《最优化方法》课程的实践环节2

选题一（课本习题）

[1. 习题5.6 5](#_Toc482443364)

[1.1. 算法流程图 5](#_Toc482443365)

[1.2. 重要表达式 6](#_Toc482443366)

[1.3. 计算结果 6](#_Toc482443367)

[1.4. 说明和分析 8](#_Toc482443368)

[2. 习题5.7 9](#_Toc482443369)

[2.1. 算法流程图 9](#_Toc482443370)

[2.2. 重要表达式 9](#_Toc482443371)

[2.3. 计算结果 10](#_Toc482443372)

[2.4. 说明和分析 13](#_Toc482443373)

[3. 习题5.8 15](#_Toc482443374)

[3.1. 算法流程图 15](#_Toc482443375)

[3.2. 重要表达式 15](#_Toc482443376)

[3.3. 参数设置 15](#_Toc482443377)

[3.4. 计算结果 16](#_Toc482443378)

[3.4.1 μ=1，无线搜索 16](#_Toc482443379)

[3.4.2 μ=1，有线搜索 17](#_Toc482443380)

[3.4.3 μ=0.1，无线搜索 18](#_Toc482443381)

[3.4.4 μ=0.1，有线搜索 19](#_Toc482443382)

[3.5. 说明和分析 20](#_Toc482443383)

[4. 习题5.9 21](#_Toc482443384)

[4.1. 算法流程图 21](#_Toc482443385)

[4.2. 重要表达式 21](#_Toc482443386)

[4.3. 参数设置 21](#_Toc482443387)

[4.4. 计算结果 21](#_Toc482443388)

[4.5. 说明和分析 21](#_Toc482443389)

[5. 习题5.19 22](#_Toc482443390)

[5.1. 算法流程图 22](#_Toc482443391)

[5.2. 重要表达式 22](#_Toc482443392)

[5.3. 参数设置 22](#_Toc482443393)

[5.4. 计算结果 22](#_Toc482443394)

[5.5. 说明和分析 22](#_Toc482443395)

[6. 习题5.27 22](#_Toc482443396)

[6.1. 算法流程图 22](#_Toc482443397)

[6.2. 重要表达式 22](#_Toc482443398)

[6.3. 参数设置 22](#_Toc482443399)

[6.4. 计算结果 22](#_Toc482443400)

[6.5. 说明和分析 22](#_Toc482443401)

[7. 习题6.4 23](#_Toc482443402)

[7.1. 算法流程图 23](#_Toc482443403)

[7.2. 重要表达式 23](#_Toc482443404)

[7.3. 参数设置 23](#_Toc482443405)

[7.4. 计算结果 23](#_Toc482443406)

[7.5. 说明和分析 23](#_Toc482443407)

# 习题5.6

## 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



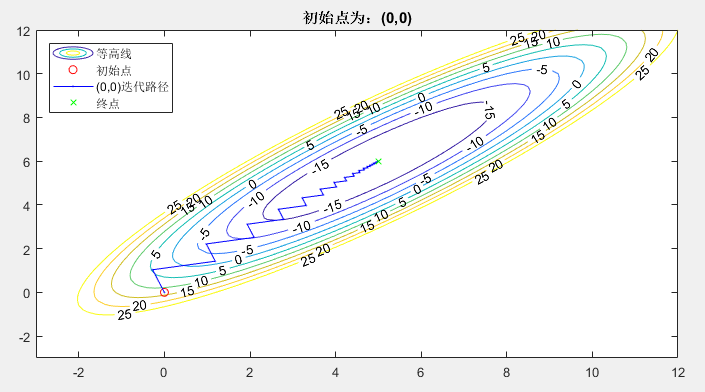
## 重要表达式

梯度公式：

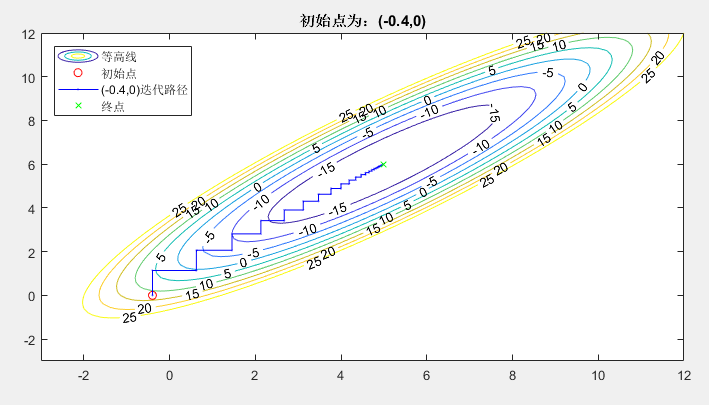
Hessian阵：

## 计算结果

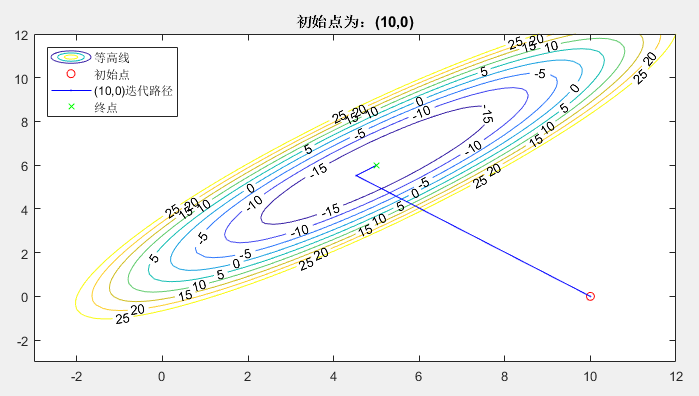
第一个点 (0,0) 的迭代路径如下图所示



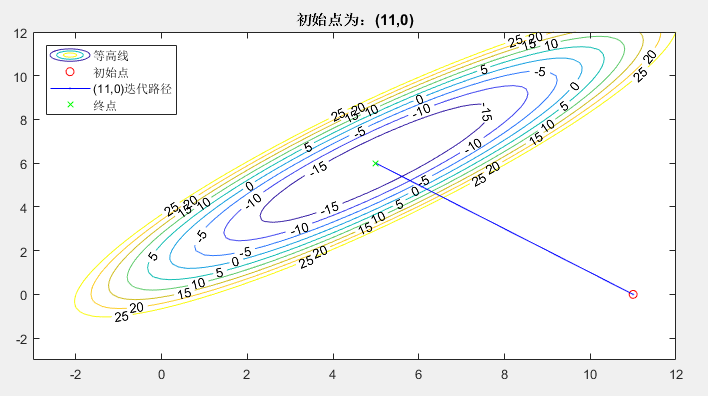
第二个点 (-0.4,0) 的迭代路径如下图所示



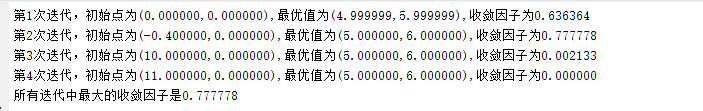
第三个点 (10,0) 的迭代路径如下图所示



第四个点 (10,0) 的迭代路径如下图所示

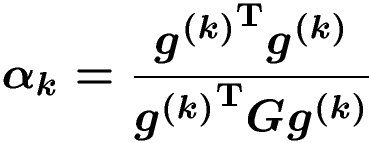


最终的结果如下图所示：



## 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



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。最优点为x=(5,6)，在最优点的Hession阵G为

故G的特征值为

G在二范数下的条件数为

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

# 习题5.7

## 算法流程图



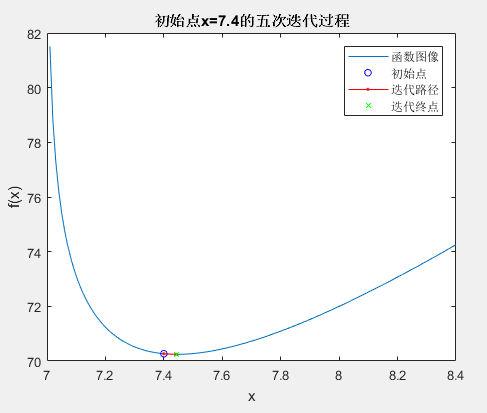
## 重要表达式

梯度公式：

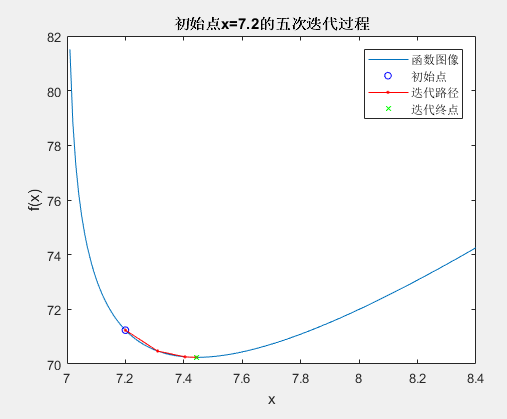
Hessian阵：

## 计算结果

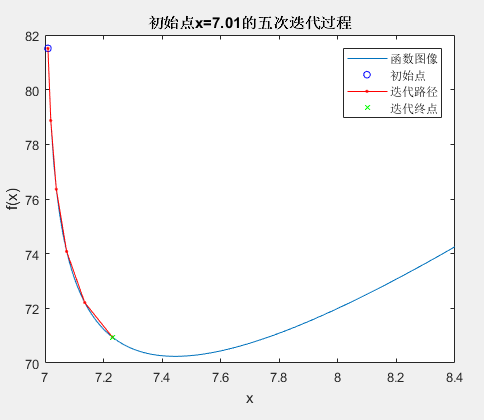
第一个点x=7.40的5步迭代结果：



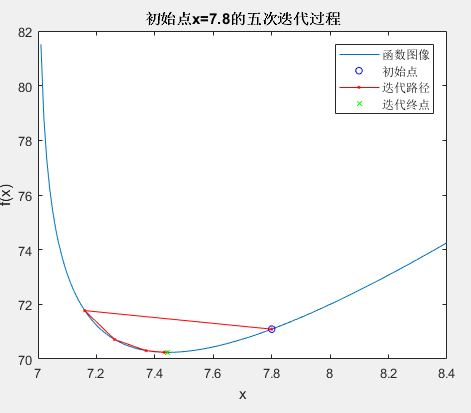
第二个点x=7.20的5步迭代结果：



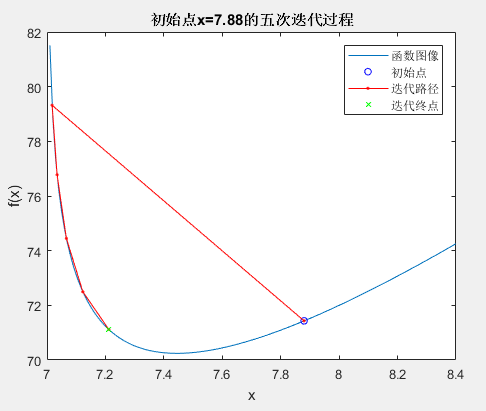
第三个点x=7.01的5步迭代结果：



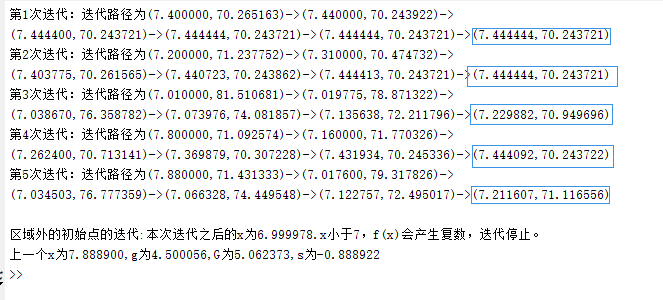
第四个点x=7.80的5步迭代结果：



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：



## 说明和分析

由函数

可以得到，f(x)的定义域为(7,+∞),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)>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：

因为迭代之后的步长需要在定义域之内，因此有

即

解之得

因此只有，迭代产生的x=x+s才会在定义域之内。

，

因此在区间内的任一点，迭代之后的点还在这个区间之内（更严格来说，迭代之后落在区间内，即(7,7.4444)内）。

**结论：**

当时，，因此s>0，古x’=x+s>x。即，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。

# 习题5.8

## 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：



## 重要表达式

梯度公式：

Hessian阵：

## 参数设置

若G负定，则

G=G+(0.5-min{λ})\*I

线搜索：

初始步长α=1

步长变化率γ=0.9

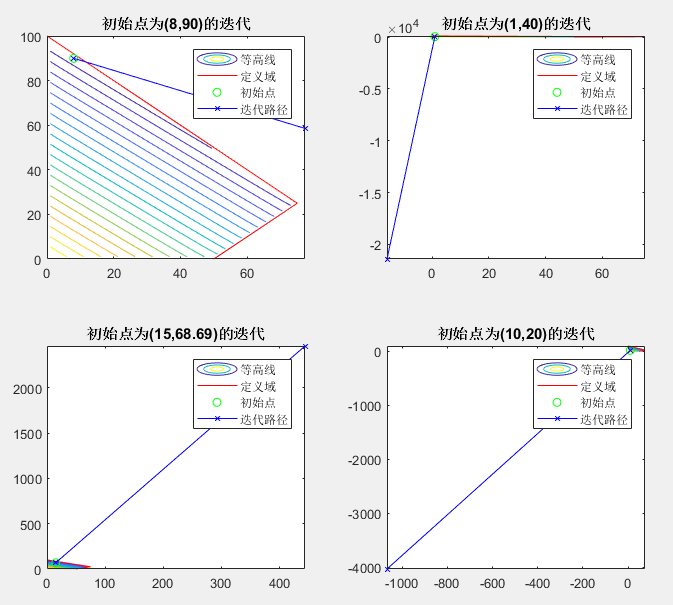
Armijo条件中ρ=0.01

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。

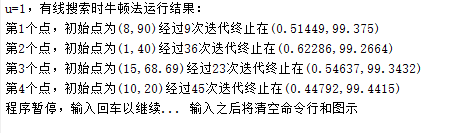
## 计算结果

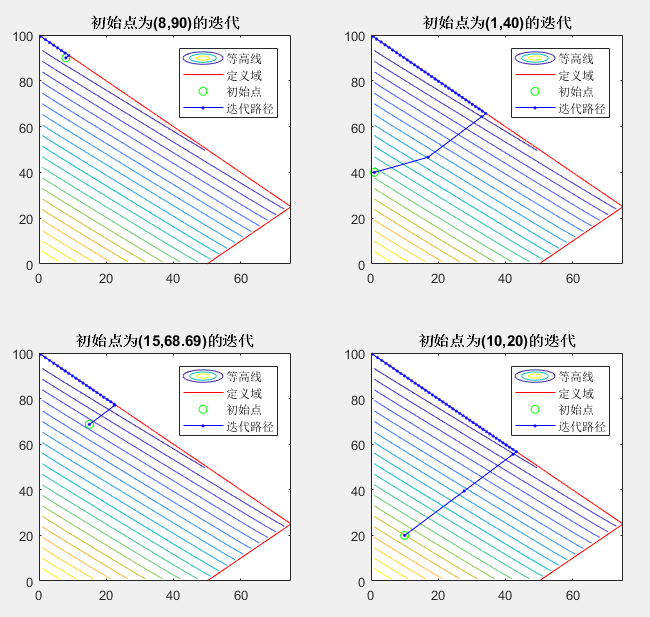
### 3.4.1 μ=1，无线搜索





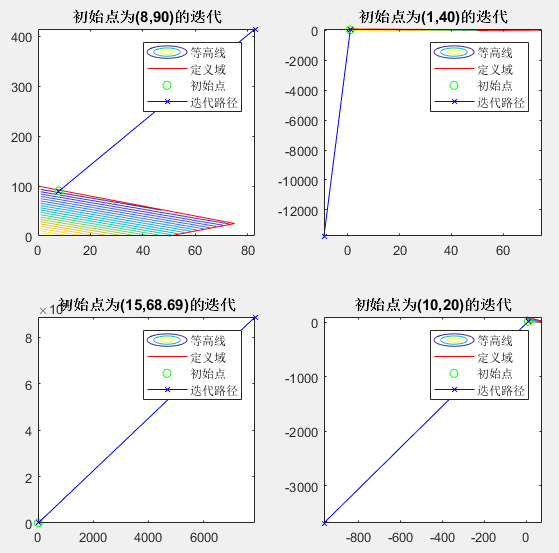
### 3.4.2 μ=1，有线搜索



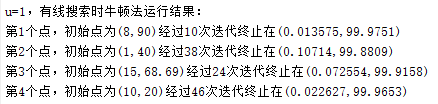


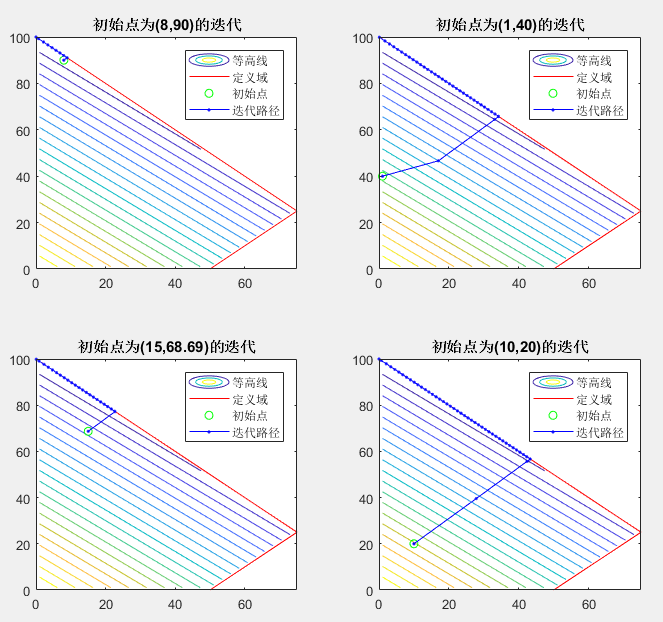
### 3.4.3 μ=0.1，无线搜索





### 3.4.4 μ=0.1，有线搜索





## 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λ倍的单位阵使得G正定，其中λ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<γ<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+α\*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。此处取。即迭代之后梯度不发生变化为止。

# 习题5.9

## 算法流程图

## 重要表达式

## 参数设置

## 计算结果

## 说明和分析

# 习题5.19

## 算法流程图

## 重要表达式

## 参数设置

## 计算结果

## 说明和分析

# 习题5.27

## 算法流程图

## 重要表达式

## 参数设置

## 计算结果

## 说明和分析

# 习题6.4

## 算法流程图

## 重要表达式

## 参数设置

## 计算结果

## 说明和分析