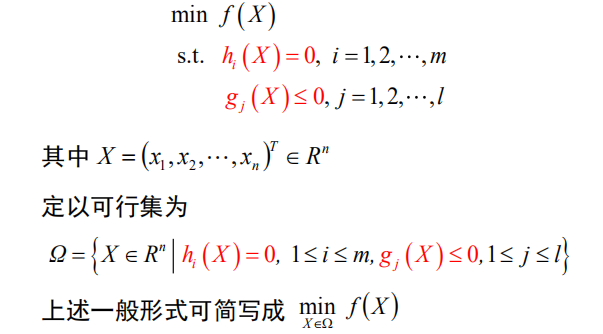
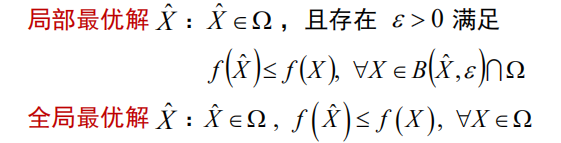
非线性规划是一种求解目标函数或约束条件中有一个或几个非线性函数的最优化问题的方法。

非线性规划的一般形式：



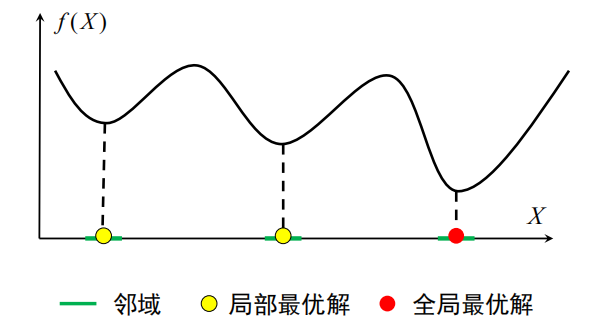
定义局部最优解和全局最优解：



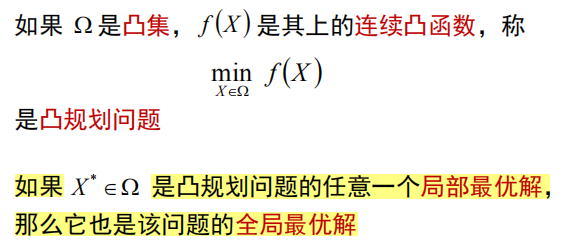
其中X的邻域可简单定义如下：



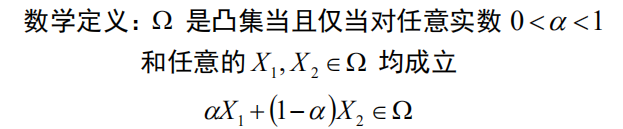
如果在上面的定义中满足 ，则称为严格局部最优解和严格全局最优解



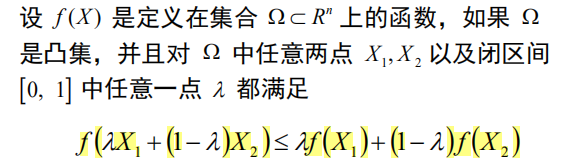
对于任意的函数而言，在众多的局部最优解中精确寻找它的全局最优解不是一个平凡的问题，然而有一种满足特定条件的函数族能够保证该问题的局部最优解即为所定义范围内的全局最优解，这个条件即为凸性，具体而言：

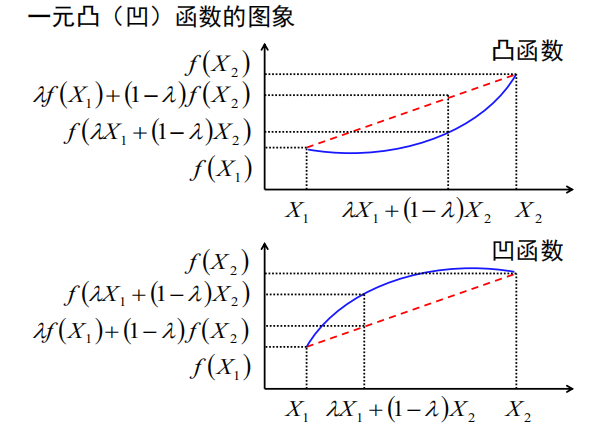


在线性规划中，我们已经知道凸集的定义：

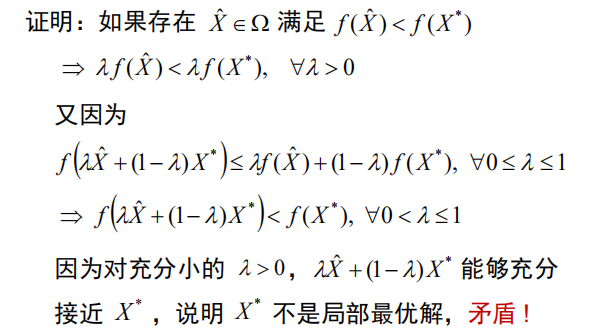


而凸函数则是指具有如下性质的函数：

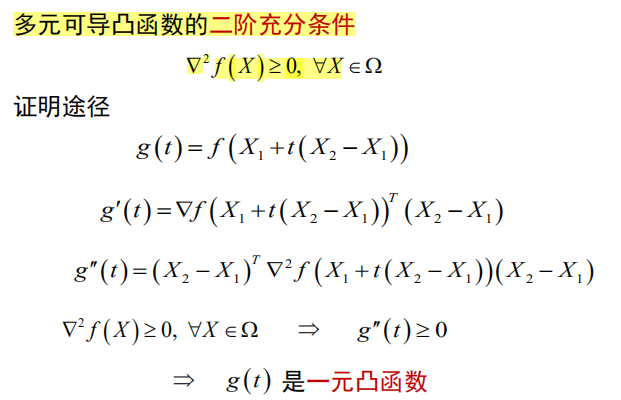




凸函数全局最优值性质的证明如下：



为了方便应用上述性质，我们引入下面的凸函数判别定理：

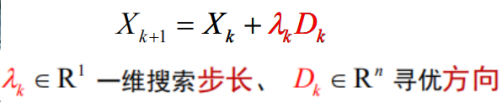


下面介绍针对非线性规划问题的基本途径“迭代算法”

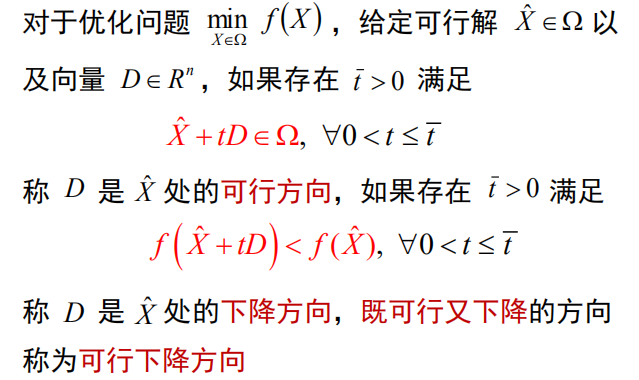
1. 函数求极值问题

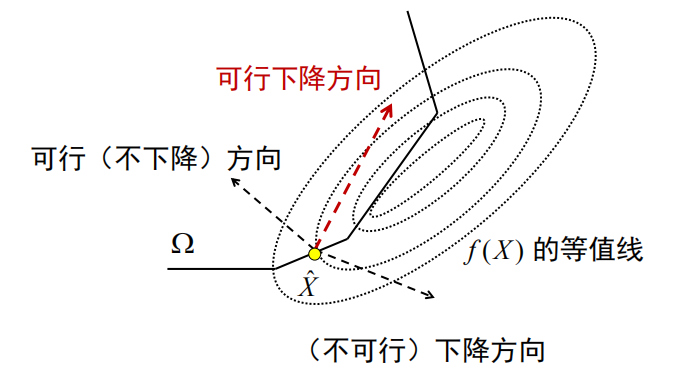


2. 迭代算法

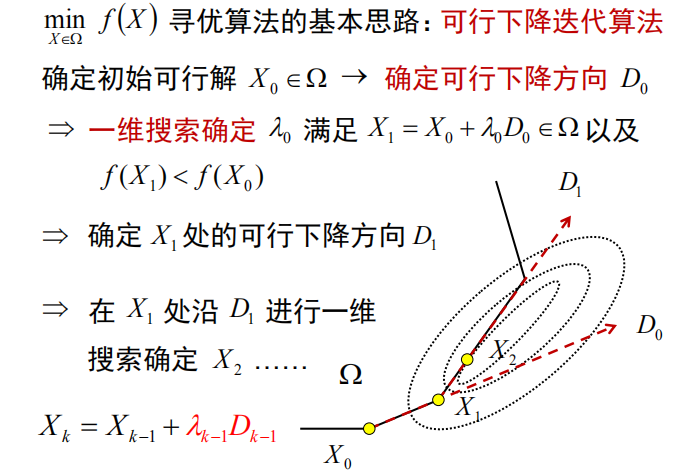


3. 下降方向

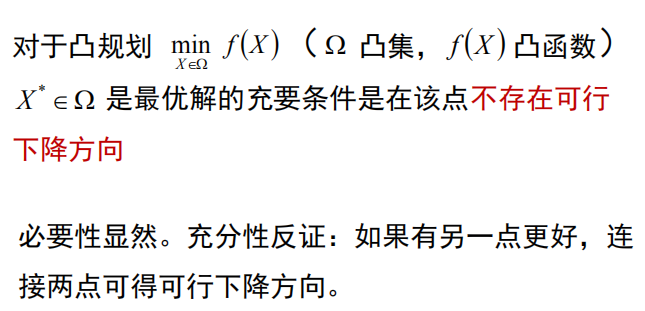




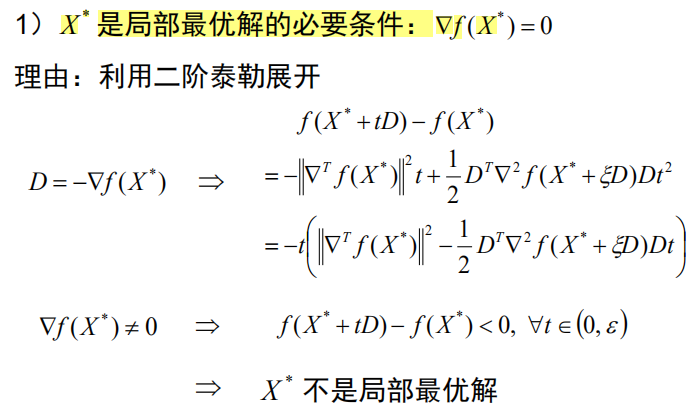
4. 全部思路

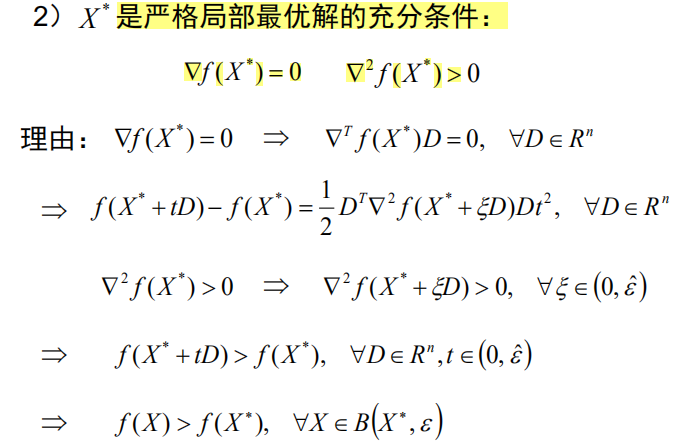


对于前面的凸优化问题，有下面的迭代终止方案：



或者由微积分中的结论有下面最优性条件：



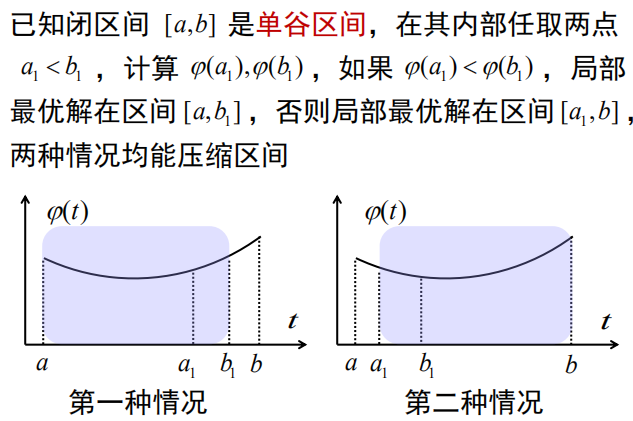


要实现上面的全部程序，需要明确下面两步的方法：

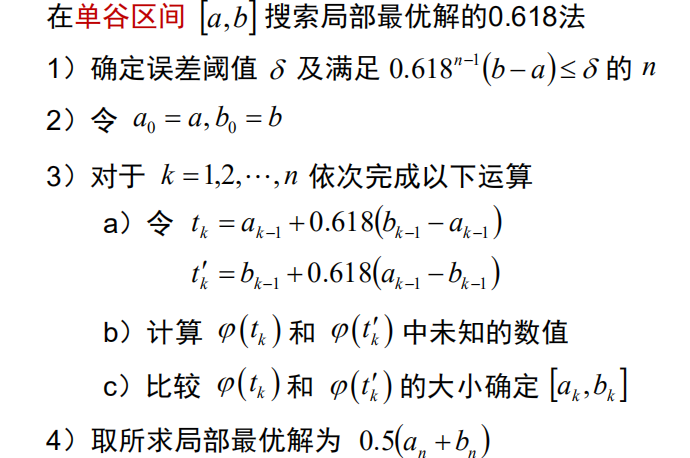
1. 一维最优解的搜索方法

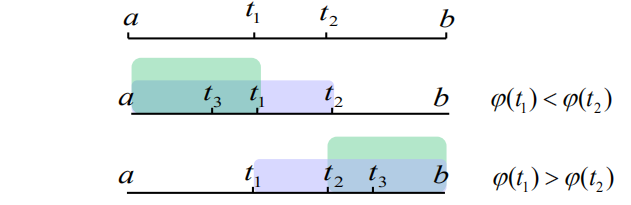
2. 每点处优化方向的确定

精确搜索的办法可以通过压缩区间来逼近，逐步缩小包含局部最优解的区间直至区间长度小于给定阈值

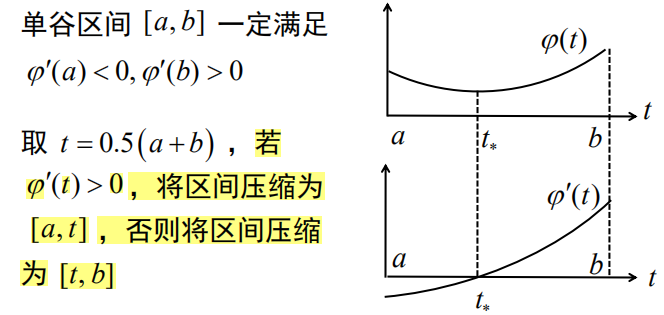


为了保证每计算一个函数值能将区间压缩一个固定比值，可以采用黄金分割比0.618作为每次收缩的大小：



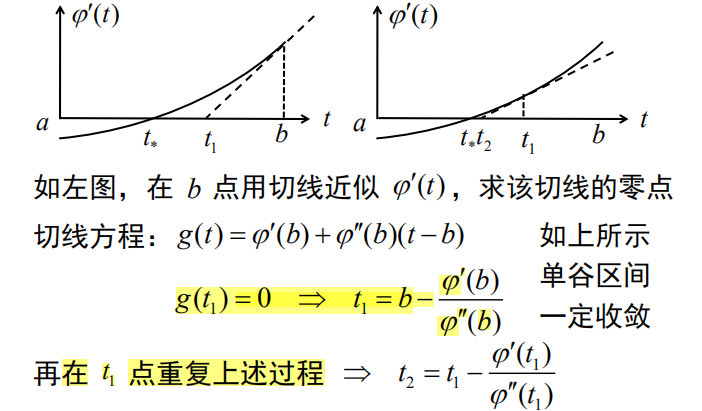


当然我们还可以采用区间对分的方法来进行精确搜索，每次只需要依据一个点的信息进行搜索，所以还需要原函数的导数信息。如下图所示，求解原函数的极值点相当于求解导函数的零点：

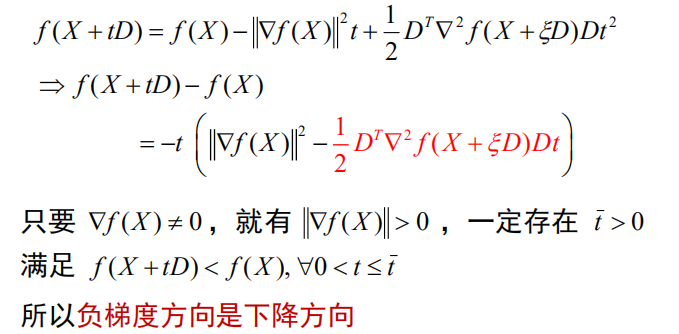


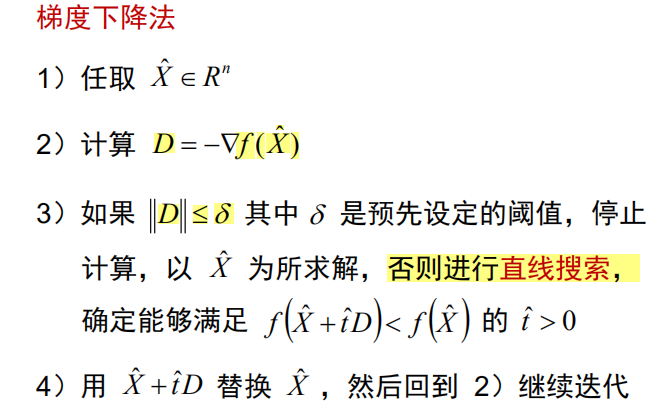
这种方法区间压缩比等于0.5 ，比仅计算函数值的0.618法好，实际效果取决于**导数计算量**。

利用二阶导数的精确搜索算法也称为Newton法，它在一维时的情况如下所示：

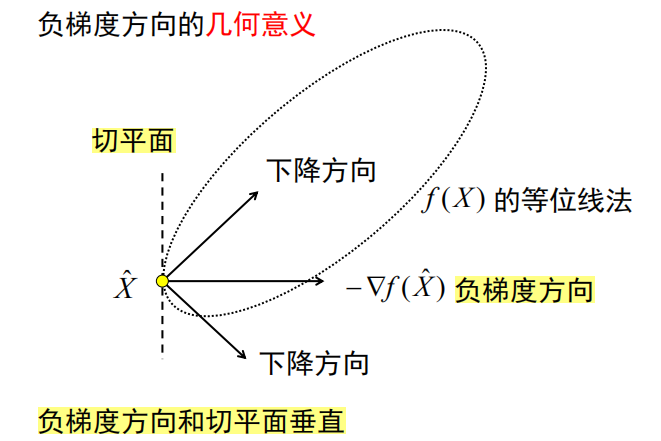


确定下降方向的经典方法如梯度下降法：

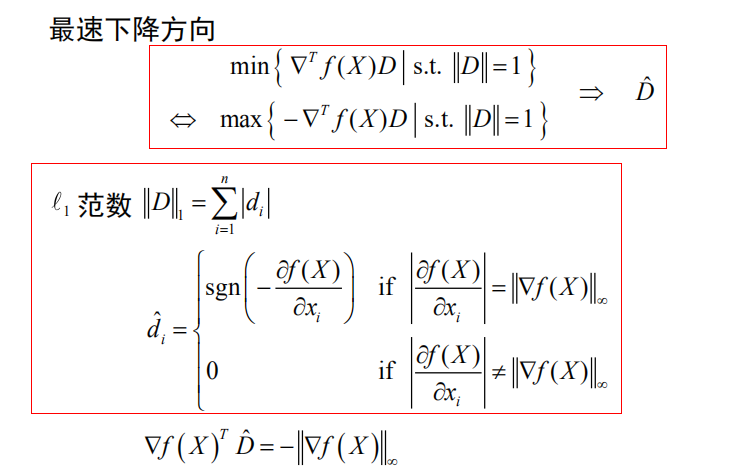


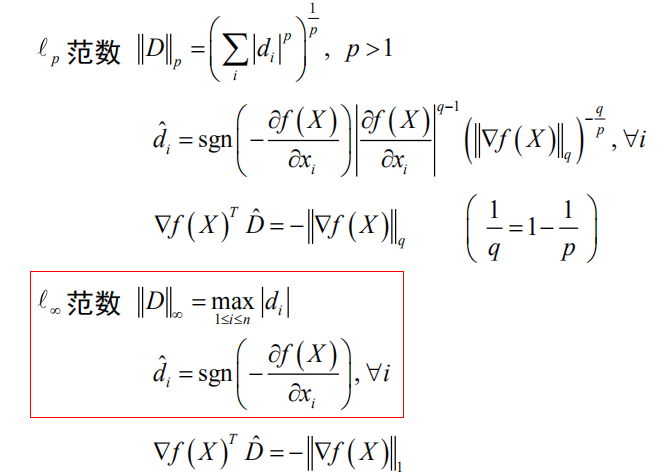


它有清楚的几何意义：

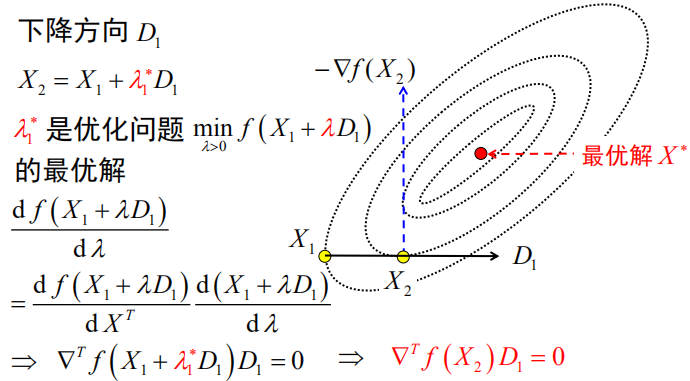


函数的梯度相当于它的l2范数，在实际中还有采用l1、l∞范数的算法，他们都使用了不同范数下的“最速下降方向”

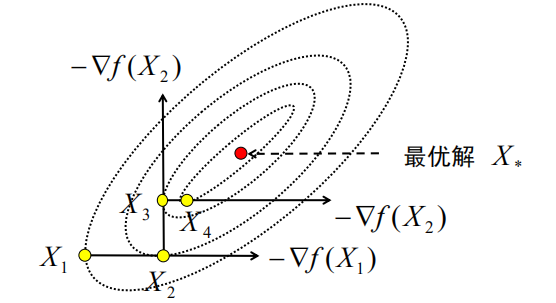




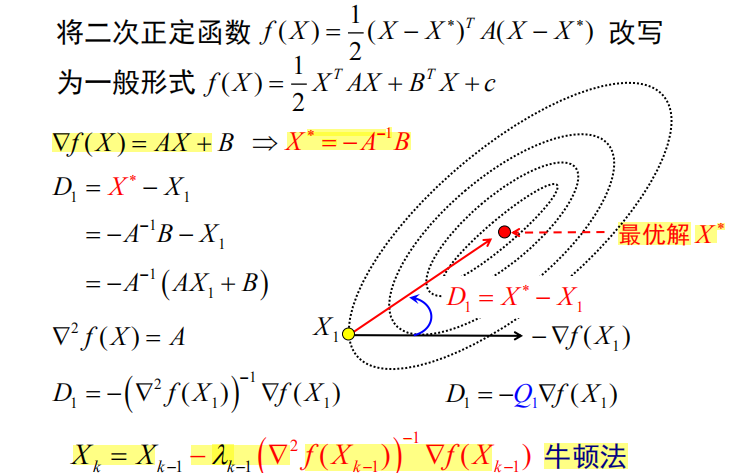
然而，负梯度等范数最速下降法下降法有其固有缺陷，这为共轭梯度法的提出作出了铺垫。精确搜索得到新点的梯度方向与搜索方向正交，即沿负梯度方向精确搜索前进时，相邻两点的梯度互相垂直：



所以在解空间中梯度下降法沿锯齿状路线前进，接近最优解时一维搜索效率很低，前进速度很慢：

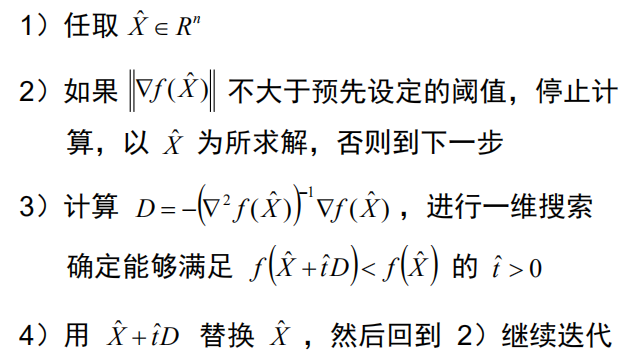


改进梯度下降法的思路可以是校正寻优方向以加快寻优速度。假设被优化的目标函数是一个二次函数



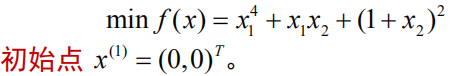
其中被称为牛顿方向

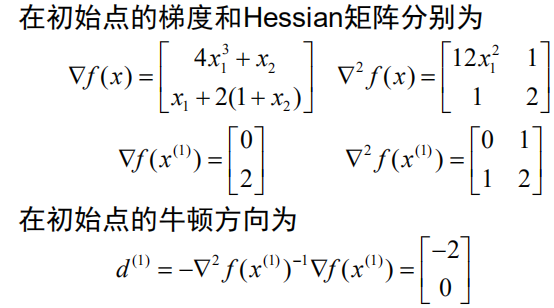
广义牛顿法的流程如下：

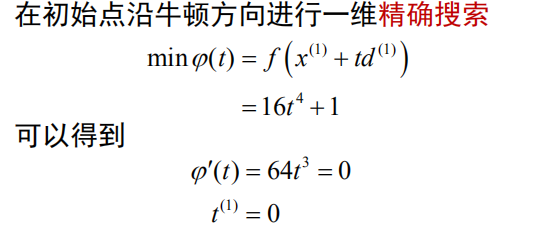


类似于一维搜索中的牛顿法，这里用来求解优化方向的牛顿法也是利用目标函数和二次函数的近似特征，可以期望的是，当目标函数和用于近似的二次函数足够接近时，牛顿法能拥有比最速下降法更快的收敛速度。

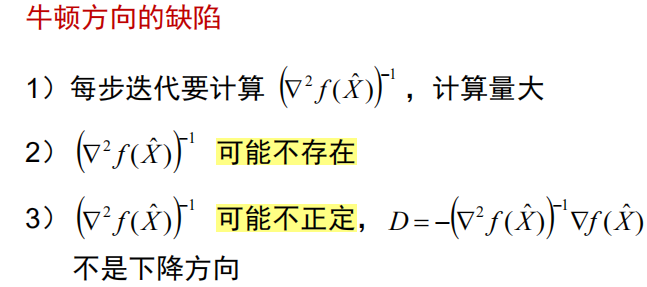
而牛顿法也有其固有缺点，例如，分析如下问题时：



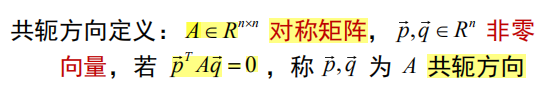




显然，用牛顿法不能产生新的点，而初始点并不是无约束优化问题的极小点。牛顿方向失效的原因在于**初始点的Hessian矩阵非正定**！



要克服牛顿法的缺陷，显然需要找办法绕过Hessian矩阵的计算，这便是共轭梯度法所采用的路径。所谓共轭方向是指：

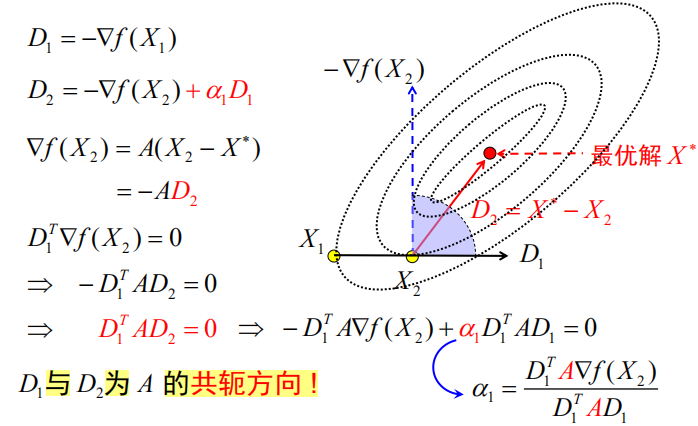


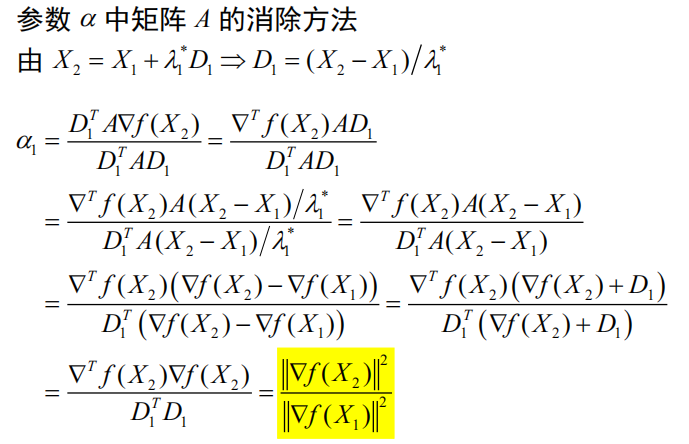
在这里依然考察在牛顿法中近似处理过的多元二次函数：



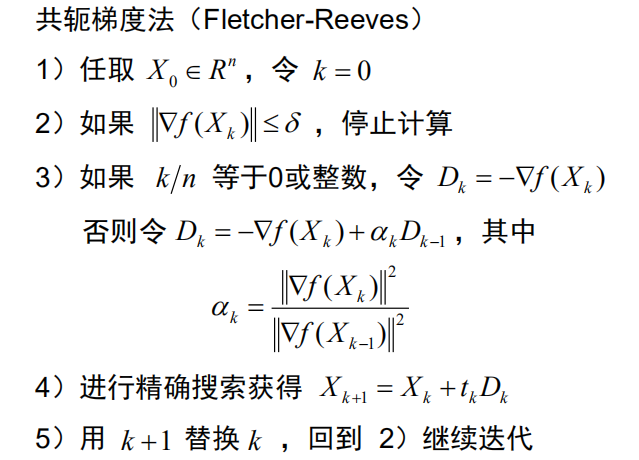
对每次梯度下降的方向作出如下的改进使之更快地收敛到极值点：

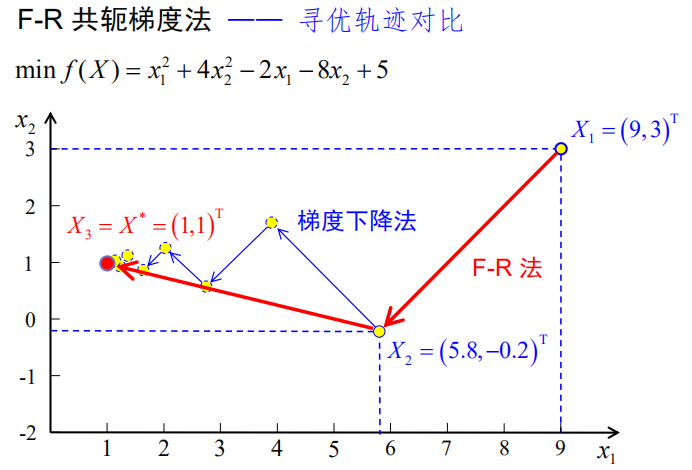




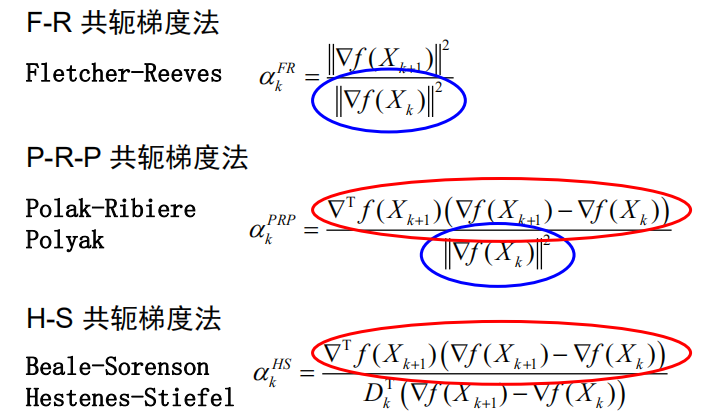


每前后两次寻优方向Dk、Dk+1之间互为A的共轭方向，所以采用上述修正寻优方法的算法也被称为共轭梯度法。上面我们已经推导出了经典的Fletcher-Reeves共轭梯度法。

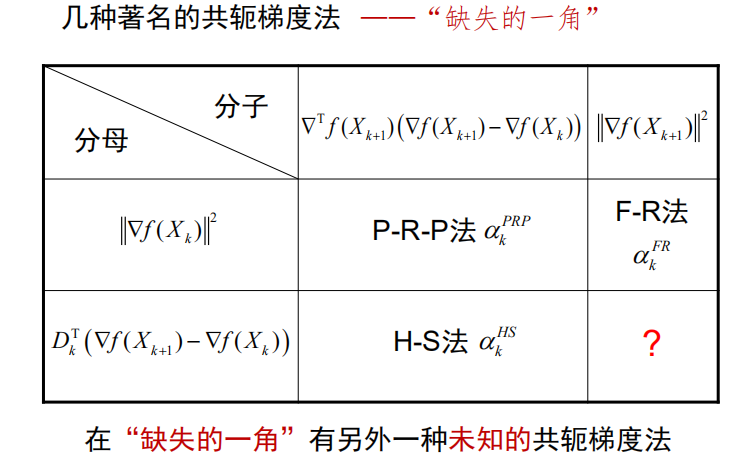


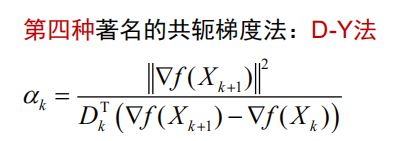


利用相似的方法可以求得其它两种共轭梯度法：

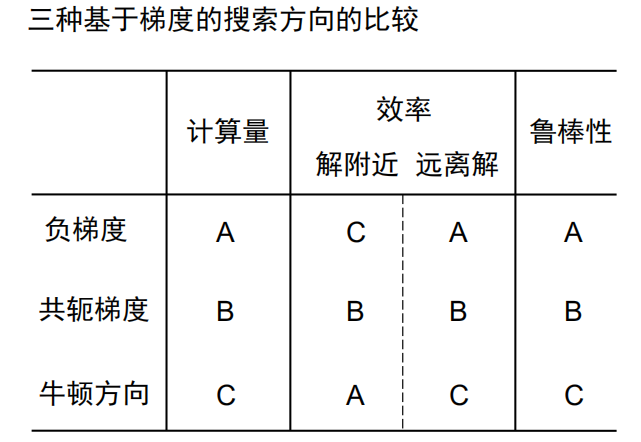


有趣的是，第四种共轭梯度算法是通过对偶性来发现的：



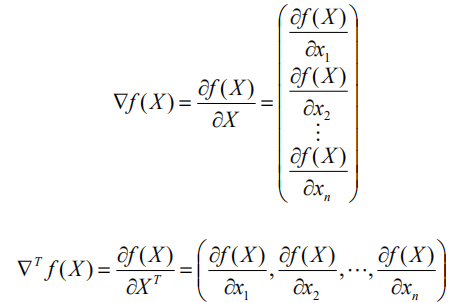


实际中三种基于梯度的搜索方向各有其特点，在实际问题的解决中要注意合理运用：

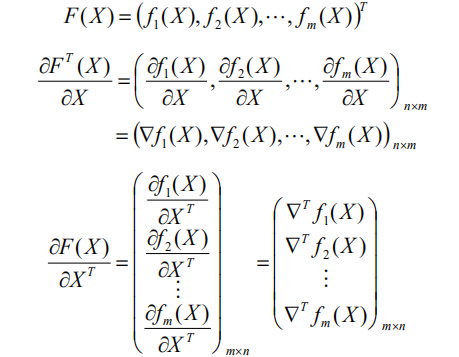


附录：多元函数及矩阵的求导方法如下

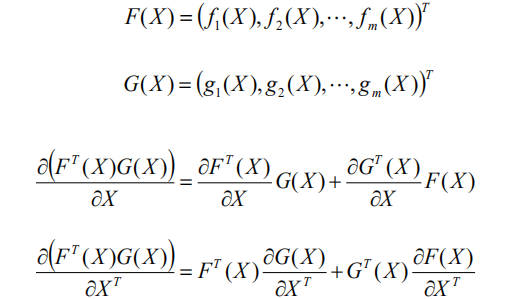
1. 标量函数求偏导数（梯度）



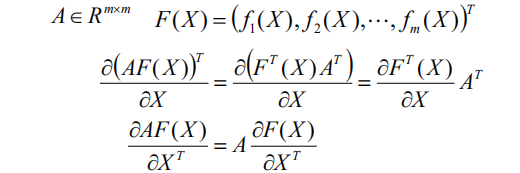
2. 向量函数求偏导数



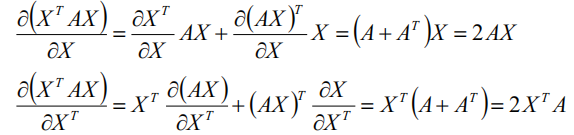
3. 对向量函数的点积求偏导数



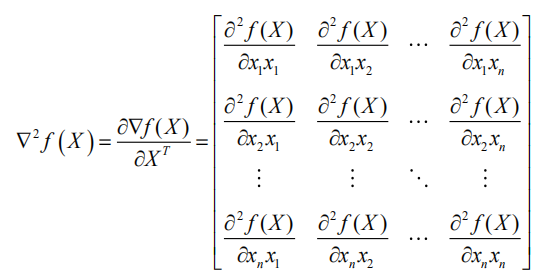
4. 对常数矩阵和向量函数的乘积求偏导数



5. 对二次函数求偏导数



6. 海赛（Hesse）矩阵



7. 多元函数在给定点沿给定方向的二阶泰勒展开

