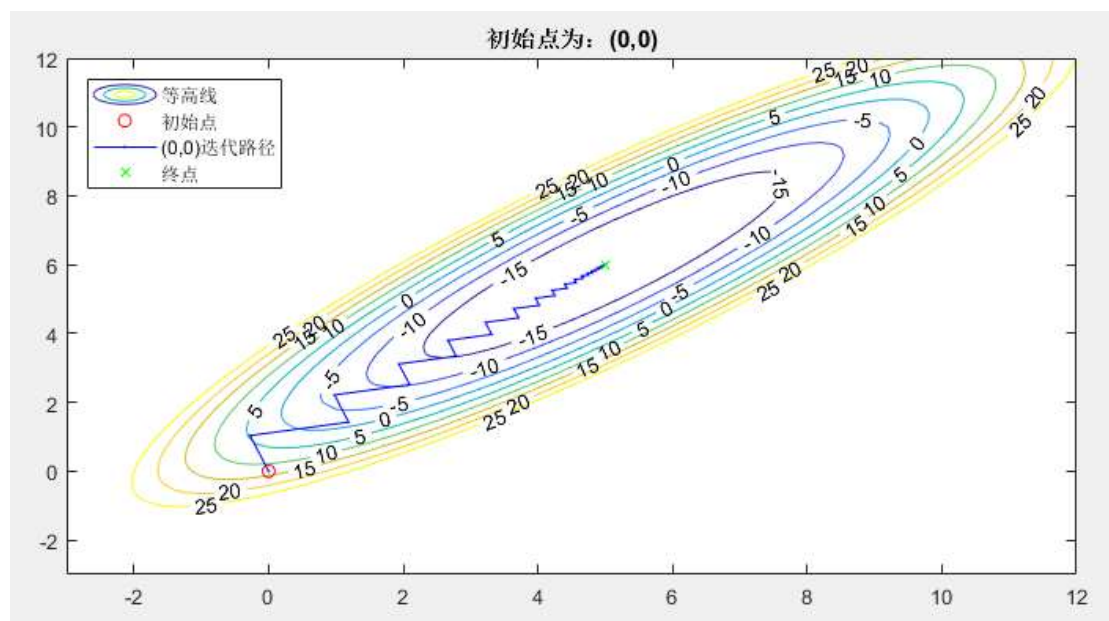
$$\nabla f = g(x) = \begin{bmatrix} 10x_1 - 9x_2 + 4 \\ -9x_1 + 10x_2 - 15 \end{bmatrix}$$

Hessian 阵：

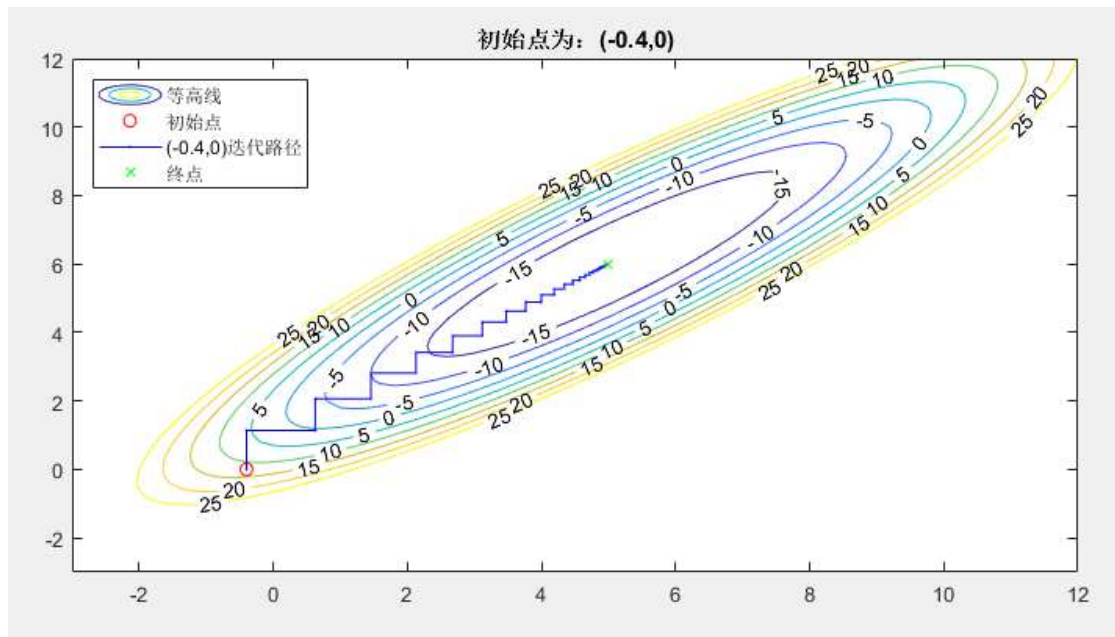
$$\nabla^2 f = G(x) = \begin{bmatrix} 10 & -9 \\ -9 & 10 \end{bmatrix} > 0$$

## 1.3. 计算结果

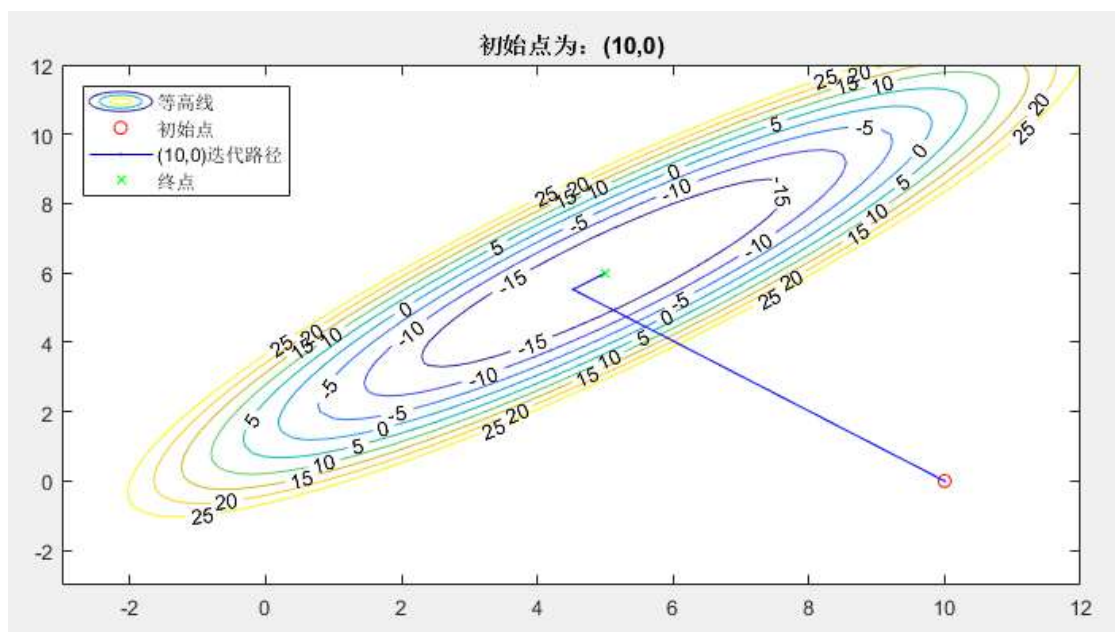
第一个点 (0,0) 的迭代路径如下图所示



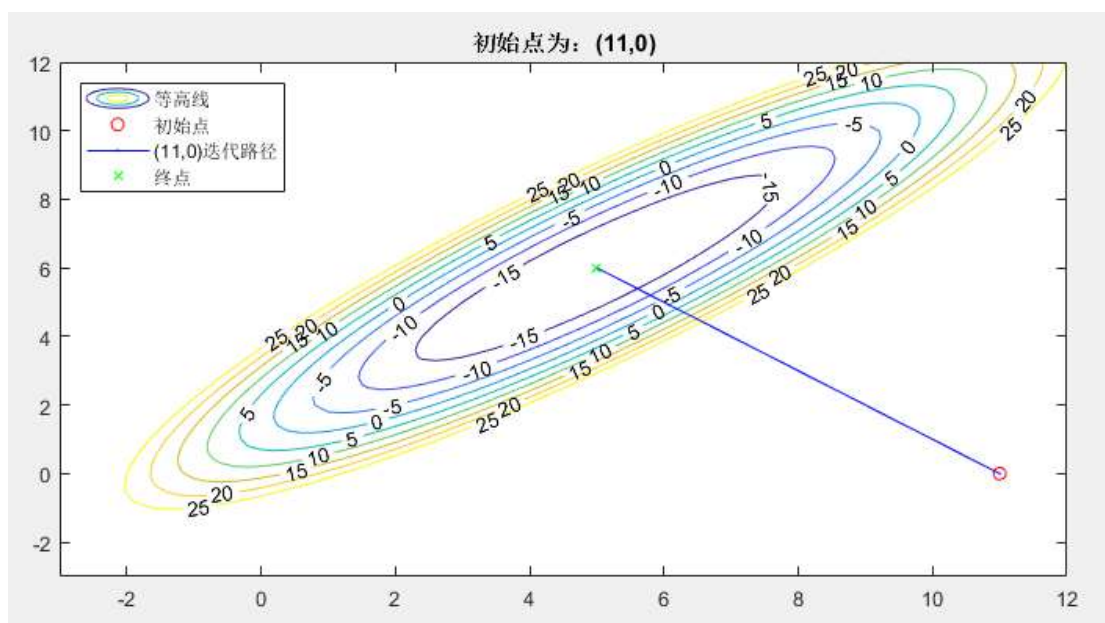
第二个点 (-0.4,0) 的迭代路径如下图所示



第三个点  $(10, 0)$  的迭代路径如下图所示



第四个点  $(10, 0)$  的迭代路径如下图所示



最终的结果如下图所示：

第1次迭代，初始点为(0.000000, 0.000000)，最优值为(4.999999, 5.999999)，收敛因子为0.636364  
 第2次迭代，初始点为(-0.400000, 0.000000)，最优值为(5.000000, 6.000000)，收敛因子为0.777778  
 第3次迭代，初始点为(10.000000, 0.000000)，最优值为(5.000000, 6.000000)，收敛因子为0.002133  
 第4次迭代，初始点为(11.000000, 0.000000)，最优值为(5.000000, 6.000000)，收敛因子为0.000000  
 所有迭代中最大的收敛因子是0.777778

## 1.4. 说明和分析

最速下降法每次选择负梯度方向为迭代方向，利用线搜索找到该方向上的极小（最小值），因为目标函数是二次函数，因此

$$\alpha_k = \frac{\mathbf{g}^{(k)\top} \mathbf{g}^{(k)}}{\mathbf{g}^{(k)\top} \mathbf{G} \mathbf{g}^{(k)}}$$

最速下降法为线性收敛，收敛因子不高于最优值处 hessian 阵的条件数。最优值为  $\mathbf{x}=(5,6)$ ，在最优值的 Hessian 阵  $\mathbf{G}$  为

$$\mathbf{G} = \nabla^2 f = \mathbf{G}(\mathbf{x}) = \begin{bmatrix} 10 & -9 \\ -9 & 10 \end{bmatrix} > 0$$

$$\mathbf{G} = \begin{bmatrix} 10 & -9 \\ -9 & 10 \end{bmatrix} = \begin{bmatrix} -9 \\ -9 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} + 19 * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



故  $G$  的特征值为

$$\lambda(G) = \{-18, 0\} + 19 = \{1, 19\}$$

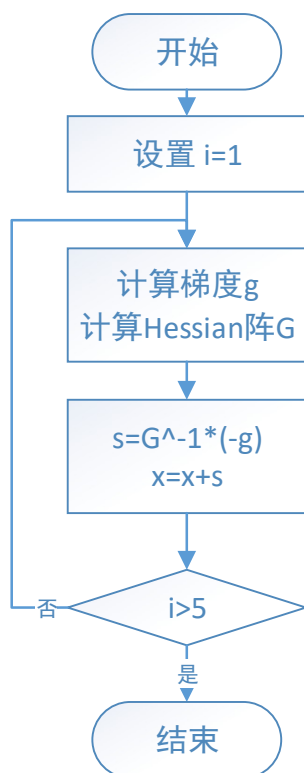
$G$  在二范数下的条件数为

$$\text{conf}(G, 2) = \left( \frac{19 - 1}{19 + 1} \right)^2 = 0.81$$

结合实验结果，最速下降法的收敛因子小于最优点处 Hessian 阵的条件数成立

## 2. 习题 5.7

### 2.1. 算法流程图



### 2.2. 重要表达式

梯度公式:

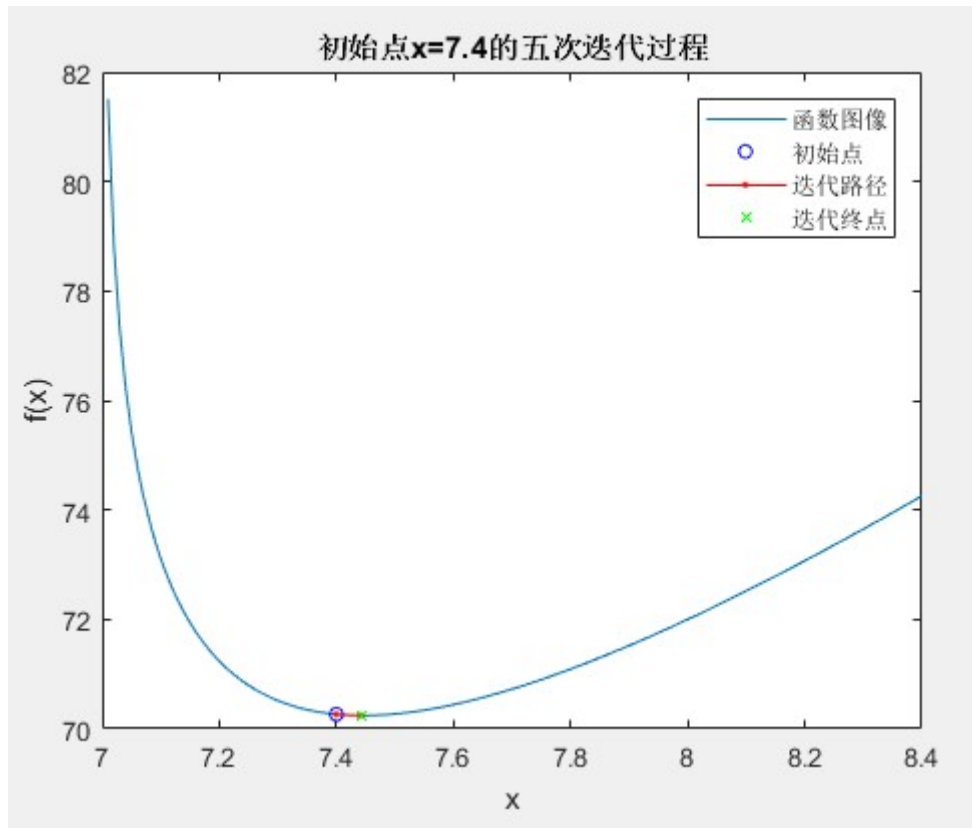
$$\nabla f = g(x) = 9 - \frac{4}{x-7}$$

Hessian 阵:

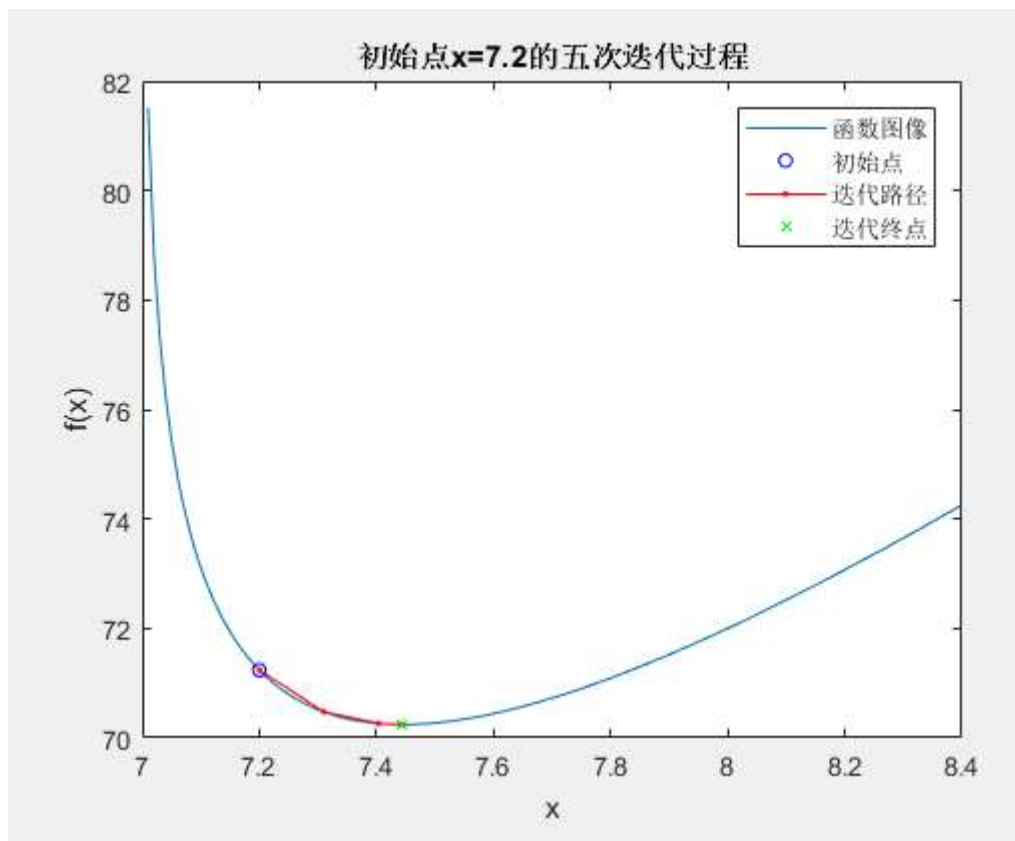
$$\nabla^2 f = G(x) = \frac{4}{(x-7)^2} > 0$$

## 2.3. 计算结果

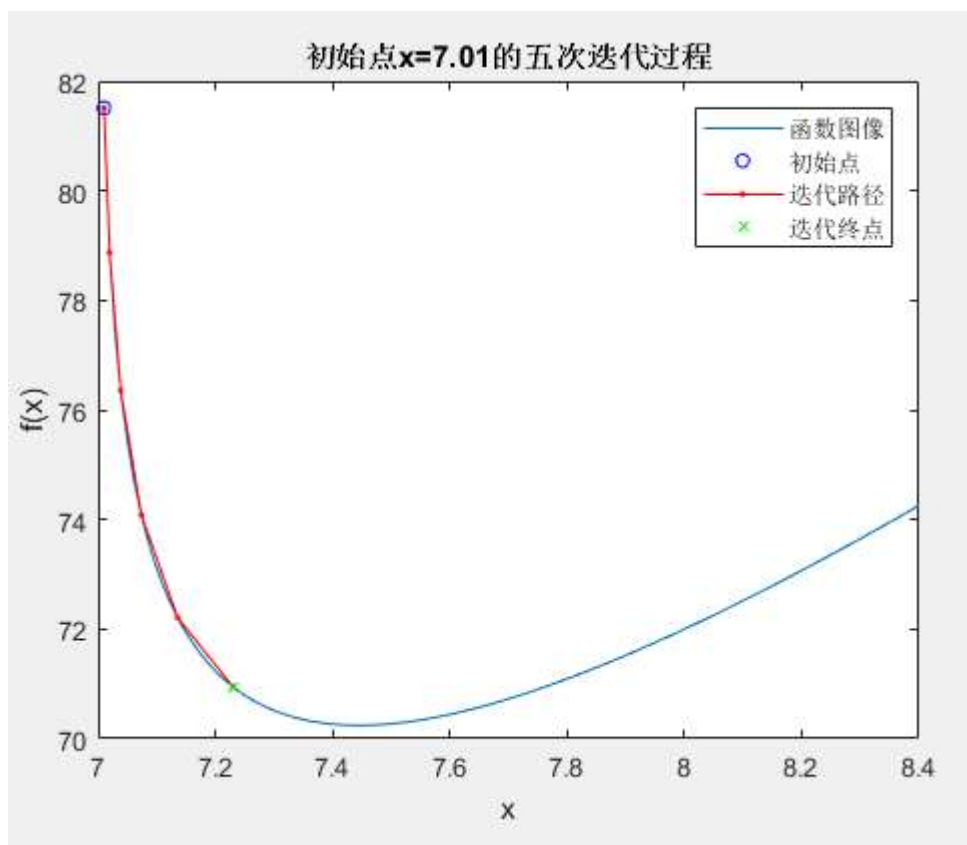
第一个点  $x=7.40$  的 5 步迭代结果：



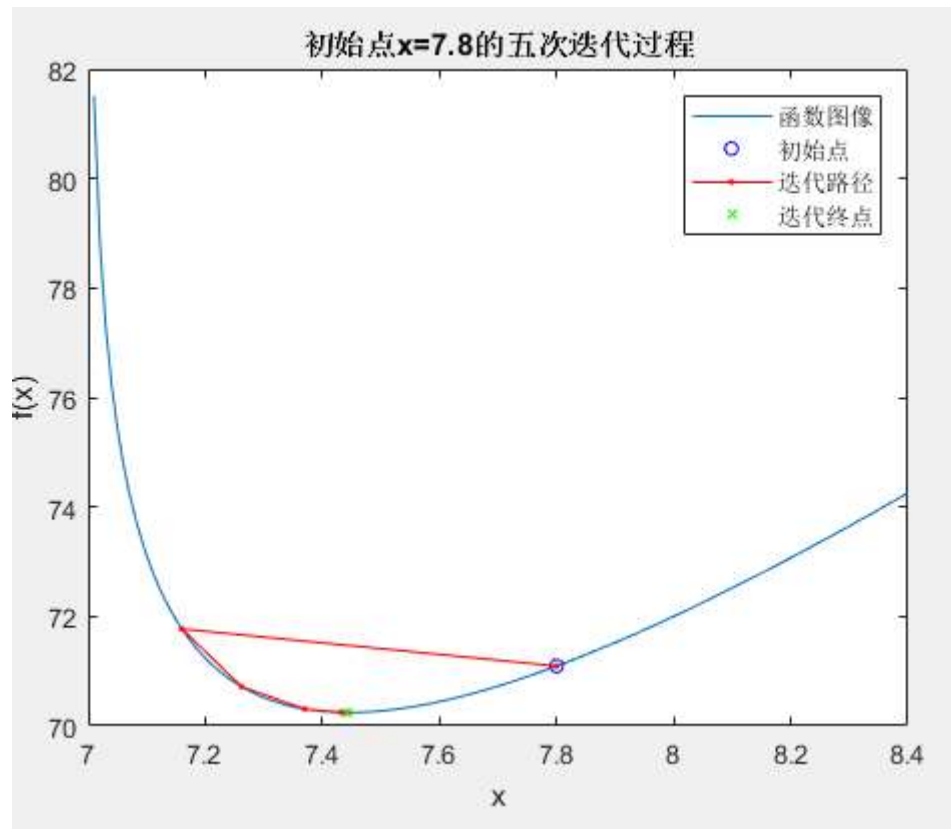
第二个点  $x=7.20$  的 5 步迭代结果：



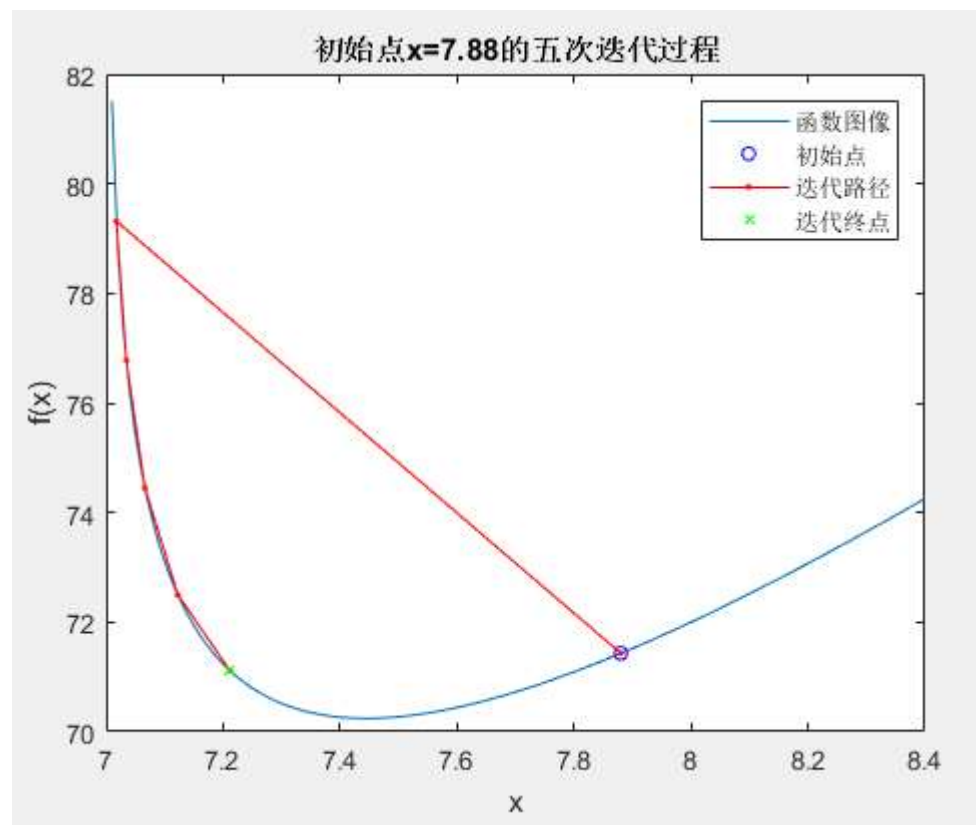
第三个点  $x=7.01$  的 5 步迭代结果:



第四个点  $x=7.80$  的 5 步迭代结果：



第五个点  $x=7.88$  的 5 步迭代结果：



五次迭代和一次区域外初始点的迭代结果如下图所示：

```
第1次迭代：迭代路径为(7.400000, 70.265163)->(7.440000, 70.243922)->
(7.444400, 70.243721)->(7.444444, 70.243721)->(7.444444, 70.243721)->(7.444444, 70.243721)
第2次迭代：迭代路径为(7.200000, 71.237752)->(7.310000, 70.474732)->
(7.403775, 70.261565)->(7.440723, 70.243862)->(7.444413, 70.243721)->(7.444444, 70.243721)
第3次迭代：迭代路径为(7.010000, 81.510681)->(7.019775, 78.871322)->
(7.038670, 76.358782)->(7.073976, 74.081857)->(7.135638, 72.211796)->(7.229882, 70.949696)
第4次迭代：迭代路径为(7.800000, 71.092574)->(7.160000, 71.770326)->
(7.262400, 70.713141)->(7.369879, 70.307228)->(7.431934, 70.245336)->(7.444092, 70.243722)
第5次迭代：迭代路径为(7.880000, 71.431333)->(7.017600, 79.317826)->
(7.034503, 76.777359)->(7.066328, 74.449548)->(7.122757, 72.495017)->(7.211607, 71.116556)

区域外的初始点的迭代：本次迭代之后的x为6.999978, x小于7, f(x)会产生复数, 迭代停止。
上一个x为7.888900, g为4.500056, G为5.062373, s为-0.888922
>>
```

## 2.4. 说明和分析

由函数

$$f(x) = 9x - 4\ln(x - 7)$$

可以得到,  $f(x)$ 的定义域为 $(7, +\infty)$ ,  $f(x)$ 在定义域内连续且二阶可导,  $f(x)$ 的导数  $g(x)$ , 二阶导数  $G(x)$ 分别为:

$$g(x) = 9 - \frac{4}{x - 7}$$

$$G(x) = \frac{4}{(x - 7)^2}$$

$G(x) > 0$ , 故  $f(x)$ 为凸函数, 该优化问题为凸优化。牛顿法每次迭代的步长  $s$  为:

$$s = G^{-1}(-g) = \frac{4}{(x - 7)^2} * \left(9 - \frac{4}{x - 7}\right) = -\frac{9}{4}x^2 + \frac{65}{2}x - \frac{469}{4}$$

因为迭代之后的步长需要在定义域之内, 因此有

$$x + s > 7$$

即

$$-\frac{9}{4}x^2 + \frac{67}{2}x - \frac{497}{4} > 0$$

解之得

$$x \in \left(7, \frac{71}{9}\right), \quad \frac{71}{9} \approx 7.8888, \quad (7, 7.8888) \subset \left(7, \frac{71}{9}\right)$$

因此只有  $x \in \left(7, \frac{71}{9}\right)$ , 迭代产生的  $x=x+s$  才会在定义域之内。

$\forall x \in (7, 7.8888)$ ,

$$h(x) = x + s = -\frac{9}{4}x^2 + \frac{67}{2}x - \frac{469}{4}$$

$$h_{\max}(x) = h\left(\frac{67}{9}\right) = \frac{67}{9}$$

$$h_{\max}(x) = h\left(\frac{71}{9}\right) = h(7) = 7$$

因此在(7,7.8888)区间内的任一点, 迭代之后的点还在这个区间之内 (更严格来说, 迭代之后落在 $\left(7, \frac{67}{9}\right)$ 区间内, 即(7,7.4444)内)。

**结论:**

当 $x \in \left(7, \frac{67}{9}\right)$ 时,  $g(x) = 9 - \frac{4}{x-7} < 0$ , 因此  $s > 0$ , 故  $x' = x + s > x$ 。

即 $\forall x \in (7, 7.8888)$ , 第一次迭代之后落入(7,7.4444)区间内, 之后每次迭代,  $x$  将至少不减 (增加或者不变), 且保持在(7,7.4444)区间之内, 逐步逼近最优点( $x=7.44444$ )

当 $x \in \left(\frac{71}{9}, +\infty\right)$ 时, 第一次迭代之后的  $x < 7$ , 落在了定义域之外, 迭代无法继续, 故算法终止。

## 3. 习题 5.8

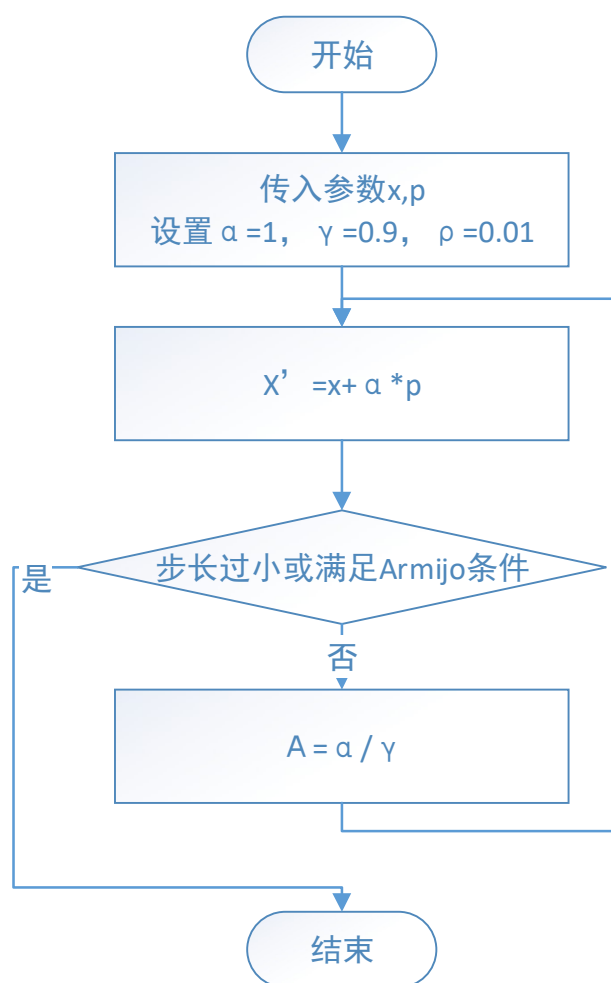
### 3.1. 算法流程图

牛顿法迭代的流程图如下图所示：





一维线搜索的子程序的流程图如下图所示：



### 3.2. 重要表达式

梯度公式：

$$\nabla f = g(x) = \begin{bmatrix} -9 + \frac{u}{100 - x_1 - x_2} + \frac{1}{x_1} - \frac{1}{50 - x_1 + x_2} \\ -10 + \frac{u}{100 - x_1 - x_2} + \frac{1}{x_2} - \frac{1}{50 - x_1 + x_2} \end{bmatrix}$$

Hessian 阵：

$$\nabla^2 f = G(x)$$

$$= \begin{bmatrix} \frac{u}{(100 - x_1 - x_2)^2} - \frac{1}{x_1^2} - \frac{1}{(50 - x_1 + x_2)^2} & \frac{u}{(100 - x_1 - x_2)^2} + \frac{1}{(50 - x_1 + x_2)^2} \\ \frac{u}{(100 - x_1 - x_2)^2} + \frac{1}{(50 - x_1 + x_2)^2} & \frac{u}{(100 - x_1 - x_2)^2} - \frac{1}{x_2^2} - \frac{1}{(50 - x_1 + x_2)^2} \end{bmatrix}$$

### 3.3. 参数设置

若  $G$  负定, 则

$$G = G + (0.5 - \min\{\lambda\}) \cdot I$$

线搜索:

初始步长  $\alpha = 1$

步长变化率  $\gamma = 0.9$

Armijo 条件中  $\rho = 0.01$

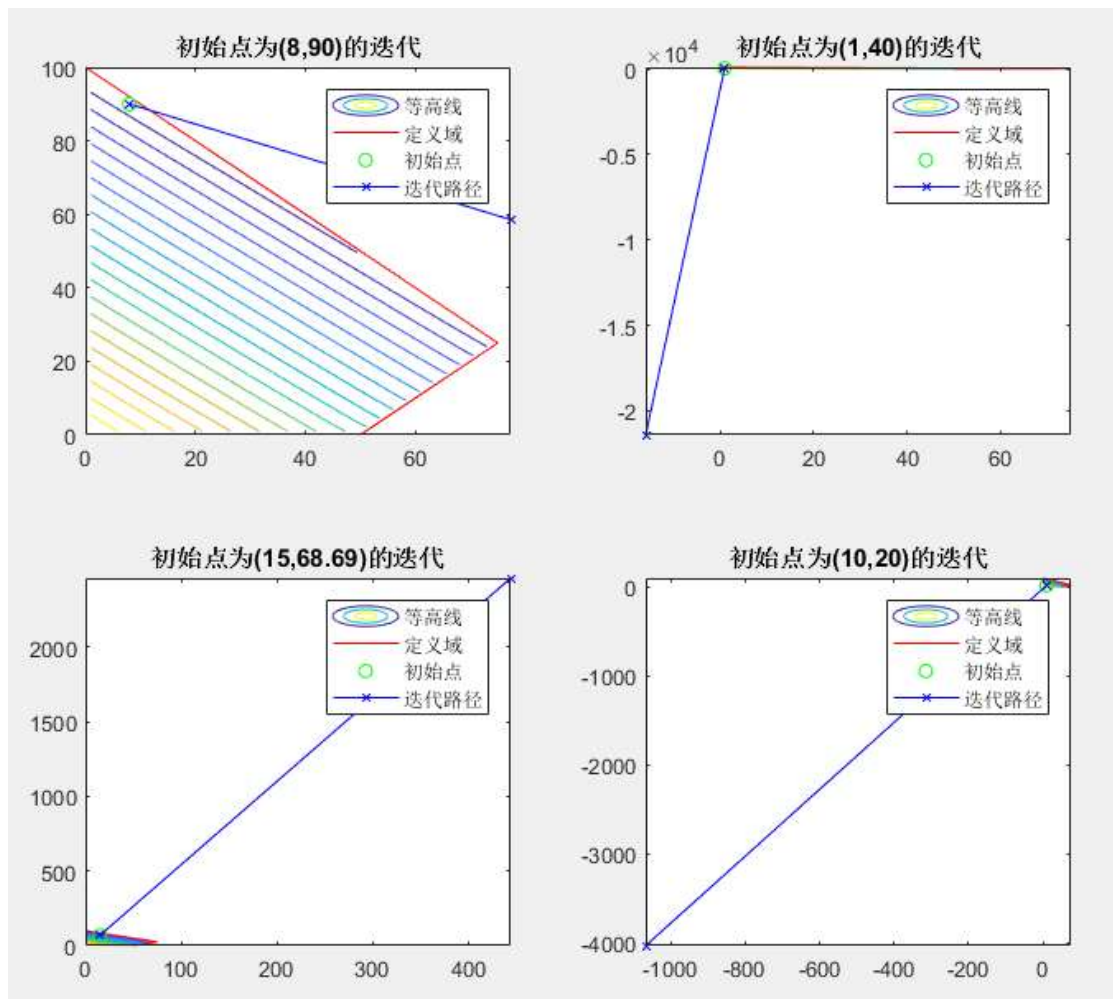
其中步长变化率是从[0.1 0.2 0.4 0.7 0.8 0.9 0.95 0.99]这 8 种情况较好的一种确定下来的, 此时牛顿法迭代的次数比较少。

### 3.4. 计算结果

#### 3.4.1 $\mu = 1$ , 无线搜索

$\mu=1$ , 无线搜索时牛顿法运行结果:

```
初始点为(8, 90)的迭代: 本次迭代在点(8.000000, 90.000000)迭代之后超出定义域。
此点处梯度为[-8.382576;-9.481313].Hessian阵为[0.234318 0.250057;0.250057 0.249819]
本次迭代的步长为(69.330440, -31.443850), 迭代之后的点为(77.330440, 58.556150)
初始点为(1, 40)的迭代: 本次迭代在点(1.000000, 40.000000)迭代之后超出定义域。
此点处梯度为[-7.994287;-9.946815].Hessian阵为[-0.999839 0.000414;0.000414 -0.000464]
本次迭代的步长为(-16.868417, -21453.386664), 迭代之后的点为(-15.868417, -21413.386664)
初始点为(15, 68.69)的迭代: 本次迭代在点(15.000000, 68.690000)迭代之后超出定义域。
此点处梯度为[-8.881665;-9.914486].Hessian阵为[-0.000778 0.003852;0.003852 0.003454]
本次迭代的步长为(428.645194, 2392.222578), 迭代之后的点为(443.645194, 2460.912578)
初始点为(10, 20)的迭代: 本次迭代在点(10.000000, 20.000000)迭代之后超出定义域。
此点处梯度为[-8.902381;-9.919048].Hessian阵为[-0.010074 0.000482;0.000482 -0.002574]
本次迭代的步长为(-1077.727523, -4055.785990), 迭代之后的点为(-1067.727523, -4035.785990)
程序暂停, 输入回车以继续... 输入之后将清空命令行和图示
```



### 3.4.2 $\mu = 1$ , 有线搜索

$\mu = 1$ , 有线搜索时牛顿法运行结果:

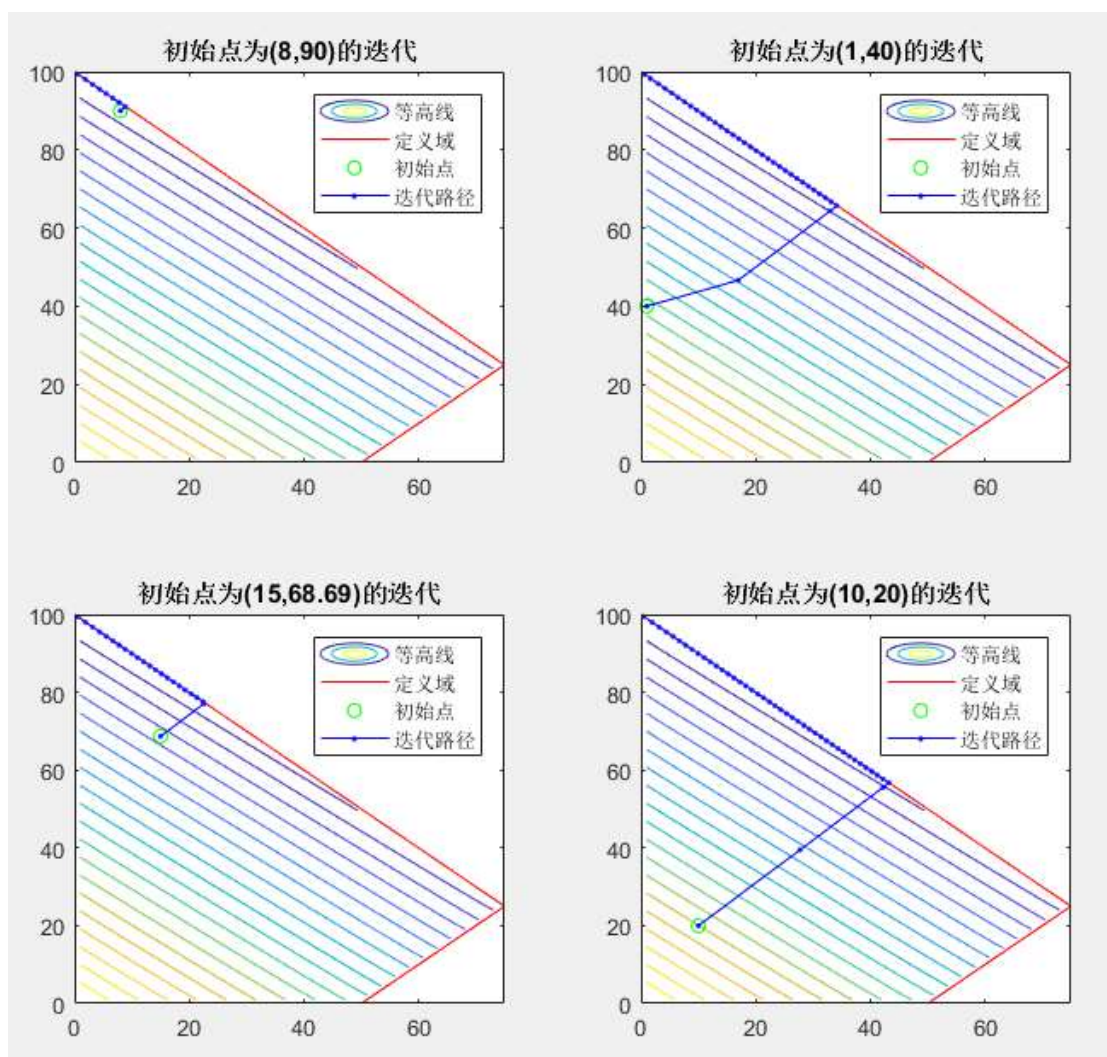
第1个点, 初始点为(8, 90)经过9次迭代终止在(0.51449, 99.375)

第2个点, 初始点为(1, 40)经过36次迭代终止在(0.62286, 99.2664)

第3个点, 初始点为(15, 68.69)经过23次迭代终止在(0.54637, 99.3432)

第4个点, 初始点为(10, 20)经过45次迭代终止在(0.44792, 99.4415)

程序暂停, 输入回车以继续... 输入之后将清空命令行和图示



### 3.4.3 $\mu = 0.1$ , 无线搜索

$u=1$ , 无线搜索时牛顿法运行结果:

初始点为(8,90)的迭代: 本次迭代在点(8.000000, 90.000000)迭代之后超出定义域。

此点处梯度为 $[-8.832576; -9.931313]$ , Hessian阵为 $[0.009318 \ 0.025057; 0.025057 \ 0.024819]$

本次迭代的步长为(74.721246, 324.708671), 迭代之后的点为(82.721246, 414.708671)

初始点为(1,40)的迭代: 本次迭代在点(1.000000, 40.000000)迭代之后超出定义域。

此点处梯度为 $[-8.009541; -9.962069]$ , Hessian阵为 $[-1.000098 \ 0.000155; 0.000155 \ -0.000723]$

本次迭代的步长为 $(-10.145665, -13790.138648)$ , 迭代之后的点为 $(-9.145665, -13750.138648)$

初始点为(15,68.69)的迭代: 本次迭代在点(15.000000, 68.690000)迭代之后超出定义域。

此点处梯度为 $[-8.936846; -9.969667]$ , Hessian阵为 $[-0.004162 \ 0.000469; 0.000469 \ 0.000071]$

本次迭代的步长为(7842.777412, 88659.658161), 迭代之后的点为(7857.777412, 88728.348161)

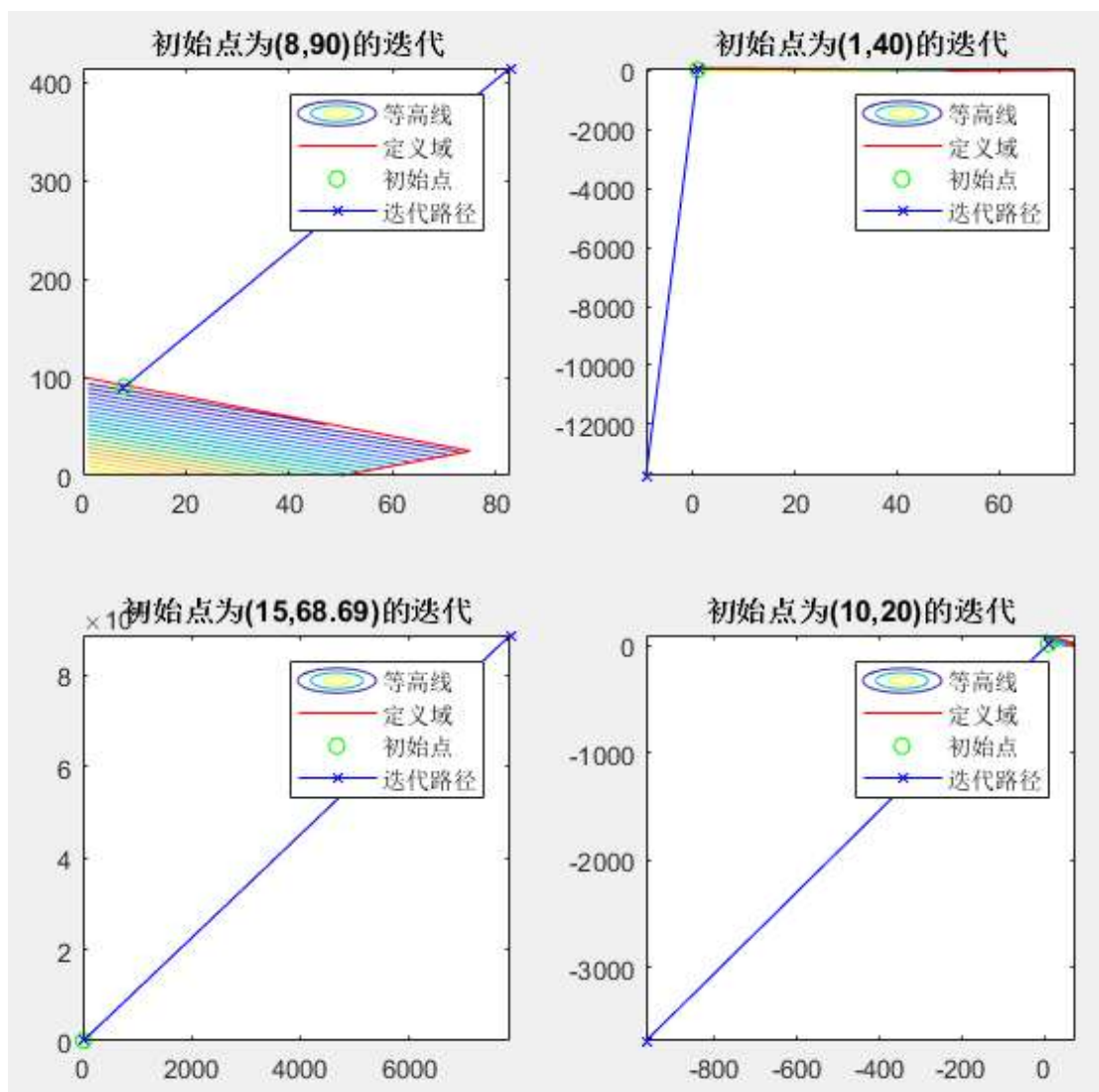
初始点为(10,20)的迭代: 本次迭代在点(10.000000, 20.000000)迭代之后超出定义域。

此点处梯度为 $[-8.915238; -9.931905]$ , Hessian阵为 $[-0.010257 \ 0.000298; 0.000298 \ -0.002757]$

本次迭代的步长为 $(-976.935760, -3707.596260)$ , 迭代之后的点为 $(-966.935760, -3687.596260)$

程序暂停, 输入回车以继续... 输入之后将清空命令和图示





### 3.4.4 $\mu = 0.1$ , 有线搜索

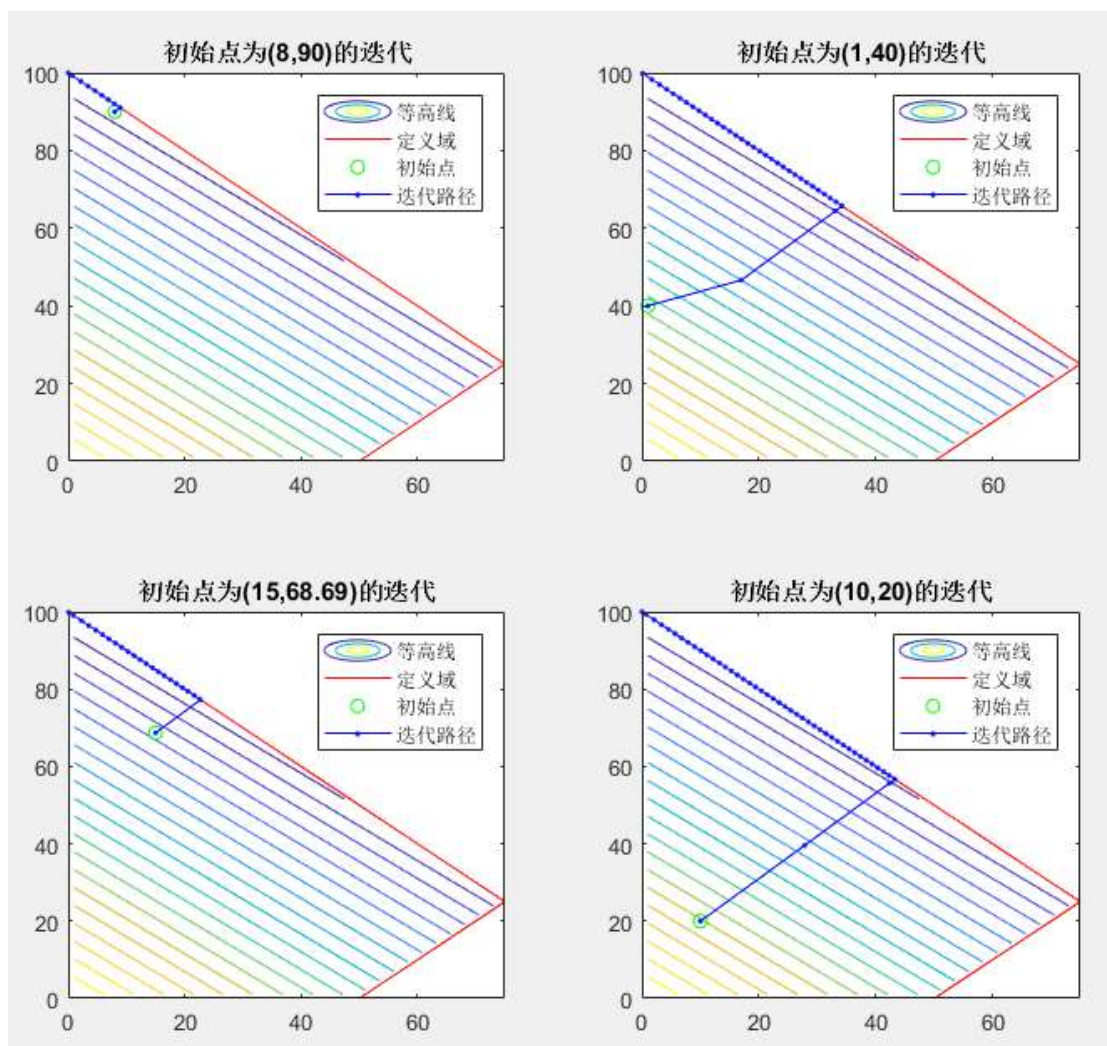
$u=1$ , 有线搜索时牛顿法运行结果:

第1个点, 初始点为(8, 90)经过10次迭代终止在(0.013575, 99.9751)

第2个点, 初始点为(1, 40)经过38次迭代终止在(0.10714, 99.8809)

第3个点, 初始点为(15, 68.69)经过24次迭代终止在(0.072554, 99.9158)

第4个点, 初始点为(10, 20)经过46次迭代终止在(0.022627, 99.9653)



### 3.5. 说明和分析

运行没有线搜索的牛顿法时，会遇到 2 个问题：1，某点处的 Hessian 阵非半正定(甚至负定)，牛顿法产生的牛顿步不是下降方向；2，牛顿步过长，可能导致迭代之后  $f$  上升，甚至可能超出定义域。

加入一维线搜索，一是遇到  $G$  负定的时候，加上一个  $\lambda$  倍的单位阵使得  $G$  正定，其中  $\lambda$  要大于最小特征值的绝对值。二是对于牛顿方向  $s$ ，结合 Armijo 条件和定义域限制迭代确定步长，确保步长不太大也不太小(事实上利用 Armijo 法则，只要设置  $0 < \gamma < 1$ ，从大到小检

验  $\alpha$  是否满足条件，会选择一个较大的满足条件的  $\alpha$ ，就保证了步长不会太小)。  $x := x + \alpha * s$  即为线搜索的迭代点。

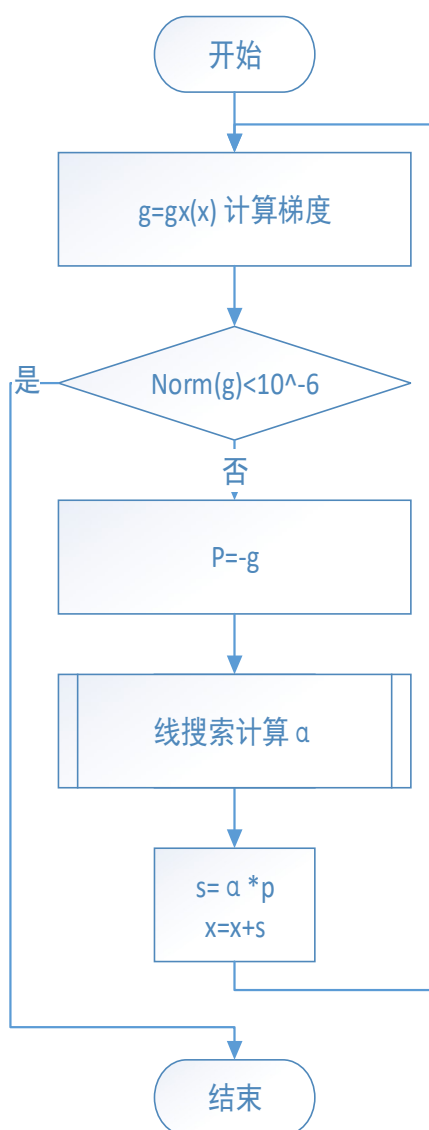
加入了线搜索之后，避免了  $G$  不半正定导致的搜索方向上升，和因步长过大导致的  $f$  增加或者超出定义域的问题，使得  $x$  最终收敛到稳定点。

此题因为有定义域限制，因此终止准则为两次相邻迭代的梯度之差  $\text{norm}(g^{(k+1)} - g^{(k)}) < \delta$ 。此处取  $\delta = 10^{-6}$ 。即迭代之后梯度不发生变化为止。

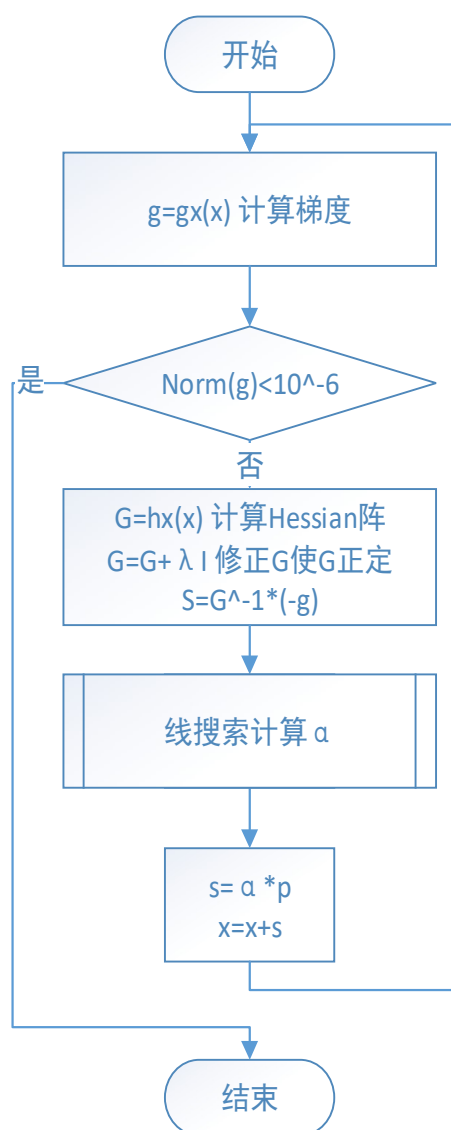
## 4. 习题 5.9

### 4.1. 算法流程图

最速下降法的如下图所示：

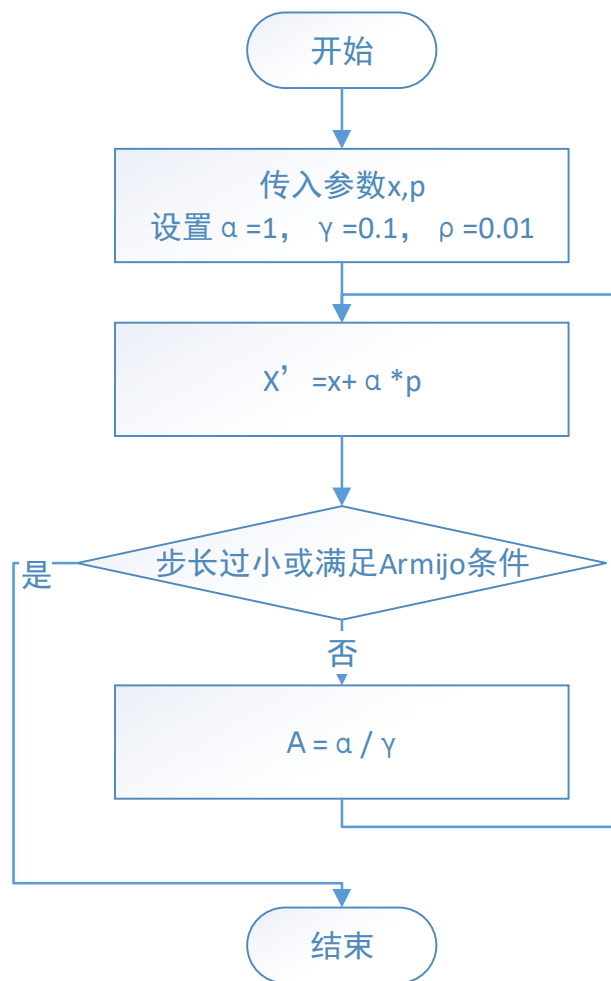


牛顿法的流程图如下图所示：



线搜索子程序的流程图如下图所示：





## 4.2. 重要表达式

梯度公式：

$$\nabla f = g(x) = \begin{bmatrix} 400x_1^3 - 400x_1x_2 + 2x_1 - 2 \\ -200x_1^3 + 200x_2 \end{bmatrix}$$

Hessian 阵：

$$\nabla^2 f = G(x) = \begin{bmatrix} 1200x_1^2 - 400x_2 + 2 & -400x_1 \\ -400x_1 & 200 \end{bmatrix}$$

## 4.3. 参数设置

初始步长  $\alpha = 1$ ;

步长变化率  $\gamma = 0.1$ ;

Armijo 条件中的  $\rho = 0.01$ ;

其中步长变化率是经过多次尝试得到的收敛速度较快的一个

$\rho$	0.01	0.03	0.07	0.1	0.2	0.4	0.7	0.8	0.9
点1线	51981	266	20565	221	10238	12958	15008	15484	17437
点1牛	22	22	22	22	22	22	22	22	22
点2线	58206	690	21992	885	9448	13192	18466	18768	20842
点2牛	393	165	90	72	38	35	35	35	34

## 4.4. 计算结果

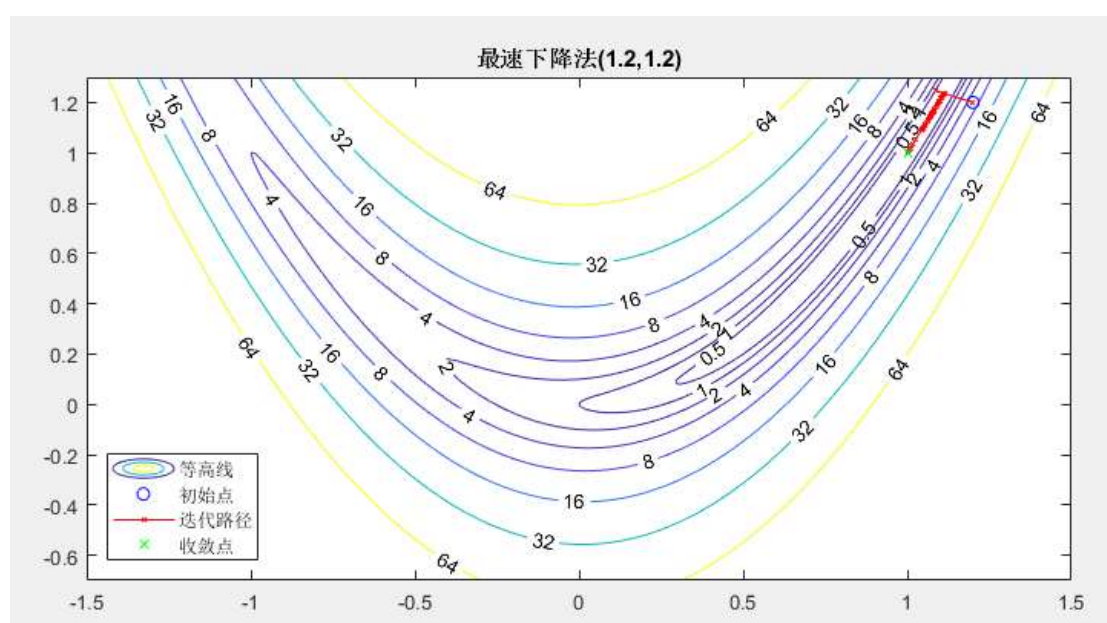
最速下降法第1个点(1.2, 1.2): 迭代次数为221, 最优值为(1, 1):

最速下降法第1个点(1.2, 1.2): 迭代次数为22, 最优值为(1, 1):

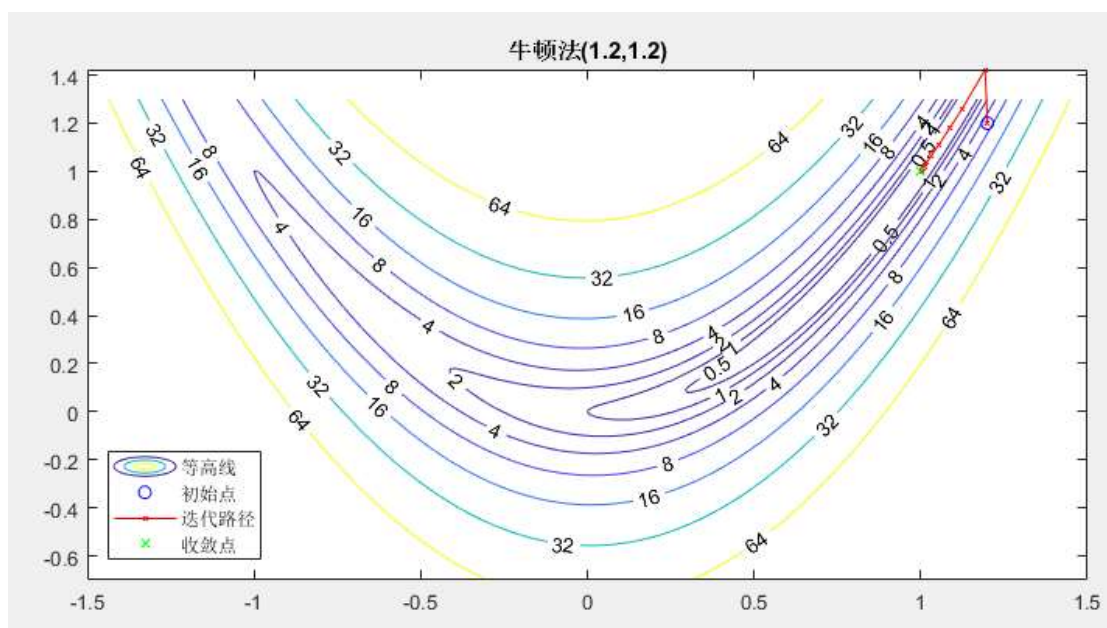
最速下降法第2个点(-1.2, 1): 迭代次数为885, 最优值为(1, 1):

最速下降法第2个点(-1.2, 1): 迭代次数为72, 最优值为(1, 1):

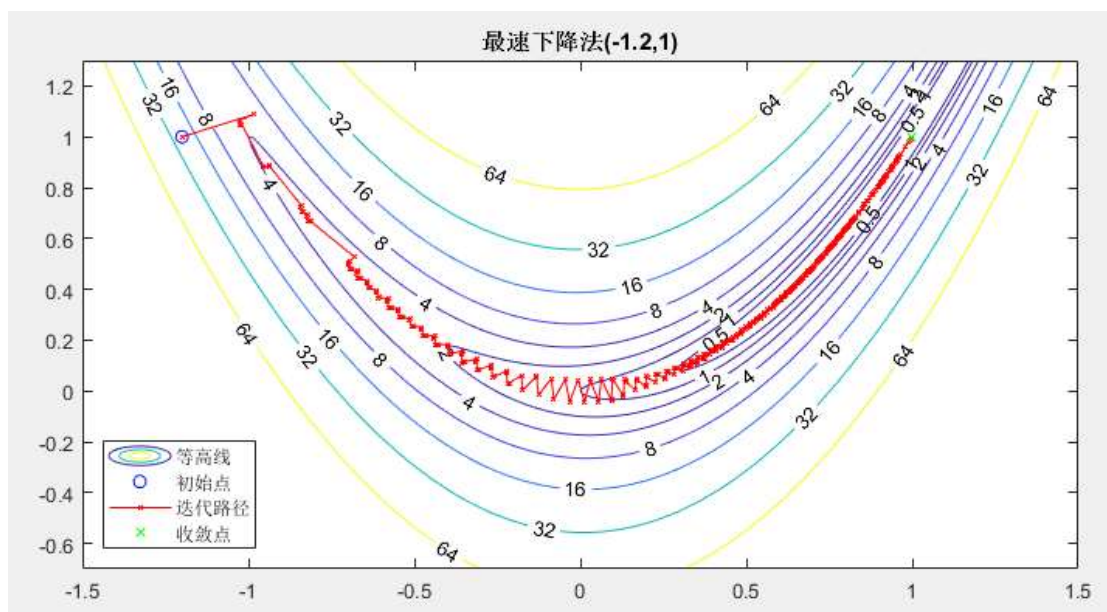
最速下降法在初始点(1.2,1.2)处的迭代路径如下图所示:



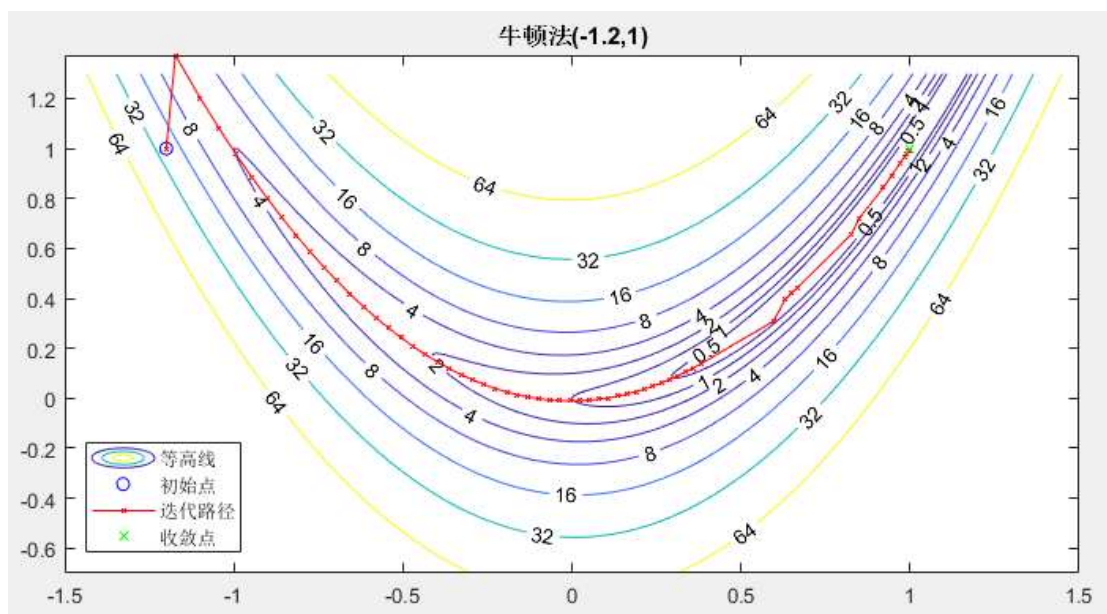
牛顿法在初始点(1.2,1.2)处的迭代路径如下图所示：



最速下降法在初始点(-1.2,1)处的迭代路径如下图所示：



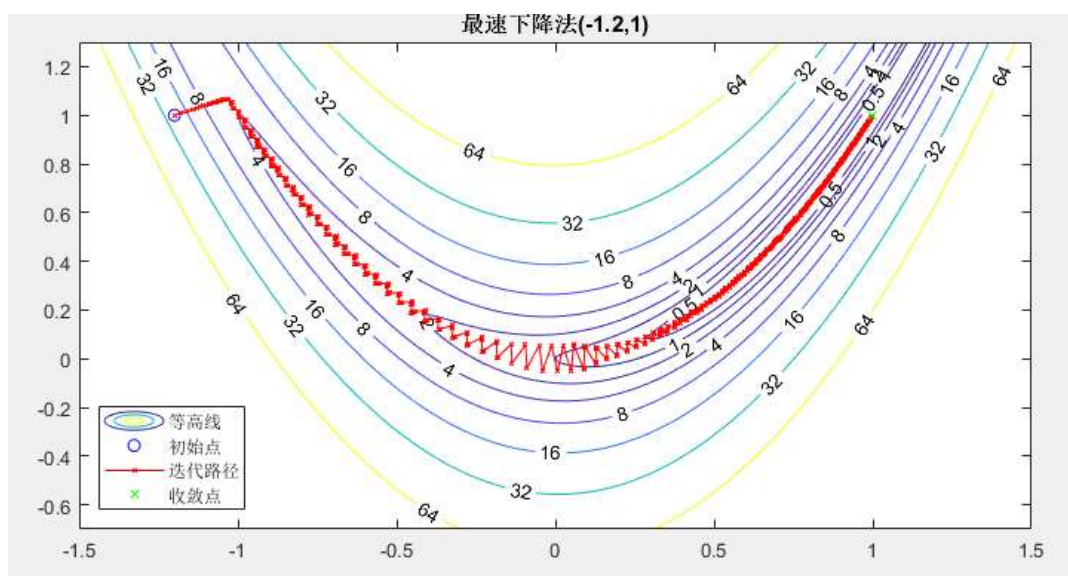
牛顿法在初始点(-1.2,1)处的迭代路径如下图所示：



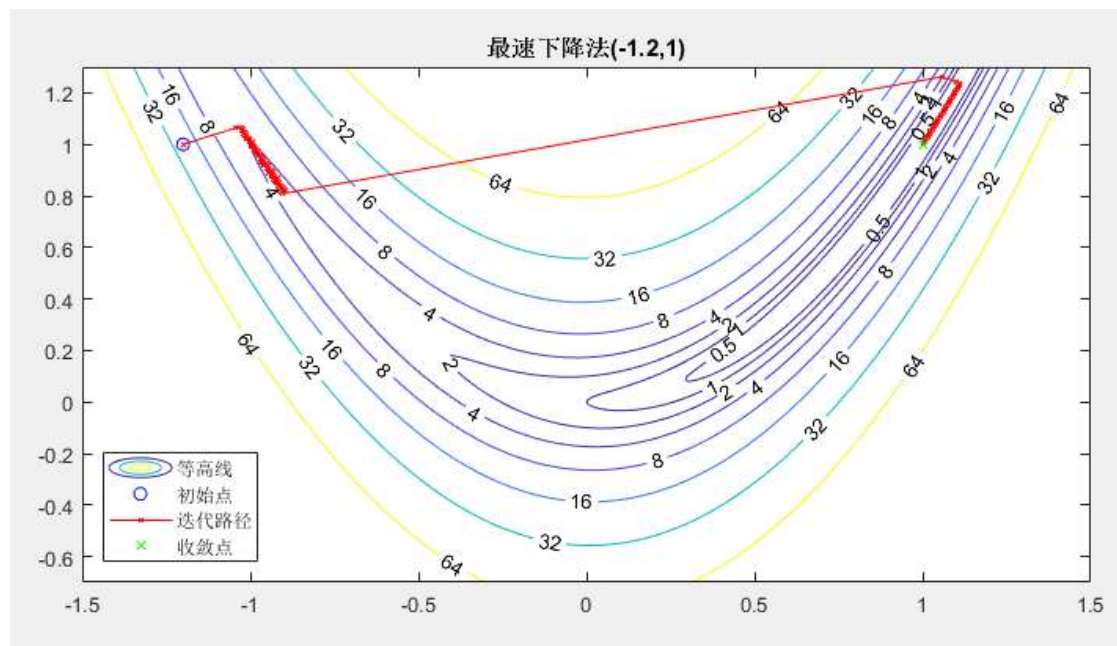
#### 4.5. 说明和分析

由此看出，对于这种复杂的函数，牛顿法的收敛速度大于最速下降法。最优点处的 Hessian 阵  $G = \begin{bmatrix} 802 & -400 \\ -400 & 200 \end{bmatrix}$ ，其特征值为 0.04 和 100.16,特征值的条件数为 0.9984。因此最速下降法的收敛较慢。牛顿法具有二次收敛性，因此收敛较快。

特别说明的是，对于线搜索  $\gamma$  参数的选取，最速下降法呈现出两种不同的迭代路径；当  $\gamma$  小于 0.2 时，大致是这样的



而当  $\gamma > 0.3$  时，最速下降法却呈现出如下图所示的迭代路径，即有一步出现了很大幅度的跳跃。

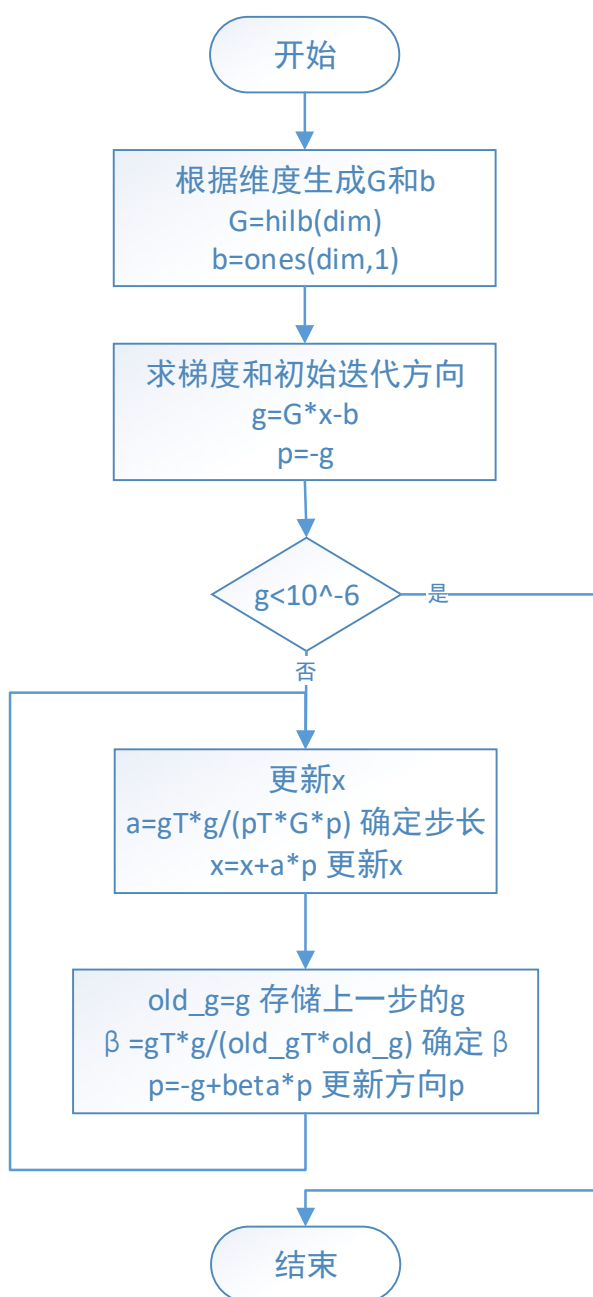




## 5. 习题 5.19

### 5.1. 算法流程图

共轭梯度法的算法流程图如下图所示：



## 5.2. 参数选择

初始点选择为 $(0,0,\dots,0)^T$

## 5.3. 重要表达式

当维度为  $n$  时的 Hessian 阵:

$$G = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \dots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \dots & \frac{1}{n} & \vdots \\ \frac{1}{3} & \vdots & \ddots & \vdots & \frac{1}{2n-3} \\ \vdots & \frac{1}{n} & \dots & \frac{1}{2n-3} & \frac{1}{2n-2} \\ \frac{1}{n} & \dots & \frac{1}{2n-3} & \frac{1}{2n-2} & \frac{1}{2n-1} \end{bmatrix}$$

当维度为  $n$  时，在  $x$  处的梯度为:

$$g = G * x - b, \text{ 其中}$$

$$b = [1 \quad 1 \quad \dots \quad 1 \quad 1]^T$$

## 5.4. 计算结果

当  $n=5$  时: 经过 6 次迭代, 最优解是(5.000000,-120.000000,630.000000,-1120.000000,630.000000)

当  $n=8$  时: 经过 19 次迭代, 最优解是(-8.000005,504.000097,-7560.000380,46199.999870,-138599.997343,216215.995671,-168167.997561,51479.999654)

当  $n=12$  时: 经过 35 次迭代, 最优解是(-9.608955,815.396946,

-16496.560142,135510.323446,-536481.215700,1025399.395714,  
-642578.292665,-657590.976689,804243.884702,663072.549159,  
-1241279.515326,465506.443773)

当  $n=20$  时：经过 66 次迭代，最优解是(-10.974913,1050.929246,  
-23956.279508,220425.673599,-965346.669654,1990103.068626,  
-1252700.007603,-1343474.309128,883233.066644,1687963.950929,  
388212.758340,-1305525.720417,-1710545.719916,-528251.066591,  
1208686.491247,2002890.863867,944594.155587,-1434053.349201,  
-2650953.308676,1887855.722610)

## 5.5. 说明和分析

此题因为维度过高，无法用图来展示。算法很简单，但是存在着一个精度的问题。因此，不仅没有符合共轭梯度法迭代次数约等于  $G$  的特征值的群这一规律，甚至当  $n$  比较大的时候，求出的解并不正确。出现这一现象的原因是希尔伯特矩阵是一种数学变换矩阵，正定，且高度病态（即，任何一个元素发生一点变动，整个矩阵的行列式的值和逆矩阵都会发生巨大变化），病态程度和阶数相关。当  $n$  较大时候，matlab 的精度不足以满足求解该问题的所需要的精度。

因为当  $n$  较大时， $G$  的特征值基本上接近于 0，直接求逆的话求得的解很不精确（matlab 给出的错误是 Matrix is close to singular or badly scaled. Results may be inaccurate.）因此求逆用了如下三种方法：1，matlab 自带的一个 pinv 的方法，2 用奇异值分解计算矩阵  $G$



的逆（若  $G = \text{SVD}^T$ ，则  $G^+ = \text{DVS}^T$ ），3 直接用 matlab 提供的求 hilbert 矩阵逆的方法 `invhilb(G)`。因为函数为二次函数，最优解满足  $Gx=b$ ，所以  $x^*=G^{-1}b$ 。用这个直接求得的  $x^*$  对结果进行验证。

以下是实验证明。

### 5.4.1 $n=5$

$$\lambda(G) = \{0.0000, 0.0003, 0.0114, 0.2085, 1.5671\}$$

特征值集中在四个群中，理论上迭代四次即可完成。事实上，

当  $n=5$  时：

经过6次迭代，最优解是 (5.000000, -120.000000, 630.000000, -1120.000000, 630.000000)  
最优解处梯度的范数为0.000000

经过4次迭代，得到的解是 (-3.806119, 45.656004, -86.466078, -36.193004, 99.241701)  
这个点的梯度的范数为0.004676

四次迭代的解和原问题的解相差较大(大概 5 次迭代可以近似等于共轭梯度法求得的解)

用 `pinv` 计算  $G^+ * b$ ，得到的解是 (5.000000, -120.000000, 630.000000, -1120.000000, 630.000000)  
SVD 解处梯度的范数为0.000000

用奇异值分解直接计算  $G^+ * b$ ，得到的解是 (5.000000, -120.000000, 630.000000, -1120.000000, 630.000000)  
SVD 解处梯度的范数为0.000000

用求逆公式计算  $(G^+)^{-1} * b$ ，得到的解是 (5.000000, -120.000000, 630.000000, -1120.000000, 630.000000)  
求逆得到的解处梯度的范数为0.000000

当  $n=5$  时，共轭梯度法和直接求所计算出的结果基本一致，且所得解处梯度都近似为 0；

### 5.4.2 $n=8$

$$\lambda(G) = \{0.0000, 0.0001, 0.0015, 0.0262, 0.2981, 1.6959\}$$

特征值集中在五个群中，理论上迭代五次即可完成。事实上，

当n=8时:

经过19次迭代, 最优解是(-8.000005, 504.000097, -7560.000380, 46199.999870, -1385  
最优解处梯度的范数为0.000000

经过5次迭代, 得到的解是(4.577301, -72.421073, 199.840775, -9.995385, -174.85813;  
这个点的梯度的范数为0.003480

五次迭代的解和原问题的解相差较大(大概 15 次迭代可以近似  
等于共轭梯度法求得的解)

用pinv计算 $G^+ * b$ , 得到的解是(-8.000000, 503.999987, -7559.999806, 46199.998871, -138599.99  
pinv解处梯度的范数为0.000000

用奇异值分解直接计算 $G^+ * b$ , 得到的解是(-8.000000, 503.999987, -7559.999806, 46199.998871, -  
SVD解处梯度的范数为0.000000

用求逆公式计算 $(G^T G)^{-1} * G^T b$ , 得到的解是(-8.000000, 504.000000, -7560.000000, 46200.000000, -1  
求逆得到的解处梯度的范数为0.000000

当 n=5 时, 共轭梯度法和直接求所计算出的结果基本一致, 且  
所得解处梯度都近似为 0; 由于解过长, 这里的图只给出了前面一  
部分, 运行程序可以得到全部的结果。

### 5.4.3 n=12

$$\lambda(G) = \{0.0000, 0.0002, 0.0037, 0.0447, 0.3803, 1.7954\}$$

特征值集中在五个群中, 理论上迭代五次即可完成。事实上,

当n=12时:

经过35次迭代, 最优解是(-9.608955, 815.396946, -16496.560142, 135510.323446, -536481.215700, 10  
最优解处梯度的范数为0.000001

经过5次迭代, 得到的解是(4.422331, -51.513922, 85.918784, 52.578369, -14.798884, -64.343400, -83  
这个点的梯度的范数为0.011496

五次迭代的解和原问题的解相差较大(大概 29 次迭代可以近似  
等于共轭梯度法求得的解)

用pinv计算 $G*b$ ，得到的解是(10.787581, -1204.243615, 32555.707206, -369337.836105, 2153968.3  
pinv解处梯度的范数为0.000474

用奇异值分解直接计算 $G*b$ ，得到的解是(10.787581, -1204.243615, 32555.707206, -369337.836105,  
SVD解处梯度的范数为0.001149

用求逆公式计算 $(G^{-1})*b$ ，得到的解是(-12.000000, 1716.000000, -60060.000000, 900900.000000, -7  
求逆得到的解处梯度的范数为0.000000

当  $n=12$  时，共轭梯度法，pinv 和求逆库函数得出的解差别较大，不过 svd 和 pinv 这两种方法得出的解差别不大。求逆公式直接求出的解处梯度依然为 0，其余的均接近于 0。因为目标函数为凸函数，因此应该为计算时的精度问题。我个人更倾向于认为用求逆公式得出的解是正确的，因为它的梯度完全为 0

#### 5.4.4 $n=20$

$\lambda(G)=\{0.0000, 0.0001, 0.0009, 0.0090, 0.0756, 0.4870, 1.9071\}$

特征值集中在六个群中，理论上迭代六次即可完成。事实上，

当 $n=20$ 时：

经过66次迭代，最优解是(-10.974913, 1050.929246, -23956.279508, 220425.673599, -965346.6  
最优解处梯度的范数为0.000001

经过6次迭代，得到的解是(-5.298580, 74.742756, -172.762104, -34.143552, 94.624648, 139.521  
这个点的梯度的范数为0.009170

六次迭代的解和原问题的解相差较大(大概 52 次迭代可以近似等于共轭梯度法求得的解)

用pinv计算 $G*b$ ，得到的解是(12.031818, -1449.886957, 42076.353249, -505370.203033, 3037045.672363, -964  
pinv解处梯度的范数为0.000264

用奇异值分解直接计算 $G*b$ ，得到的解是(12.031818, -1449.886957, 42076.353249, -505370.203033, 3037045.67  
SVD解处梯度的范数为0.000264

用求逆公式计算 $(G^{-1})*b$ ，得到的解是(-20.013184, 7984.000000, -789944.000000, 34323200.000000, -8246804.  
求逆得到的解处梯度的范数为42572045163.466431

当  $n=20$  时，得出的结果相差更大了，甚至于直接调用 matlab 库函数给出的求逆函数得出的解，梯度都是无穷大。事实上，当  $n=20$  时， $\text{hilb}(20) * \text{invhilb}(20)$  接近 0 矩阵，而不是单位阵。

由此可得，受精度限制，用这些简单的方法，matlab 无法解出该问题的解。

## 6. 习题 5.27

### 6.1. 题目说明

本题是一个数据拟合问题，即寻找最好的  $\mathbf{x}$ ，使得

$$\varphi(t_i, \mathbf{x}) = d_i, i = 1, 2, \dots, 6$$

误差最小。该问题可以改写成如下的数学规划问题：

$$\text{minimize} \quad \sum_{i=1}^6 (\varphi(t_i, \mathbf{x}) - d_i)^2$$

其中

$$\varphi(t_i, \mathbf{x}) = (1 - \frac{x_1}{x_2} t_i)^{\frac{1}{cx_1} - 1}, c = 96.05$$

容易得到，该问题的定义域为

$$\begin{cases} \frac{x_1}{x_2} t_i < 1 \\ x_1 \neq 0 \\ x_2 \neq 0 \end{cases}$$

直接解该问题，难度过大，需要做一次变换，令

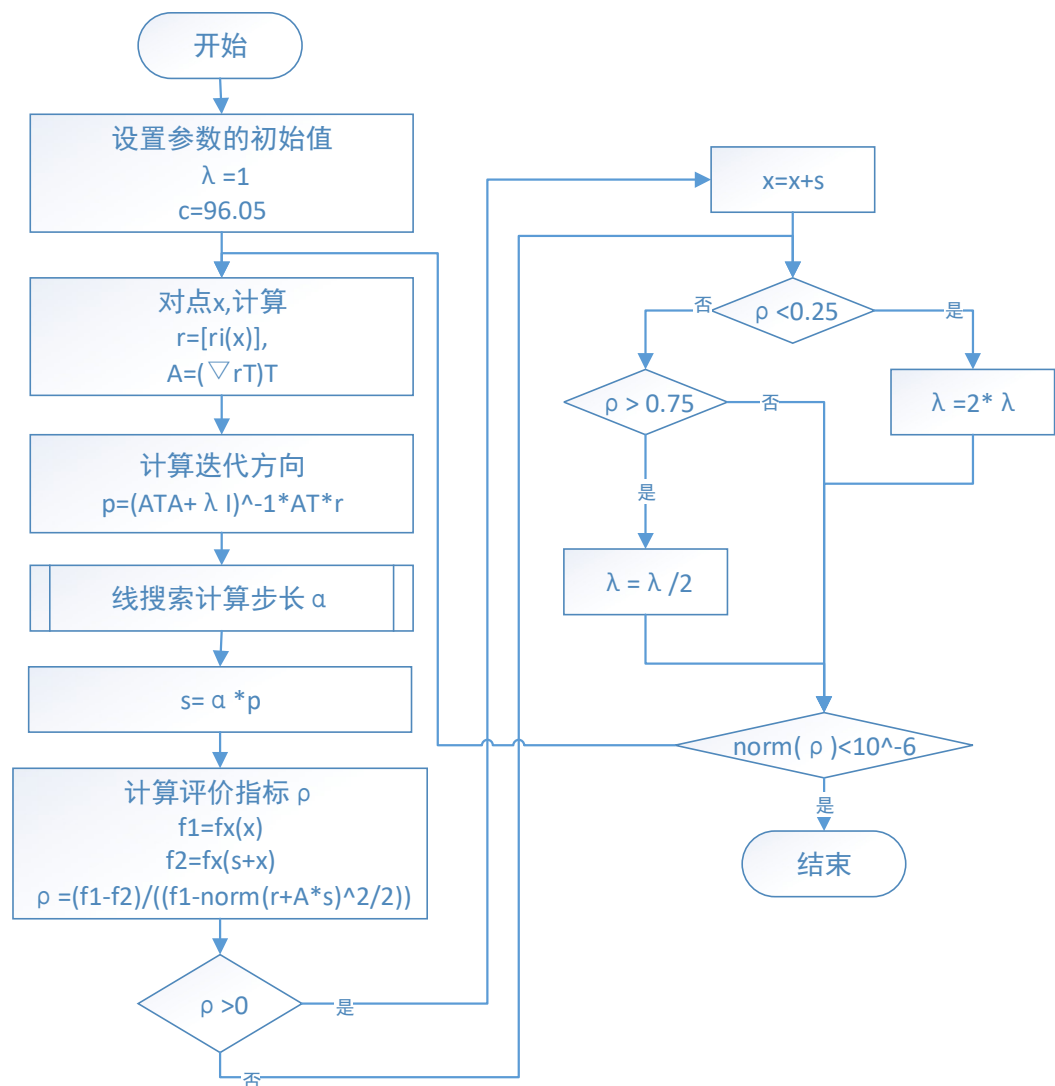
$$\begin{cases} y_1 = \frac{x_1}{x_2} \\ y_2 = \frac{1}{cx_1} - 1 \end{cases}$$

则原问题可以转化为：

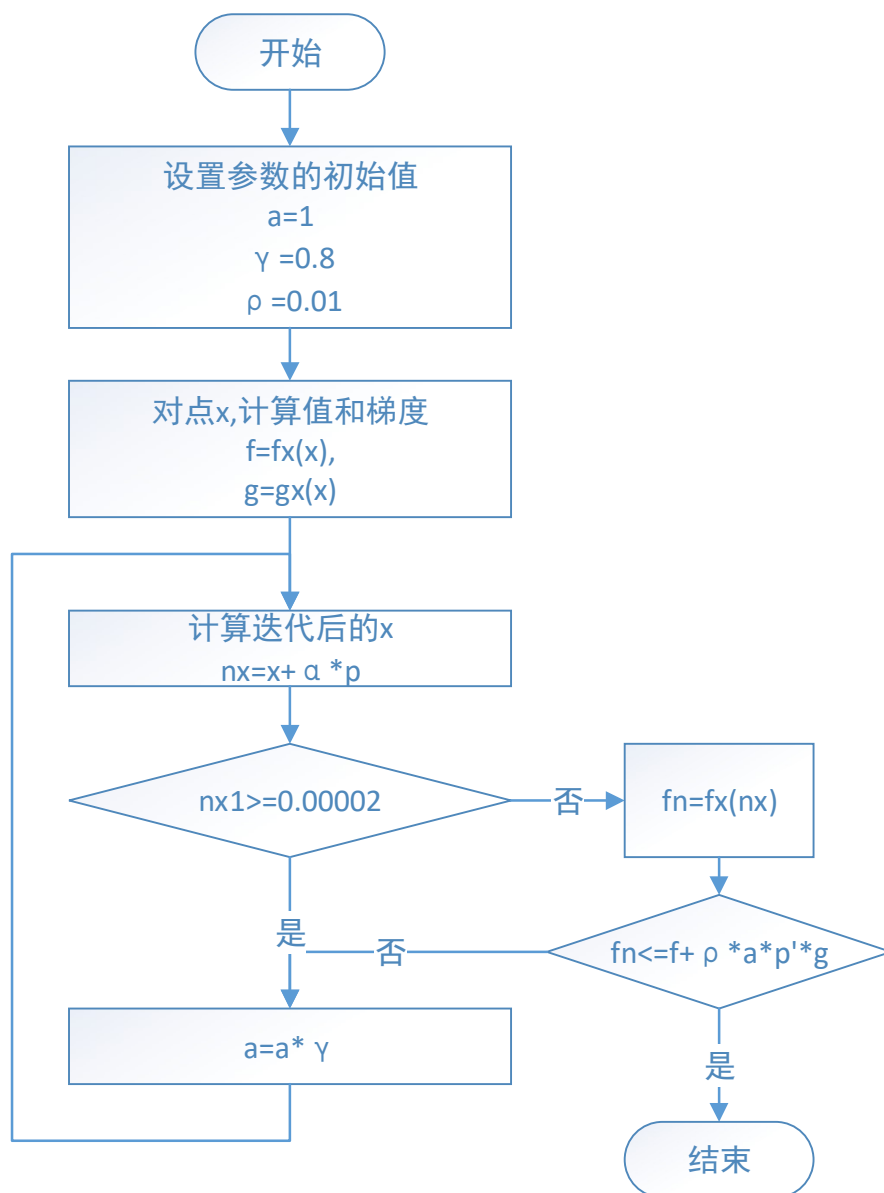
$$\begin{aligned}
& \text{minimize} && \sum_{i=1}^6 (\varphi(t_i, y) - d_i)^2 \\
& \text{subject to} && \varphi(t_i, y) = (1 - t_i y_1)^{y_2} \\
& && y_1 < 1/50000 \\
& && y_2 \text{ 无限制}
\end{aligned}$$

## 6.2. 算法流程图

对于进行坐标转换之后的输入初始点  $x$ ，高斯-牛顿法的流程图如下图所示：



线搜索子程序的流程图如下图所示：



### 6.3. 重要表达式

余量

$$r = (1 - x_1 t).x^2 - d$$

Hessian 阵:

$$G \approx A^T * A$$

其中,

$$A^T = \nabla r^T$$

$$= \begin{bmatrix} -t_1 * x_2 * (1 - t_1 * x_1)^{x_2-1} & ((1 - t_1) * x_1)^{x_2} * \log(1 - t_1 * x_1) \\ \dots & \dots \\ -t_6 * x_2 * (1 - t_6 * x_1)^{x_2-1} & ((1 - t_6) * x_1)^{x_2} * \log(1 - t_6 * x_1) \end{bmatrix}$$

## 6.4. 参数设置

### 6.4.1 初始点的选择

由于  $0 < d < 1$ , 且  $d$  随着  $t$  的增大而减小, 由  $d \approx \varphi(t_i, y) = (1 - t_i y_1)^{y_2}$  可得:

i: 若  $1 - t_i y_1 < 1$ , 则  $y_2 > 1$ , 即  $1/50000 > y_1 > 0, y_2 > 1$

ii: 若  $1 - t_i y_1 > 1$ , 则  $y_2 < 0$ , 即  $y_1 < 0, y_2 < 0$

对情况 i, 可选择初始点  $y = (1/100000, 2.126)$ , 即  $x = (0.00333, 333)$

对情况 ii, 可选择初始点  $y = (-1/50000, -2.041)$ , 即  $x = (-0.01; 500)$

### 6.4.2 $\gamma$ 的选择

$\gamma$	0.01	0.1	0.2	0.4	0.6	0.8	0.9	0.99
点 1	14	14	14	14	14	14	14	14
点 2	2	3	3	3	3	3	3	3

$\gamma$  参数对迭代次数影响微乎其微, 对结果也没有任何影响。结合实验, 最终选择了  $\gamma = 0.01$  作为参数。



## 6.5. 计算结果

对于两个不同的初始点，计算结果分别是

第一个迭代点的经过14次迭代，迭代路径为:

(0.00333, 333)→(0.0033246, 308.8351)→(0.0033217, 306.3131)→(0.0033202, 306.0804)→  
(0.0033194, 306.061)→(0.0033191, 306.0604)→(0.0033189, 306.0609)→(0.0033188, 306.0612)→  
(0.0033187, 306.0614)→(0.0033187, 306.0615)→(0.0033187, 306.0615)→(0.0033187, 306.0616)→  
(0.0033187, 306.0616)→(0.0033187, 306.0616)→(0.0033187, 306.0616).

第一个迭代点的解为(0.0033187, 306.0616), 最优值为0.023997

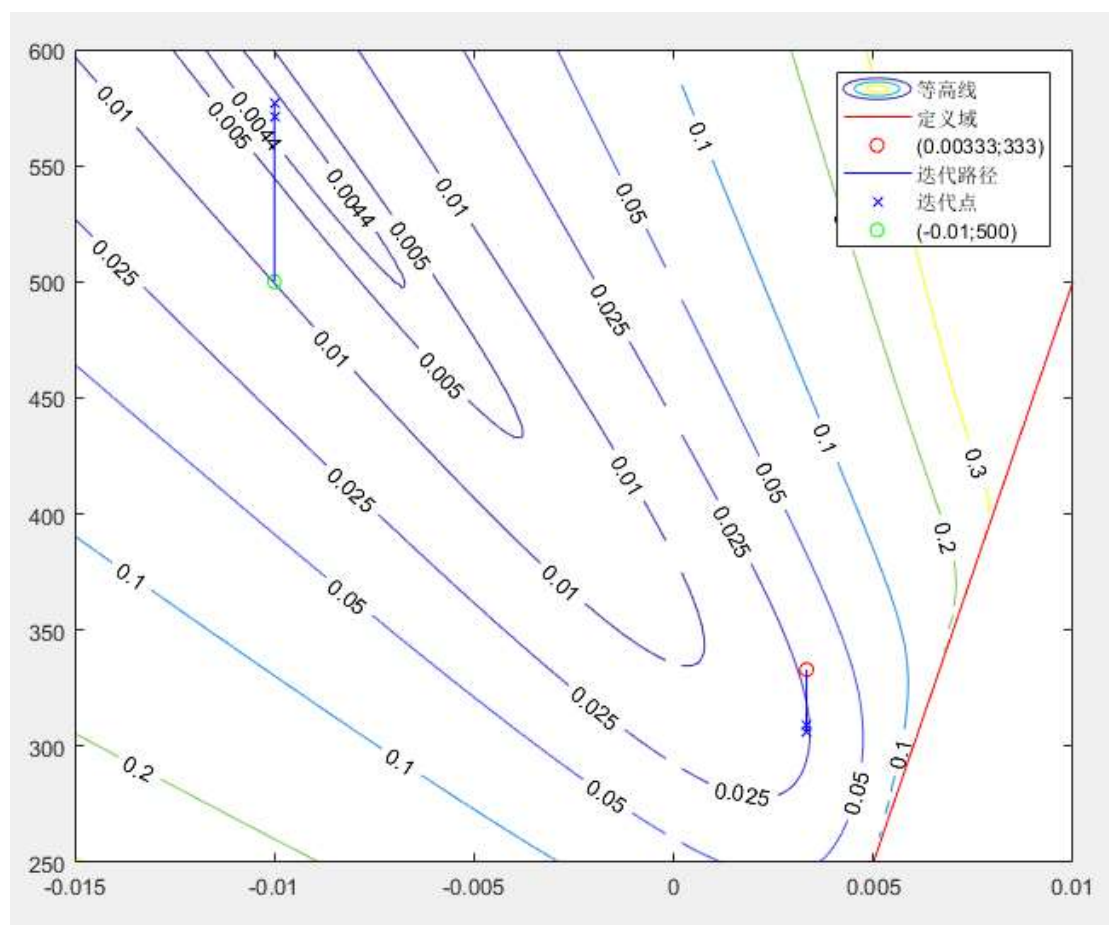
第二个迭代点的经过2次迭代，迭代路径为:

(-0.01, 500)→(-0.0099993, 576.9417)→(-0.009999, 571.1731).

第二个迭代点的解为(-0.009999, 571.1731), 最优值为0.004281

所以该问题的最优解为(-0.00999,571.1731), 最优值为 0.004281

等高线和求解过程的迭代路径如下图所示，图中的等高线和迭代路径为原始点，即没有进行坐标变换之前的图像和路径。



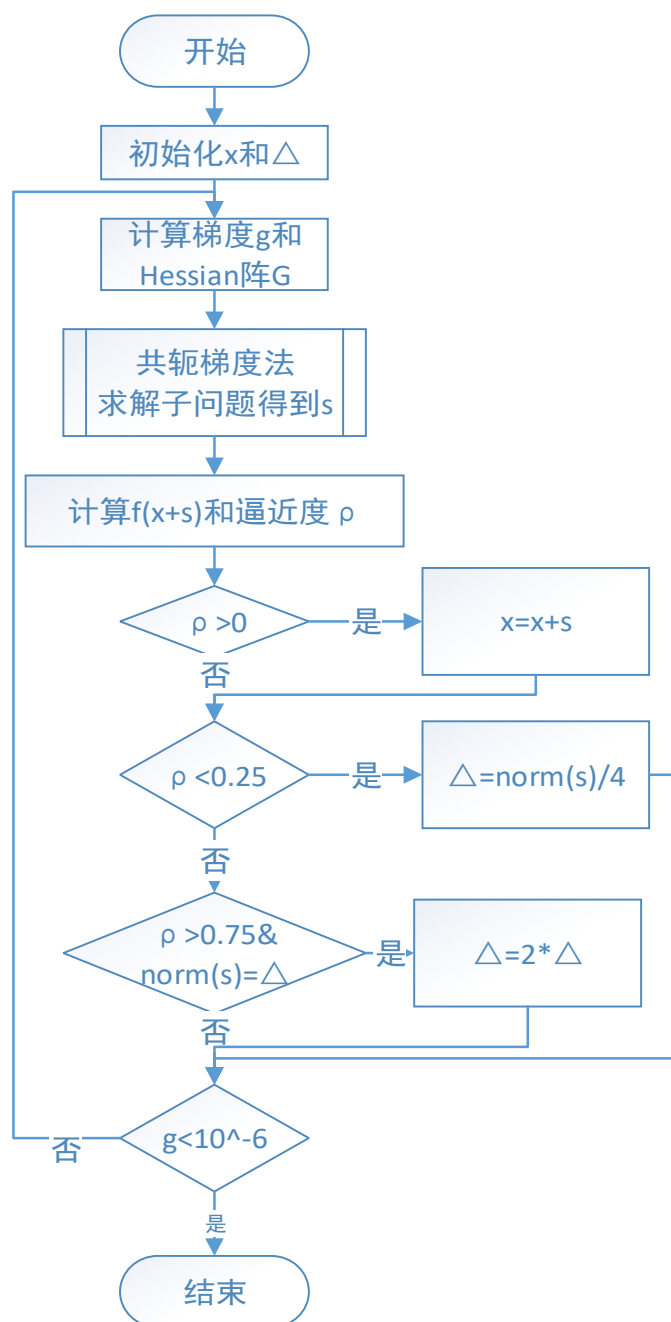
## 6.6. 说明和分析

本题用共轭梯度法解，直接计算比较困难，所以要做一次坐标变换。需要注意的是定义域问题，如果不加定义域限制，计算出的结果将会是复数，致使迭代无法继续。

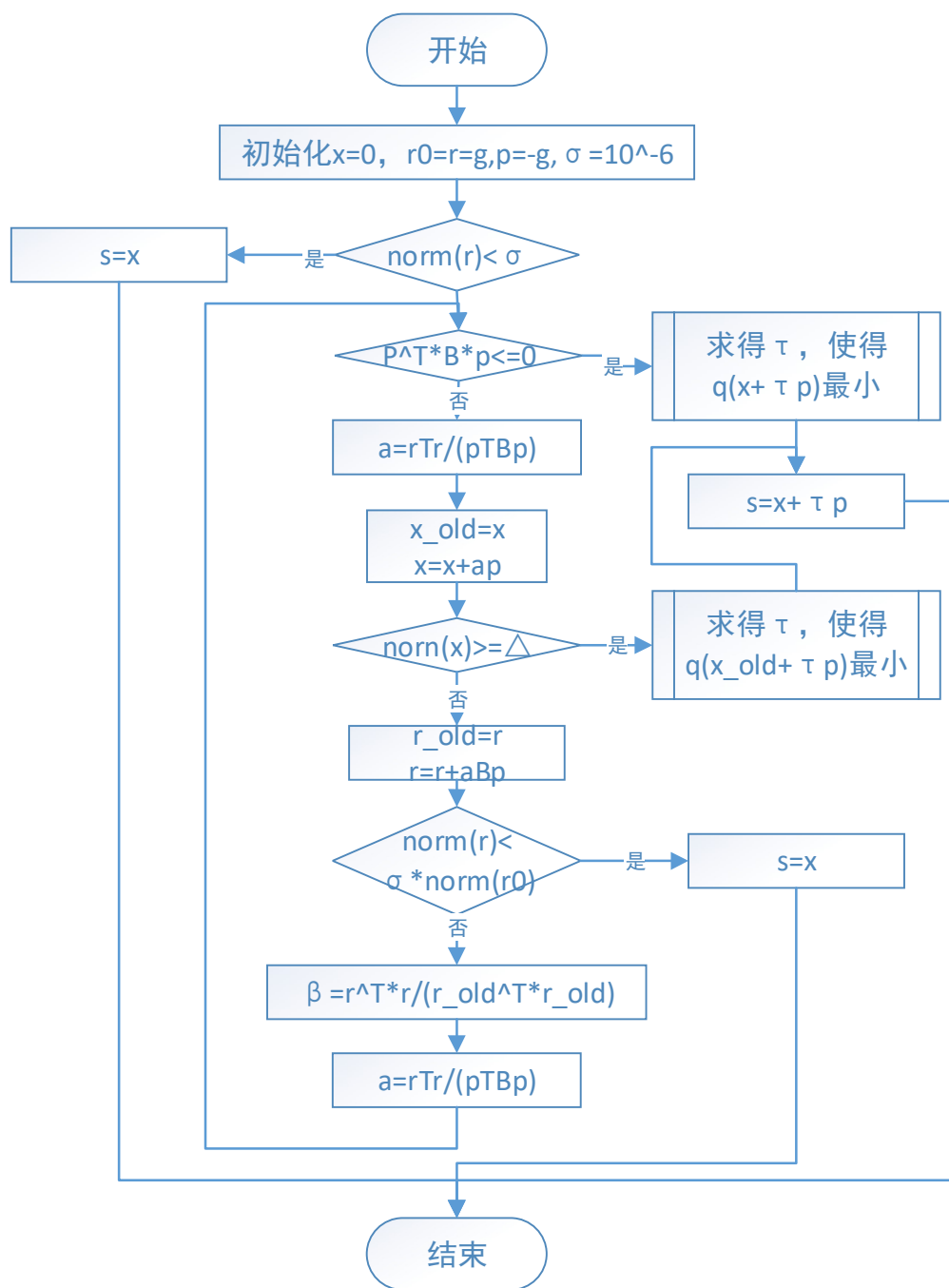
## 7. 习题 6.4

### 7.1. 算法流程图

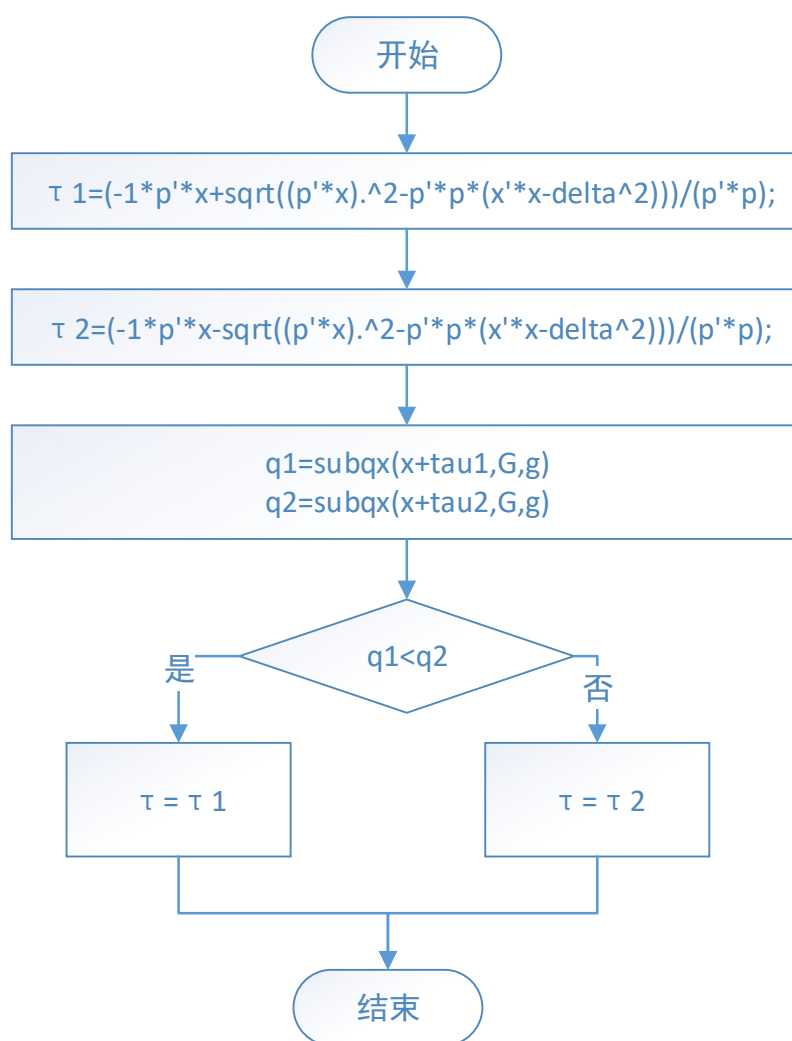
信赖域法的流程图如图所示：



子问题的 steihaug 共轭梯度法的流程图如图所示：



求  $\tau$  的子程序：



## 7.2. 重要表达式

在点  $x$  处的梯度：

$$g = \begin{bmatrix} 40x_1^3 - 40x_1x_2 + 2x_1 - 2 \\ 20x_2 - 20x_1^2 \\ \vdots \\ 40x_{2n-1}^3 - 40x_{2n-1}x_{2n} + 2x_{2n-1} - 2 \\ 20x_{2n} - 20x_{2n-1}^2 \end{bmatrix}$$

在点  $x$  处的 Hessian 阵

G

$$= \begin{bmatrix} 120x_1^2 - 40x_2 + 2 & -40x_1 & \dots & 0 \\ -40x_1 & 20 & & \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 120x_{2n-1}^2 - 40x_{2n} + 2 & -40x_{2n-1} \\ & & -40x_{2n-1} & 20 \end{bmatrix}$$

在  $x$  处的信赖域子问题为:

$$\text{minimize } y = \frac{1}{2}s^T Gs + g^T s$$

$$\text{subject to } s^T s = \Delta^2$$

### 7.3. 计算结果

当  $n=10$  时,

当 $n=10$ 时:

第1次迭代: 子问题经过1次迭代, 达到信赖域边界停止迭代。

第2次迭代: 子问题经过1次迭代, 达到信赖域边界停止迭代。

第3次迭代: 子问题经过1次迭代, 达到信赖域边界停止迭代。

第4次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

第5次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

第6次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

第7次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

第8次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

第9次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

第10次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

最优解是:

1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00

1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00

当  $n=50$  时

当 $n=50$ 时:

第1次迭代: 子问题经过1次迭代, 达到信赖域边界停止迭代。

第2次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

第3次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

第4次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

第5次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

第6次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

第7次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

第8次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

第9次迭代: 子问题经过2次迭代, 满足停止条件停止迭代。

最优解是:

1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00  
1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00  
1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00  
1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00  
1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00  
1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00  
1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00  
1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00  
1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00  
1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00

## 7.4. 说明和分析

### 7.5.1 子问题最多迭代两次:

无论  $n$  取多少, 当初始点为全 0 时, 或者更进一步地, 当初始点  $x$  满足  $x_1 = x_3 = \cdots = x_{2n-1}$ ,  $x_2 = x_4 = \cdots = x_{2n}$  时, 总会有  $g_1 = g_3 = \cdots = g_{2n-1}$ ,  $g_2 = g_4 = \cdots = g_{2n}$ , 进一步地, 每次迭代方向总满足  $p_1 = p_3 = \cdots = p_{2n-1}$ ,  $p_2 = p_4 = \cdots = p_{2n}$ , 则迭代产生的下一个点  $x = x + ap$  也满足  $x_1 = x_3 = \cdots = x_{2n-1}$ ,  $x_2 = x_4 = \cdots = x_{2n}$ 。即只要初始点满足  $x_1 = x_3 = \cdots = x_{2n-1}$ ,  $x_2 = x_4 = \cdots = x_{2n}$ , 经过该信赖域法和 steihaug 求解信赖域子问题的迭代过程的所有的点均会满足  $x_1 = x_3 = \cdots = x_{2n-1}$ ,  $x_2 = x_4 = \cdots = x_{2n}$ 。

若  $x$  满足上述条件，则  $x$  处的迭代方向  $p$  和  $g$  也满足  $p_1 = p_3 = \dots = p_{2n-1}$ ,  $p_2 = p_4 = \dots = p_{2n}$ ,  $g_1 = g_3 = \dots = g_{2n-1}$ ,  $g_2 = g_4 = \dots = g_{2n}$ 。所以整个子问题都可以分解为  $n$  个完全相同的子问题对，因此子问题最优解处也必满足  $x_1 = x_3 = \dots = x_{2n-1}$ ,  $x_2 = x_4 = \dots = x_{2n}$ ，最优解处的 Hessian 阵  $G$  必定满足

$$G_{1,1} = G_{3,3} = \dots = G_{2n-1,2n-1}$$

$$G_{2,2} = G_{4,4} = \dots = G_{2n,2n}$$

$$G_{1,2} = G_{2,1} = G_{3,4} = G_{4,3} \dots = G_{2n-1,2n} = G_{2n,2n-1}$$

因此， $G$  最多有 2 个不同的特征值。由于子问题是用共轭梯度法来解的，根据共轭梯度法的特点，子问题最多迭代 2 次（子问题最优解处特征值的群数量）即停止迭代。

## 7.5.2 逼近度 $\rho$ 的计算

根据公式

$$\rho = \frac{\delta f^{(k)}}{\delta q^{(k)}} = \frac{f^{(k)} - f(x^{(k)} + s^{(k)})}{f^{(k)} - q^{(k)}(s^{(k)})}$$

其中

$$q^{(k)}(s) = f^{(k)} + g^{(k)T} s + \frac{1}{2} s^T G s$$

但是在实际应用的过程中，最优化  $q(s)$  不需要加上常数  $f^{(k)}$ ，即  $f(x)$  在子问题迭代起点的值  $f(x^{(k)})$ 。所以如果  $q(s)$  在计算时候没有加上常数的部分，那么逼近度的计算公式应该修改为

$$\rho = \frac{\delta f^{(k)}}{\delta q^{(k)}} = \frac{f(x^{(k)} + s^{(k)}) - f^{(k)}}{q^{(k)}(s^{(k)})}$$



其中

$$q^{(k)}(s) = g^{(k)T} s + \frac{1}{2} s^T G s$$

即直接用 $q(s)$ 来表示预计下降量。真实下降量的表示不变。