《最优化方法》课程的实践环节2

选题一（课本习题）

目录

[1. 习题5.6 5](#_Toc482541077)

[1.1. 算法流程图 5](#_Toc482541078)

[1.2. 重要表达式 6](#_Toc482541079)

[1.3. 计算结果 6](#_Toc482541080)

[1.4. 说明和分析 8](#_Toc482541081)

[2. 习题5.7 10](#_Toc482541082)

[2.1. 算法流程图 10](#_Toc482541083)

[2.2. 重要表达式 10](#_Toc482541084)

[2.3. 计算结果 11](#_Toc482541085)

[2.4. 说明和分析 14](#_Toc482541086)

[3. 习题5.8 16](#_Toc482541087)

[3.1. 算法流程图 16](#_Toc482541088)

[3.2. 重要表达式 17](#_Toc482541089)

[3.3. 参数设置 18](#_Toc482541090)

[3.4. 计算结果 18](#_Toc482541091)

[3.4.1 μ=1，无线搜索 18](#_Toc482541092)

[3.4.2 μ=1，有线搜索 19](#_Toc482541093)

[3.4.3 μ=0.1，无线搜索 20](#_Toc482541094)

[3.4.4 μ=0.1，有线搜索 21](#_Toc482541095)

[3.5. 说明和分析 22](#_Toc482541096)

[4. 习题5.9 24](#_Toc482541097)

[4.1. 算法流程图 24](#_Toc482541098)

[4.2. 重要表达式 25](#_Toc482541099)

[4.3. 参数设置 25](#_Toc482541100)

[4.4. 计算结果 26](#_Toc482541101)

[4.5. 说明和分析 28](#_Toc482541102)

[5. 习题5.19 30](#_Toc482541103)

[5.1. 算法流程图 30](#_Toc482541104)

[5.2. 重要表达式 31](#_Toc482541105)

[5.3. 计算结果 31](#_Toc482541106)

[5.4. 说明和分析 32](#_Toc482541107)

[5.4.1 n=5 33](#_Toc482541108)

[5.4.2 n=8 33](#_Toc482541109)

[5.4.3 n=12 34](#_Toc482541110)

[5.4.4 n=20 35](#_Toc482541111)

[6. 习题5.27 36](#_Toc482541112)

[6.1. 算法流程图 36](#_Toc482541113)

[6.2. 重要表达式 36](#_Toc482541114)

[6.3. 参数设置 36](#_Toc482541115)

[6.4. 计算结果 36](#_Toc482541116)

[6.5. 说明和分析 36](#_Toc482541117)

[7. 习题6.4 36](#_Toc482541118)

[7.1. 算法流程图 36](#_Toc482541119)

[7.2. 重要表达式 36](#_Toc482541120)

[7.3. 参数设置 36](#_Toc482541121)

[7.4. 计算结果 36](#_Toc482541122)

[7.5. 说明和分析 36](#_Toc482541123)

# 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



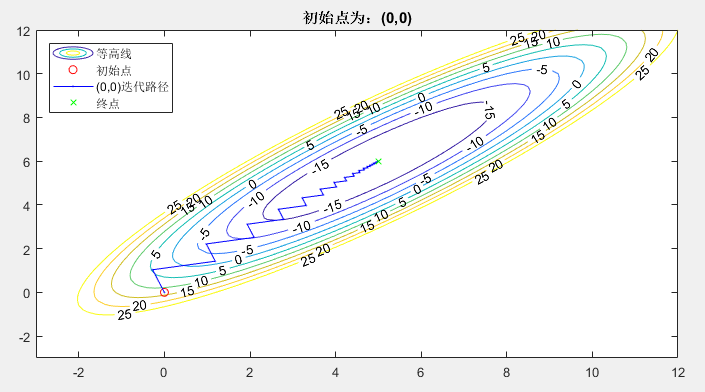
## 重要表达式

梯度公式：

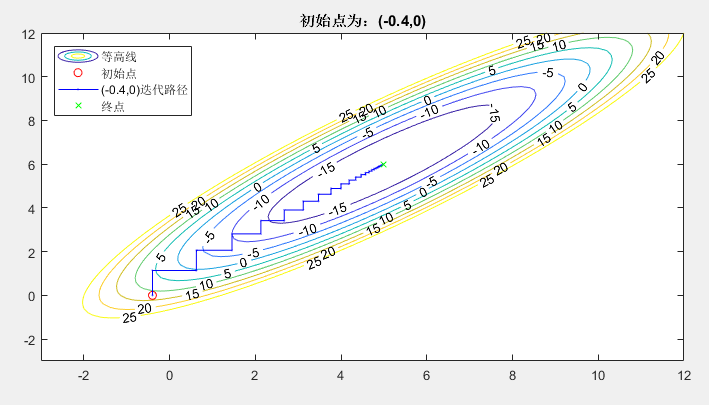
Hessian阵：

## 计算结果

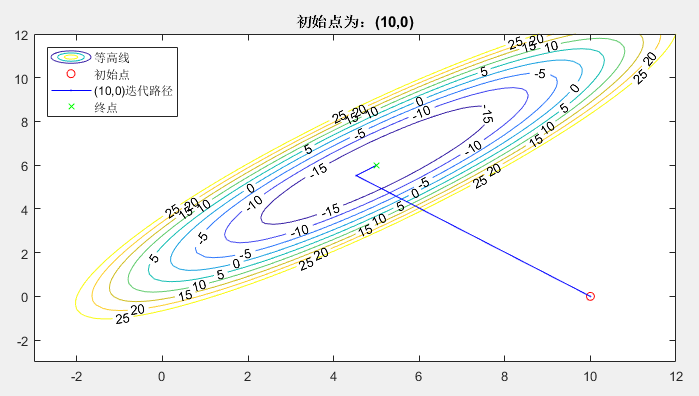
第一个点 (0,0) 的迭代路径如下图所示



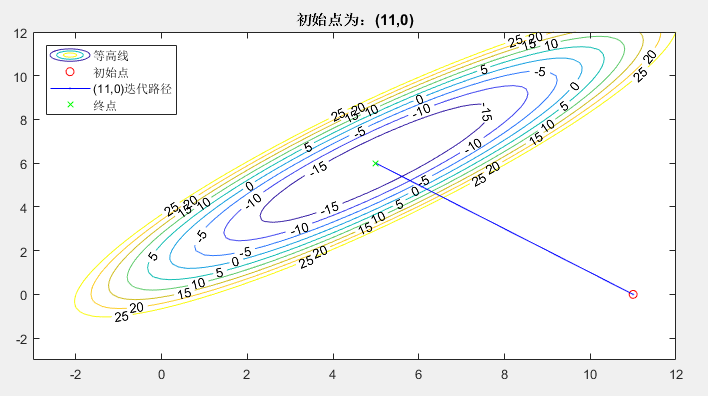
第二个点 (-0.4,0) 的迭代路径如下图所示



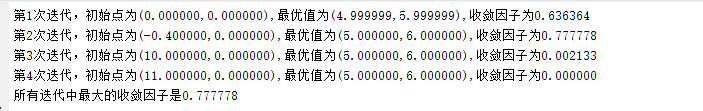
第三个点 (10,0) 的迭代路径如下图所示



第四个点 (10,0) 的迭代路径如下图所示

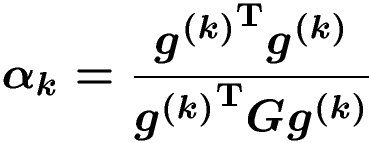


最终的结果如下图所示：



## 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



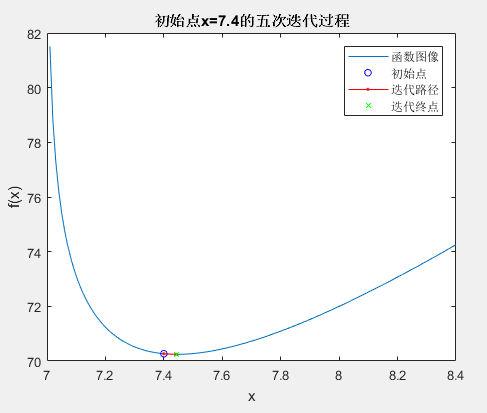
## 重要表达式

梯度公式：

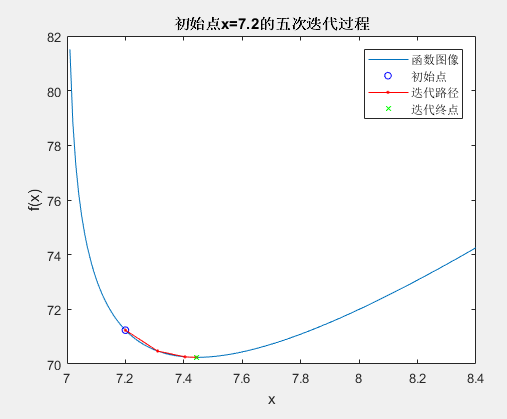
Hessian阵：

## 计算结果

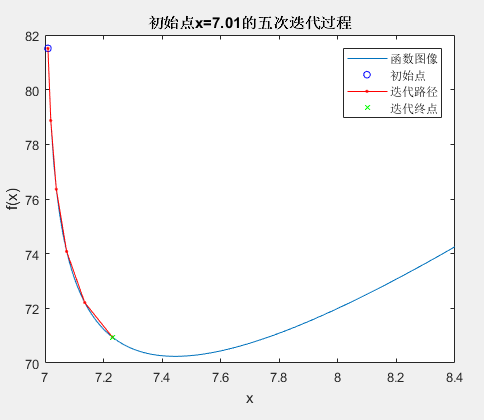
第一个点x=7.40的5步迭代结果：



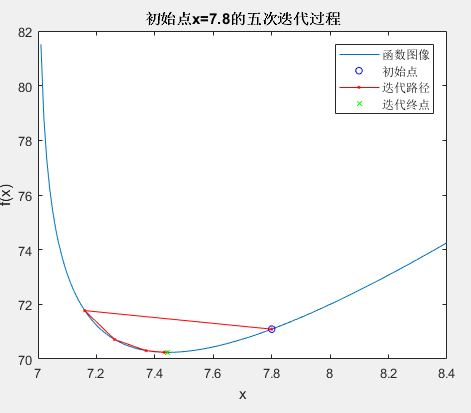
第二个点x=7.20的5步迭代结果：



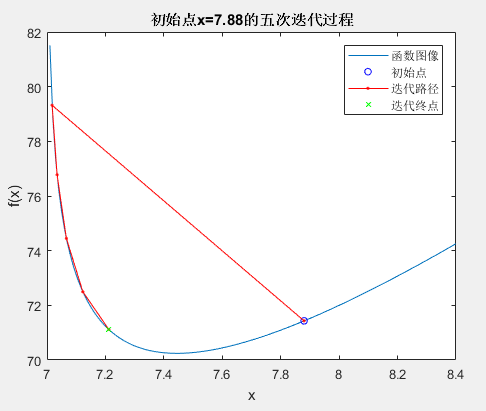
第三个点x=7.01的5步迭代结果：



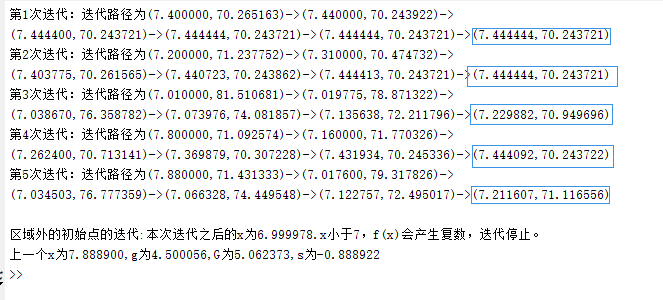
第四个点x=7.80的5步迭代结果：



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

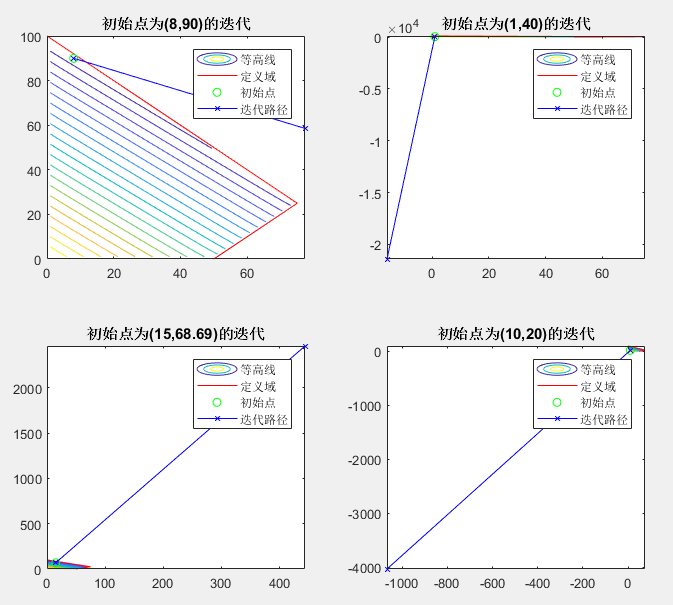
Armijo条件中ρ=0.01

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。

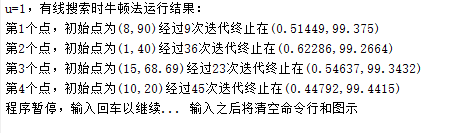
## 计算结果

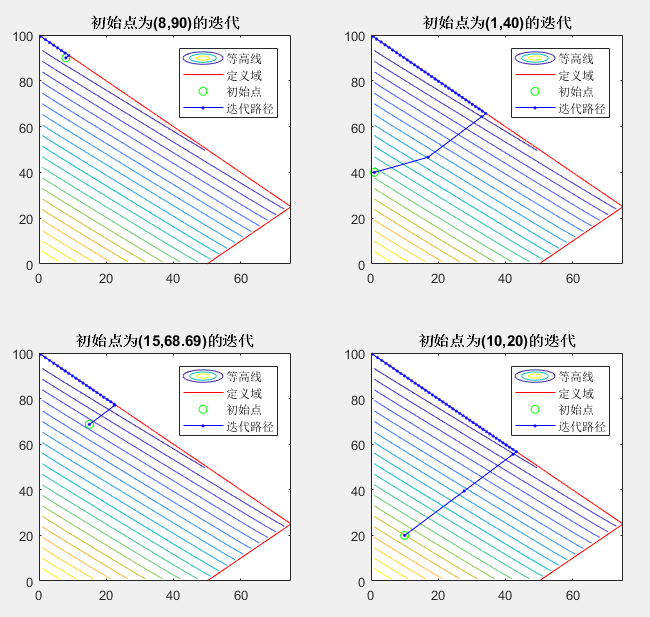
### 3.4.1 μ=1，无线搜索





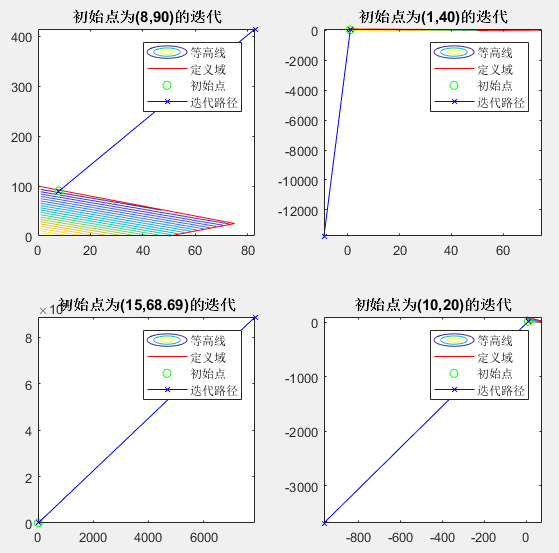
### 3.4.2 μ=1，有线搜索



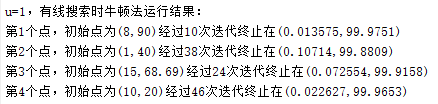


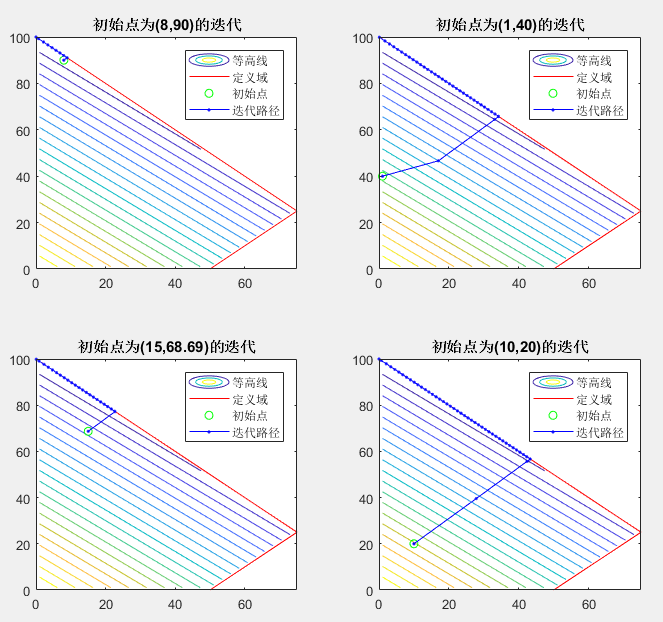
### 3.4.3 μ=0.1，无线搜索





### 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

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：



## 重要表达式

梯度公式：

Hessian阵：

## 参数设置

初始步长α=1;

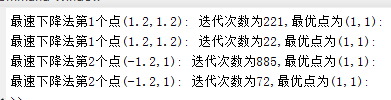
步长变化率γ=0.1;

Armijo条件中的ρ=0.01;

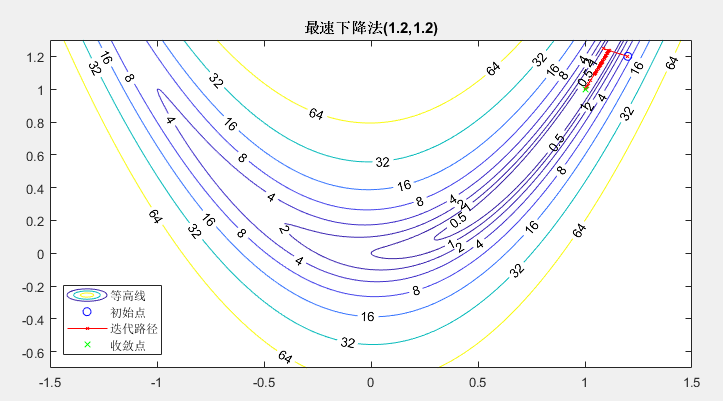
其中步长变化率是经过多次尝试得到的收敛速度较快的一个

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ρ | 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 |

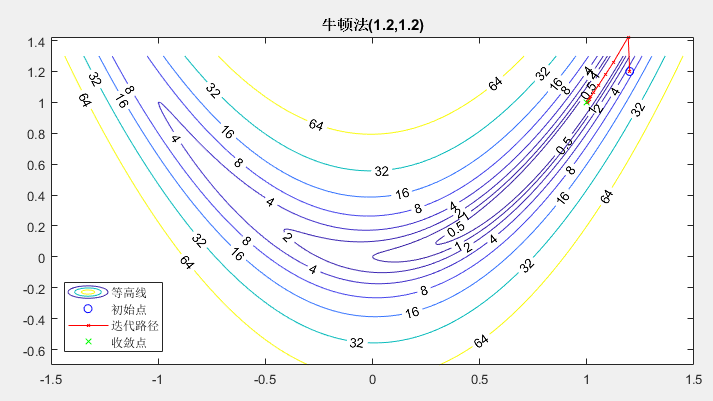
## 计算结果



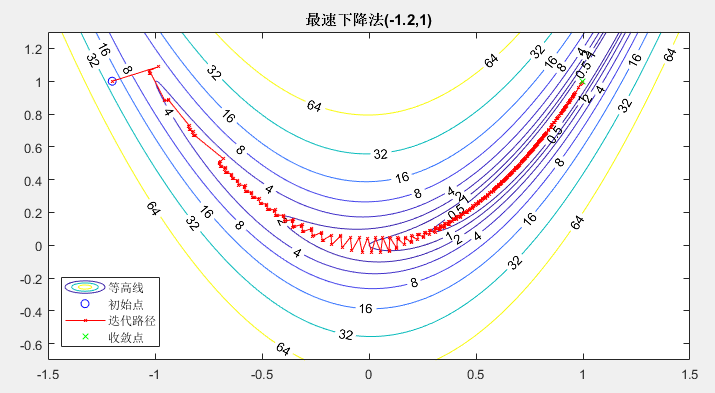
最速下降法在初始点(1.2,1.2)处的迭代路径如下图所示：



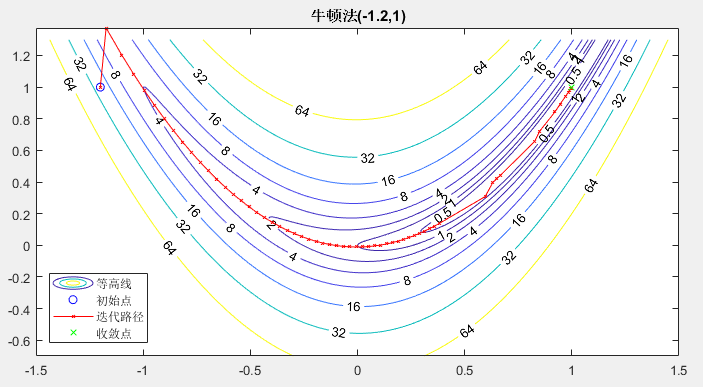
牛顿法在初始点(1.2,1.2)处的迭代路径如下图所示：



最速下降法在初始点(-1.2,1)处的迭代路径如下图所示：



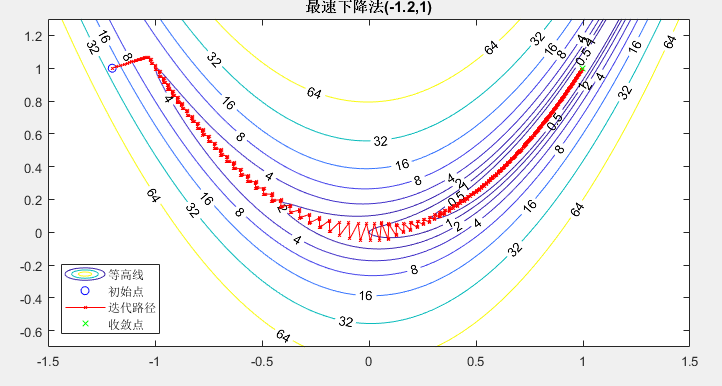
牛顿法在初始点(-1.2,1)处的迭代路径如下图所示：



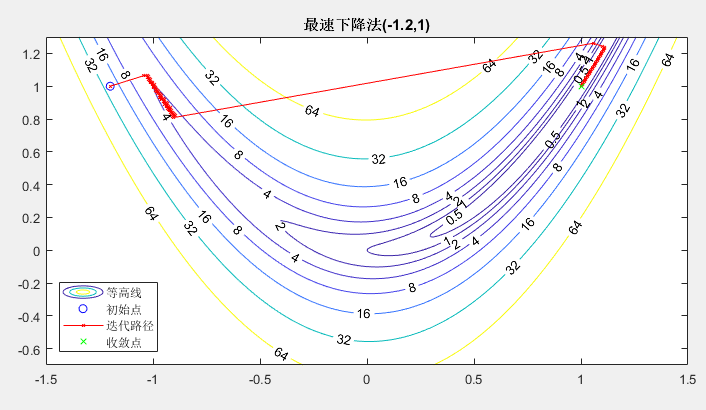
## 说明和分析

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，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γ参数的选取，最速下降法呈现出两种不同的迭代路径；当γ小于0.2时，大致是这样的



而当γ>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。



# 习题5.19

## 算法流程图

共轭梯度法的算法流程图如下图所示：



## 重要表达式

当维度为n时的Hessian阵：



当维度为n时，在x处的梯度为：



## 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)

## 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。

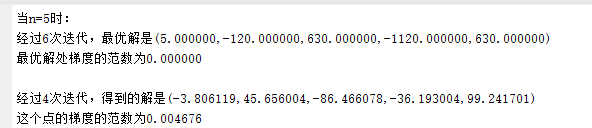
因为当n较大时，G的特征值基本上接近于0，直接求逆的话求得的解很不精确（matlab给出的错误是Matrix is close to singular or badly scaled. Results may be inaccurate.）因此求逆用了如下三种方法：1，matlab自带的一个pinv的方法，2用奇异值分解计算矩阵G的逆（若G=SVDT，则G+=DVST）,3直接用matlab提供的求hilbert矩阵逆的方法invhilb(G)。因为函数为二次函数，最优解满足Gx=b，所以x\*=G-1b。用这个直接求得的x\*对结果进行验证。

以下是实验证明。

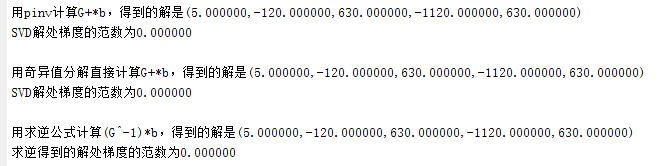
### 5.4.1 n=5

λ(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，



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)

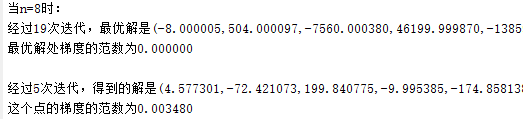


当n=5时，共轭梯度法和直接求所计算出的结果基本一致，且所得解处梯度都近似为0；

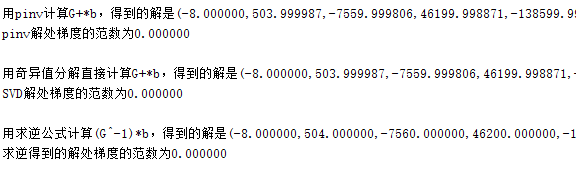
### 5.4.2 n=8

λ(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，



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)



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。

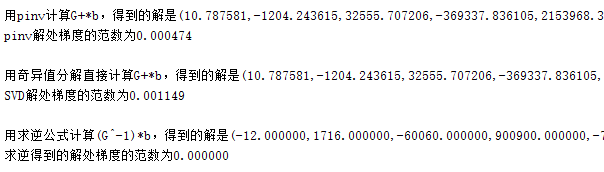
### 5.4.3 n=12

λ(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，



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)

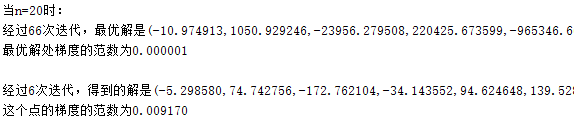


当n=12时，共轭梯度法，pinv和求逆库函数得出的解差别较大，不过svd和pinv这两种方法得出的解差别不大。求逆公式直接求出的解处梯度依然为0，其余的均接近于0。因为目标函数为凸函数，因此应该为计算时的精度问题。我个人更倾向于认为用求逆公式得出的解是正确的，因为它的梯度完全为0

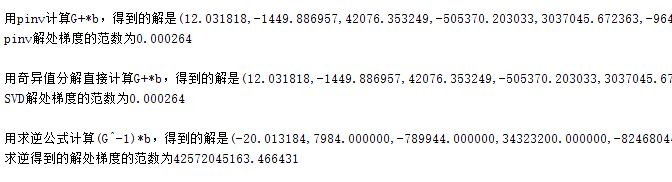
### 5.4.4 n=20

λ(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，



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)



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，hilb(20)\* 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。

# 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