梯度型优化算法的程序实现

（四种方法）

**1.最速下降法**

最速下降法是用负梯度方向



作为搜索方向（因此也称梯度法）。设在附近连续可微, 为搜索方向向量, 。由泰勒展开式得

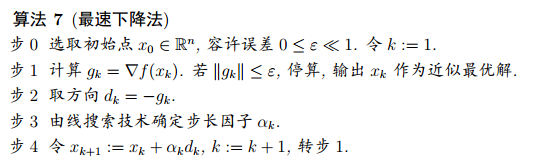
，

那么目标函数在处沿下降的变化率为



其中是和的夹角。要使变化率最小，只有，即取负梯度方向作为最速下降方向。

算法1：



见程序1 grad.m

例：利用程序1求解无约束优化问题



该问题有精确解

解 首先建立两个分别计算目标函数和梯度的 m 文件；

利用程序1，终止准则取为。

调用程序>> x0=[-1.2 1]';

>> [x,val,k]=grad('fun','gfun',x0)

最速下降法的收敛速度是比较缓慢的

**2.牛顿法**

牛顿法也是求解无约束优化问题最早使用的经典算法之一。其基本思想是用迭代点处的处的一阶导数 (梯度) 和二阶导数 (Hesse 阵) 对目标函数进行二次函数近似, 然后把二次模型的极小点作为新的迭代点, 并不断重复这一过程, 直至求得满足精度的近似极小点。

设的Hesse阵连续，截取其在处的泰勒展开式的前三项得

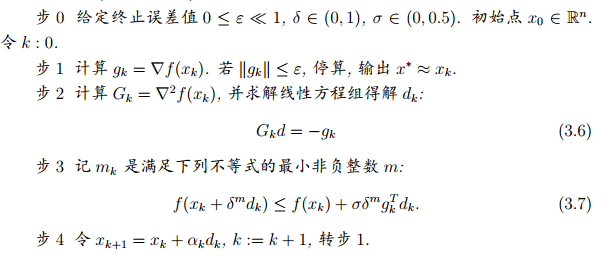


其中，，。求二次函数的稳定点，得牛顿迭代公式



在迭代上式中每步迭代需要求 Hesse 阵的逆,在实际计算中可通过先解得,然后令来避免求逆。

牛顿法最突出的优点是收敛速度快具有局部二阶收敛性。初始点需要足够“靠近”极小点,否则有可能导致算法不收敛。由于实际问题的精确极小点一般是不知道的,因此初始点的选取给算法的实际操作带来了很大的困难。为了克服这一困难,可引入线搜索技术以得到大范围收敛的算法,即所谓的阻尼牛顿法。（基于 Armijo 搜索的阻尼牛顿法）

算法2：

程序见2 dampnm.m

例：利用程序2求解无约束优化问题



该问题有精确解

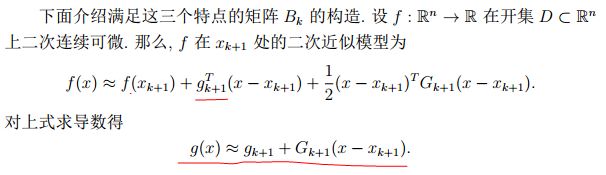
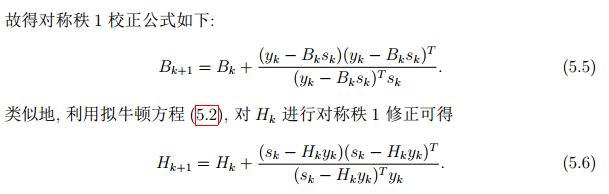
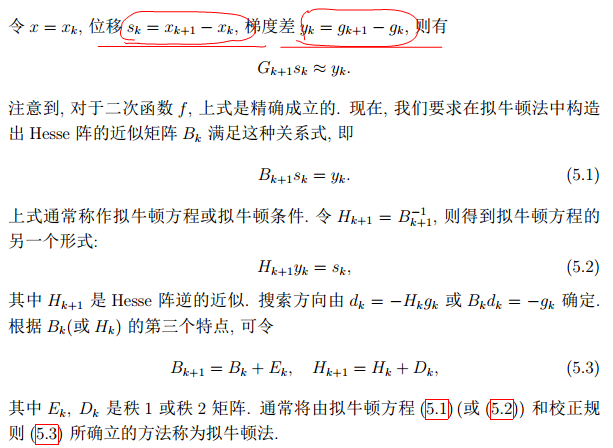
解 除了例中建立的两个计算目标函数和梯度的m文件之外, 还需建立求Hesse阵的m文件，终止准则取为

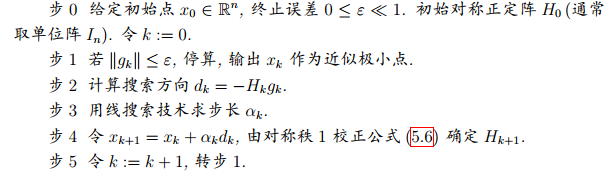
程序2的调用：

>> x0=[-1.2 1]';

>> [x,val,k]=dampnm('fun','gfun','Hesse',x0)

**3.拟牛顿法**

拟牛顿法的基本思想是在基本牛顿法的步2中用Hesee阵的某个近似矩阵取代。

算法3：（对称秩1算法）

程序见 sr1.m

例：利用程序3求解无约束优化问题



该问题有精确解

解 终止准则取为

程序的调用：

x0=[-1.2 1]’;  
[x,val,k]=sr1(’fun’,’gfun’,x0)

**4.共轭梯度法**

共轭梯度法是在每一迭代步利用当前点处的最速下降方向来生成关于凸二次函数的Hesee阵G的共轭方向, 并建立求在上的极小点的方法.

设正定二次目标函数，则的梯度和Hesse

阵分别为

我们取初始方向为初始点处的负梯度方向，即



从出发沿方向进行精确线性搜索得求步长，令



其中满足条件



在处，用在的负梯度方向与的组合来生成，即



然后选取系数使与关于G共轭，即令

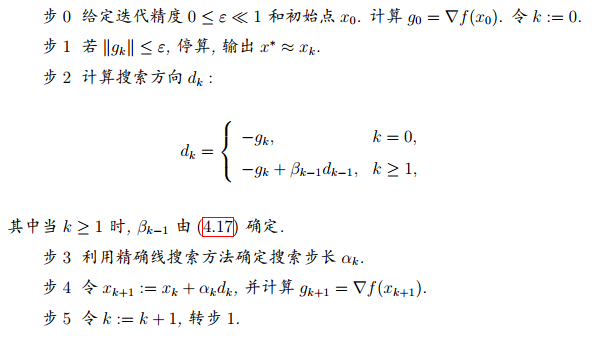


来确定。得



则可知第k步的搜索方向: ， 其中

同时有

算法4

程序见 frcg.m

例：利用程序4求解无约束优化问题



该问题有精确解

解 终止准则取为。

程序的调用：

x0=[-1.2 1]’;  
[x,val,k]=frcg(’fun’,’gfun’,x0)