1. 最速下降法

基本思想：最速下降法通过迭代的方式求函数 $f(x)$的最优解，给定一个初始点，通过迭代找到下一个点，我们希望找到的下一个点能比上一个点有更优的函数值。给定点 x_k，点 x_k处的负梯度方向为最速下降方向，至少在点 x_k的临近范围内是这样的。所以我们可以在点 x_k处选择搜索方向：

$$p_k = -\nabla f(x_k)$$

在选定了搜索方向之后我们就可以沿着搜索方向进行搜索，然后选择点 x_{k+1}，其中

$$x_{k+1} = x_k - t_k \nabla f(x_k)$$

其中 t_k为沿负梯度方向的搜索距离，我们称为步长因子. 在知道迭代公式之后，我们希望得到步长因子 t_k能够满足

$$f\big(x_k - t_k \nabla f(x_k)\big) = min f\big(x_k - t \nabla f(x_k)\big)$$

即我们希望 x_{k+1}为搜索方向上的最小点。

算法描述：

（1）选择初始点 x_0；计算 $f_0=f(x_0)$以及梯度 $g_0=g(x_0)$;k=0

（2）做线搜索$x_{k+1} = x_k - t_k \nabla f(x_k)$;计算 $f_{k+1}=f(x_{k+1})$, $g_{k+1}=g(x_{k+1})$;

（3）判断是否满足终止条件，满足输出 x_{k+1},否则 k=k+1，转 2.

常见的终止条件如函数值的变化范围小于阈值 ε 时终止。我们可以知道最速下降法关键步骤是求解每一步的步长因子 t_k，而对于

正定二次函数：

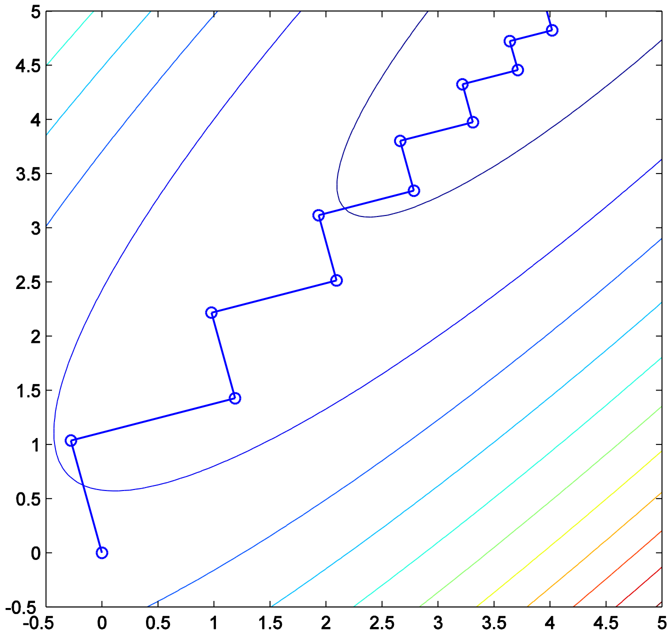
$$f(x)=\frac{1}{2}x^TQx+b^Tx+c$$

由于最速下降法的连续两次的搜索方向是垂直的，所以可以计算得：

$$t_k=\frac{g_k^Tg_k}{g_k^TQg_k}$$

上式便是最速下降法应用于正定二次函数时的步长迭代公式.

实验过程中锯齿现象如图：



2. 牛顿法

牛顿法的基本思想是利用迭代点 x_k 处的一阶导数(梯度)和二阶导数(Hessen 矩阵)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值。

基本牛顿法是一种是利用导数的算法，它每一步的迭代方向都是

沿着当前点函数值下降的方向。对于一个需要求解的优化函数 $f(x)$，求函数的极值的问题可以转化为求导函数 $f'(x)=0$。对函数 $f(x)$ 进行泰勒展开到二阶，得到

$$f(x)=f(x_k)+f'(x_k)(x-x_k)+\frac{1}{2}f''(x_k)(x-x_k)^2$$

对上式求导并置 0，则有

$$f'(x_k)+f''(x_k)(x-x_k)=0$$

得：$x=x_k-\frac{f'(x_k)}{f''(x_k)}$，这就是牛顿法的更新公式。

算法流程：

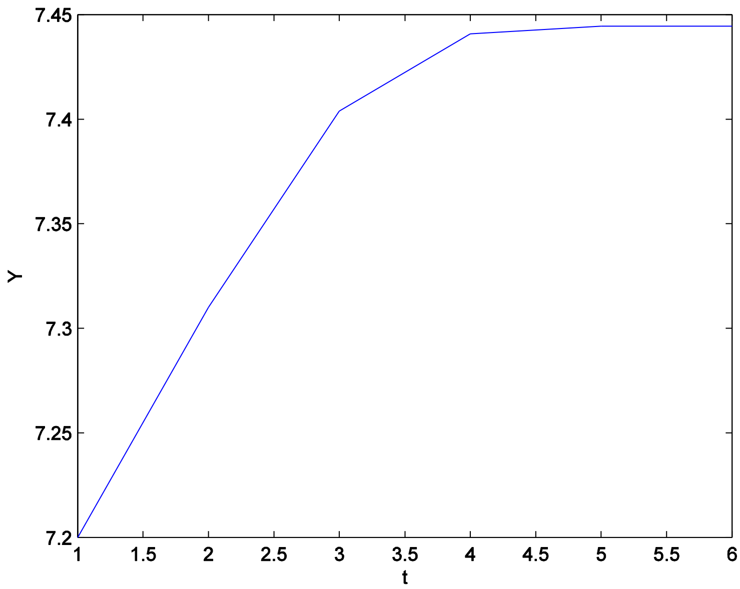
（1）给定终止条件 ε，初始点 x_0，k=1；

（2）计算 $g_k=\nabla f(x_k)$，若 $\|g_k\|\le\varepsilon$，停止，输出 $x^*\approx x_k$；

（3）计算 $G_k=\nabla^2 f(x_k)$，并求解线性方程组得到解 d_k：$G_k d=-g_k$；

（4）令 $x_{k+1}=x_k+d_k$，k=k+1，并转 2.

算法运行示意图：



3. 带线搜索的牛顿法

回溯线搜索：回溯线搜索是一种在求解无约束凸优化问题中，调整搜索步长非常简单有效的方法，也是实际应用中常用的方法。

回溯线搜索的算法流程为：

输入:当前点 x，搜索方向∇x, 参数$\alpha \in (0,0.5), \beta \in (0,1)$；

输出：步长 t。

（1）令 t=1；

（2）如果 $f(x+t\Delta x) > f(x) + \alpha t \nabla f(x)^T \Delta x$, $t = \beta t$；

（3）重复第 2 步，直到 $f(x+t\Delta x) \le f(x) + \alpha t \nabla f(x)^T \Delta x$；

带线搜索的牛顿法：基本牛顿法有两个前提：小的增量 s_k和 G_k正定。当迭代过程中条件不满足时，可以用回溯线搜索方法对牛顿法进行修正，原理如下：

（1）搜索方向的修正：$p_k = \begin{cases} -(G_K)^{-1} g_k, & G_k\text{正定} \\ -g_k & \text{否则} \end{cases}$

（2）步长的修正：$x_{k+1}=x_k+t*p_k$ 其中 t 是回溯线搜索返回的数值，p_k为搜索方向。

算法大致流程如下：

（1）给定终止条件ε，初始点 x_0，k=1；

（2）计算 $g_k = \nabla f(x_k)$, 若$\|g_k\| \le \varepsilon$, 停止，输出$x^* \approx x_k$；

（3）计算 $G_k = \nabla^2 f(x_k)$，判断 Gk 的正定性，选择合适的搜索方向 $p_k = \begin{cases} -(G_K)^{-1} g_k, & G_k\text{正定} \\ -g_k & \text{否则} \end{cases}$.

（4）运行回溯线搜索方法 backtrackingls 得到最佳步长 t。

（5）令 $x_{k+1}=x_k+t*p_k$，k=k+1, 并转 2.

4. 回溯线搜索的最速下降法：

最速下降方法（梯度下降法或者梯度法）：$\begin{cases} minimize\ p^T g^k \\ subject\ to\ \ \|p\|_2 = 1 \end{cases}$，每次迭代计算梯度和步长。如果 f(x) 连续可微有下界，且 g(x) 是连续的，以负梯度作为搜索方向。

步长可选择回溯线搜索确定：$x^{k+1} = x^k - \alpha_k g^k$.

算法的大致流程如下：

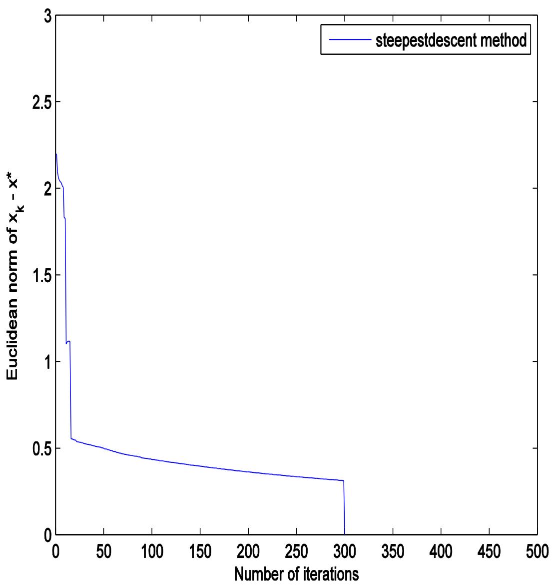
（1）选择初始点 x_0；计算 $f_0=f(x_0)$ 以及梯度 $g_0=g(x_0)$；k=0

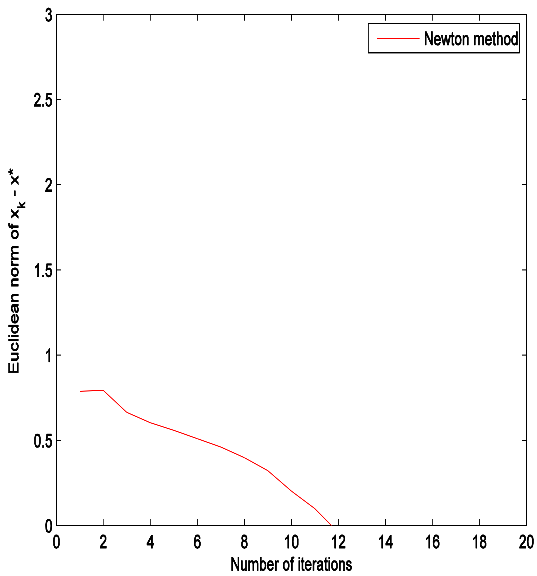
（2）运行回溯线搜索方法，得到 t_k；

（3）做线搜索 $x_{k+1} = x_k - t_k \nabla f(x_k)$；计算 $f_{k+1}=f(x_{k+1})$，$g_{k+1}=g(x_{k+1})$；

（4）判断是否满足终止条件，满足输出 x_{k+1}，否则 k=k+1，转 2.

带回溯线搜索的最速下降法和牛顿法运行示意图如下：





5. 共轭梯度法

共轭梯度法是一种求解对称正定线性方程组 Ax=b 的迭代方法。事实上，求解 Ax=b 的解等价求解：$\min\|Ax-b\|_2^2$ ，将其展开后等价于 $\min\frac{1}{2}x^TA^TAX-b^TAx$，即原问题转换为一个二次规划问题。共轭梯度法是介于梯度下降法与牛顿法之间的一个方法，是一个一阶方法。它克服了梯度下降法收敛慢的缺点，又避免了存储和计算牛顿法所需要的二阶导数信息。在 n 维的优化问题中，共轭梯度法最多 n 次迭代就能找到最优解（是找到，不是接近），但是只针对二次规划问题。共轭梯度法的思想就是找到 n 个两两共轭的共轭方向，每次沿着一个方向优化得到该方向上的极小值，后面再沿其它方向求极小值的时候，不会影响前面已经得到的沿哪些方向上的极小值，所以理论上对 n 个方向都求出极小值就得到了 n 维问题的极小值。

目标函数标准形式为：$\min\frac{1}{2}x^TQX-b^Tx$，我们需要找到 n 个相互 Q-conjugate 的基向量 d1, d2, ⋯, dn, 它们相互共轭且线性无关。因此空间中任意向量 x 都可以用这组基向量表示：$\mathrm{x}=\sum_{i=1}^{n}a_id_i$，目标函数可以化简为：

$$\min_{a_1..a_n}\frac{1}{2}\sum_{i=1}^{n}(a_i^2d_i^TQd_i-a_ib^Td_i)$$

求最小值，直接求导可以得到：

$$a_i=\frac{b^Td_i}{d_i^TQd_i}$$

所以最优解$x^*=\sum_{i=1}^{n}a_id_i$。

现在，我们只需要构建 Q 向量组 d1,d2,⋯,dnd1,d2,⋯,dn。有一种称为 Gram-Schmidt 过程的算法可以用来找到这组向量。在实际算法实现中，我们往往是边求 a_i,边求 d_i。下面给出完整的共轭梯度算法。

(1) 初始化 x_0，计算 $g_0=G*x_0-b$;

(2) 设置 $p_0=-g_0$，$k=0$

(3) 当残差$\|g_k\| > \varepsilon$,进行如下循环：

$$D = G * pk;$$

$$a_k = \frac{g_k^T g^K}{p_k^T d};$$

$$x_{k+1} = x_k + a_k p_k;$$

$$g_{k+1} = g_k + a_k * d;$$

$$\beta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k};$$

$$p_{k+1} = -g_{k+1} + \beta_{k+1} p_k;$$

$$\mathrm{k} = \mathrm{k} + 1;$$

(4)返回 x_k 作为 $x*$;

题意中数据迭代次数变化如下：

5	7
8	20
12	39
20	63

随着 n 的增大，迭代次数增加。

6. 带线搜索的高斯牛顿法

高斯—牛顿迭代法(Gauss-Newton iteration method)是非线性回

归模型中求回归参数进行最小二乘的一种迭代方法，该法使用泰勒级数展开式去近似地代替非线性回归模型，然后通过多次迭代，多次修正回归系数，使回归系数不断逼近非线性回归模型的最佳回归系数，最后使原模型的残差 r 平方和达到最小。其直观思想是先选取一个参数向量的参数值 β，若函数 $f_t(X_t, \beta)$ 在 β_0 附近有连续二阶偏导数，则在 β_0 的邻域内可近似地将 $f_t(X_t, \beta)$ 看作是线性，因而可近似地用线性最小二乘法求解。

令 $f(x)=\frac{1}{2}\sum_{i=1}^{m} r_i(x)^2$，则 $g(x)=A(x)^T r(x)$，$G(x)=A(x)^T A(x)+\sum_{i=1}^{m} r_i(x)\nabla^2 r_i(x)$。通常 $r_i(x)$ 都很小，所以 $G(x)\approx A(x)^T A(x)$。基本高斯牛顿法的迭代为：

（1）通过 ${A_k}^T A_k s=-{A_k}^T r_k$，求解 s_k；

（2）$x_{k+1}=x_k+s_k$

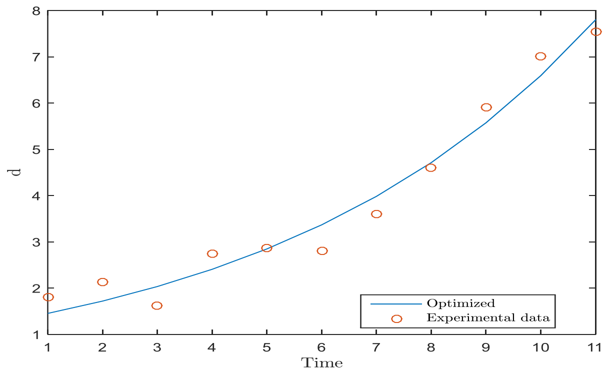
带回溯线搜索的 GN 法：

主要区别于 GN 法是搜索步长的加入，流程如下：

（1）求解 ${A_k}^T A_k s=-{A_k}^T r_k$；

（2）用回溯线搜索得到最佳步长 t_k；

（3）$x_{k+1}=x_k+t_k s_k$



算法运行示意图如下：

7. 基于 Steihaug 共轭梯度法的信赖域法：

因为共轭梯度法在迭代过程中产生的近似解的范数在增大，因此如果迭代满足$\left\|x^{i+1}\right\|_2 > \Delta$,则在求解信赖域问题的解必然在边界上,即$\|s^*\|_2 = \Delta$。如果存在 i 满足$p_i^T B p_i \le 0$或者$\left\|x^i + \alpha_i p_i\right\|_2 > \Delta$,则算法应该在第 i 次迭代后停止。在这两种情况下，最简单的终止条件策略是停在点$s' = x^i + \alpha_B p_i$，其中α_B是二次方程：

$$\left\|x^i + \alpha p_i\right\|_2^2 = \Delta^2$$

的根，且点s'满足信赖域条件。

基于 Steihaug 共轭梯度法的信赖域法的算法流程如下：

（1） 设置阈值ϵ，$x_0=0, r_0=g, p_0=-g$;

（2） 如果$\|r_0\| < \epsilon$，$s' = x_0$，退出；

（3） 执行以下循环(循环变量 j)，直到满足条件退出：

[1]如果$p_j^T B p_j \le 0$，解 $\left\|x_j + \alpha p_j\right\|_2^2 = \Delta^2$，并计算$s'$后，退出。

[2]$\alpha_j = \frac{r_j^T r_j}{p_j^T B p_j}$;$x_{j+1} = x_j + \alpha_j p_j$;

如果$\left\|x^{i+1}\right\|_2 > \Delta$,解$\left\|x_j + \alpha p_j\right\|_2^2 = \Delta^2$,并计算$s'$后。退出；

[3]$r_{j+1} = r_j + \alpha_j B p_j$;

如果$\|r_{i+1}\| < \varepsilon$，则$s' = x_{j+1}$，退出。

[4]$\beta_{j+1} = \frac{r_{j+1}^T r_{j+1}}{r_j^T r_j}$;

[5]$p_{j+1} = -r_{j+1} + \beta_{j+1} p_j$;