该问题中已知传感矩阵 A 和观测到的信号 b，需求解原始信号 x。因此需使算法得到的 Ax 与 b 尽可能接近，即优化|Ax-b|^2，由于 x 为稀疏向量，因此采取如下优化目标：

优化目标： $F = 0.5 * \|Ax - b\|_2^2 + \lambda \Sigma \sqrt{x_i^2 + \varepsilon} := F1 + F2$

优化目标梯度： $\mathrm{dF = A'*(A*x - b) + \lambda * x./((x.^\wedge 2 + epsilon).^\wedge 0.5) := dF1 + dF2}$

Hassan 矩阵： $\mathrm{dF\_2 = A'*A + \lambda * diag(epsilon./((x.^\wedge 2 + epsilon).^\wedge 1.5)) := H1 + H2}$

迭代停止准则：

由于目标函数为下凸函数，极小值点梯度为 0，因此设置迭代停止准则为：优化目标函数的梯度各分量平均值小于 10^(-5)。

λ 选取：

当 λ 较大时，dF 中的 dF2 较大，因此达到同样迭代停止准则时，x 更接近 0，得到的真实信号误差（即 $\|Ax - b\|_2^2$）较大。而 λ 过小时，得到 x 可能不够稀疏。

因此对于各 ε，不断增大 λ，计算模型得到的观测结果与真实观测结果 b 的相对误差（$err = \frac{\|Ax - b\|_2^2}{\|b\|_2^2}$），选择能够使该相对误差的最终结果约下降为 10^(-4) 的 λ。

步长：

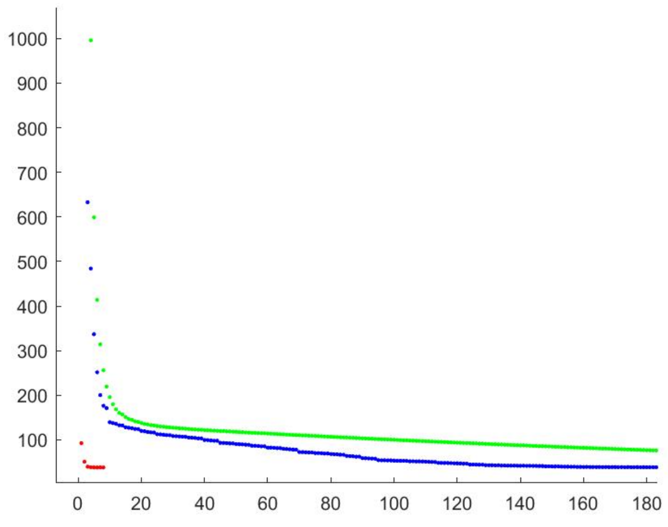
为便于比较各算法，梯度下降法、BB 算法均搭配 Armijo 准则选取。初值设置为 1.

一、 $\varepsilon = 10^\wedge(-2)$

根据上述 λ 选取准则，得到的 λ 为 1.

其中牛顿法由于 Hassen 矩阵在迭代过程中可能非正定，采用修正牛顿法:，$t^k = -\min(\lambda(\nabla^2 f(x_k))) + m$，$G_k = \nabla^2 f(x_k) + t^k * I$，$x^{k+1} = x^k - a * G_k^{-1} * \nabla f(x^k)$，不断增大 m，发现 m 在 2 附近时牛顿法收敛最快。

下图为各算法得到的目标函数值与迭代步数关系：



（其中绿色为梯度下降，蓝色为 BB 算法，红色为牛顿法）

图中可以看出，相比于其它两种算法，牛顿法收敛最快，约迭代 8 次，而 BB 算法、梯度下降法分别约为 583 次、2077 次。

梯度下降法、BB 算法、牛顿法迭代总时间分别为 29s、13s、0.24s，平均每轮迭代时间约为 0.014s、0.0223s、0.03s，可以看出，虽然牛顿法收敛最快，但每轮迭代计算量最大，而梯度下降法虽然收敛最慢，但每轮计算量最小。

由于迭代是否停止是跟据梯度各分量绝对值的平均值判断的，因此各算法得到的相对误差、目标函数值相似，没有显著差别。此时目标函数值约为 39. 其中 $F1 \approx 0.613$，$F2 \approx 38.85$，$\|x\|_1 \approx 20.7$。为衡量 x 稀疏度，多次运行计算平均值

$$\frac{\|x\|_1}{\|u\|_1} \approx 1.2$$

该题中已知原始信号为 u，因此为比较算法对信号还原程度，计算 $\frac{\|x-u\|_2^2}{\|u\|_2^2}$，多次运行计算平均值约为 0.045. 该值较小，可以看出算法对信号还原度较高。

二、 $\varepsilon = 10^{\wedge}(-4)$

“λ选取”中的 err 没有明显变化，λ 仍取 1.

对于牛顿法，Hassen 矩阵中的 H2 为对角元素均正的对角矩阵，因此当该矩阵中元素较大时，Hassen 矩阵更容易正定。

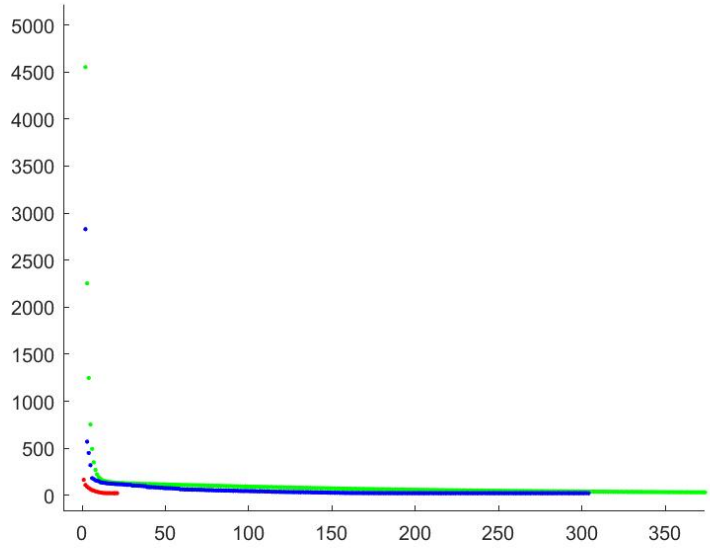
考虑 ε 减小时 H2 如何变化：令 n=m+d,(d>1)，$\rho := \frac{H2(10^{-m})}{H2(10^{-n})} = 10^d (\frac{x_i^2 + 10^{-m-d}}{x_i^2 + 10^{-m}})^{1.5}$，若 $\rho > 1$，则 $\frac{x_i^2 + 10^{-m-d}}{x_i^2 + 10^{-m}} > 10^{\frac{-2d}{3}}$，

$10^{-m} < \frac{(1 - 10^{\frac{-2d}{3}}) x_i^2}{10^{\frac{-2d}{3}} - 10^{-d}}$，令该不等式为(a)。当 m=2, d=2 时，使得 $\rho > 1$ 的 x_i 约满足：$|x_i| > 0.02$。由于 x_i 满足标准正态分布，因此 x_i 满足 $|x_i| > 0.02$ 的概率较大（约为 99%)，即此时的 H2 对角线元素比 $\varepsilon = 10^{\wedge}(-2)$ 时 H2 对角线元素小的可能性较大。

因此应增大上述牛顿迭代公式中的 m，当 m 在 10 附近时牛顿法收敛最快。因此牛顿法迭代公式为：

$t^k = -\min(\lambda(\nabla^2 f(x_k))) + 10$，$G_k = \nabla^2 f(x_k) + t^k * I$，$x^{k+1} = x^k - a * G_k^{-1} * \nabla f(x^k)$

下图为各算法得到的目标函数值与迭代步数关系：



（其中绿色为梯度下降，蓝色为BB算法，红色为牛顿法）

图中可以看出，相比于其它两种算法，牛顿法收敛最快，约迭代28次，而BB算法、梯度下降法分别约为374次、719次。但此时牛顿法迭代法比 $\varepsilon=10^{\wedge}(-2)$ 时收敛更慢，而其他两种算法收敛更快。

此时目标函数值约为24. 其中 $F1\approx 0.14$，$F2\approx 23.84$，$\|x\|_1\approx 22$，多次运行计算平均值 $\frac{\|x\|_1}{\|u\|_1}\approx 1$。可以看出在此种迭代停止准则情况下 $\varepsilon=10^{\wedge}(-4)$ 时得到的x更稀疏。

多次计算求平均值，得 $\frac{\|x-u\|_2^2}{\|u\|_2^2}\approx 0.0015$，因此 $\varepsilon=10^{\wedge}(-4)$ 时算法对信号还原度更高。

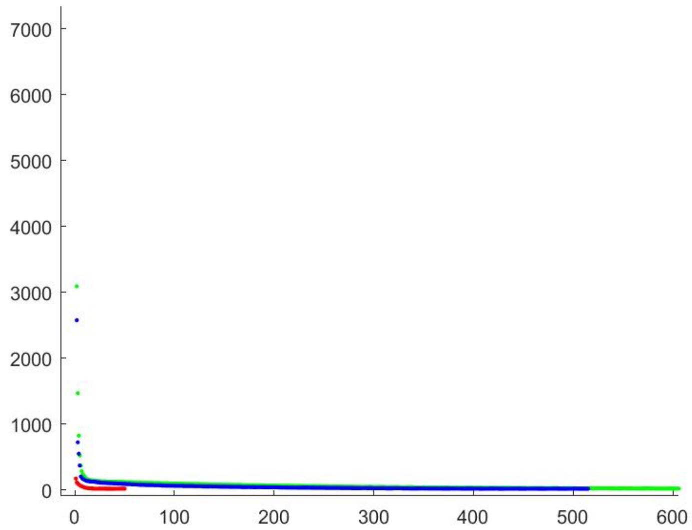
三、 $\varepsilon=10^{\wedge}(-6)$

λ仍取1。

对于牛顿法，此时m在10附近时牛顿法收敛最快。因此牛顿法迭代公式为：

$t^k=-\min(\lambda(\nabla^2 f(x_k)))+10$，$G_k=\nabla^2 f(x_k)+t^k*I$，$x^{k+1}=x^k-a*G_k^{-1}*\nabla f(x^k)$

下图为各算法得到的目标函数值与迭代步数关系：



（其中绿色为梯度下降，蓝色为 BB 算法，红色为牛顿法）

图中可以看出，相比于其它两种算法，牛顿法收敛最快，约迭代 50 次，而 BB 算法、梯度下降法分别约为 515 次、991 次。可以看出，随着 ε 的减小，牛顿法迭代越来越慢。

此时目标函数值约为 21. 其中 $F1 \approx 0.118$， $F2 \approx 20.755$, $\|x\|_1 \approx 20.6$，多次运行计算平均值 $\frac{\|x\|_1}{\|u\|_1} \approx 0.99$。此时得到的 x 与 $\varepsilon = 10^{\wedge}(-4)$ 时得到的 x 稀疏度相似。

多次计算求平均值，得 $\frac{\|x-u\|_2^2}{\|u\|_2^2} \approx 3.8*10^{-4}$，因此 $\varepsilon = 10^{\wedge}(-6)$ 时算法对信号还原度更高。

四、不同噪声水平

若 ε 取 10ˆ(-4)。

1. 噪声水平取 0 时：

牛顿法约迭代 20 次，而 BB 算法、梯度下降法分别约为 462 次、1573 次。此时梯度下降法收敛比噪声为 0.01 时更慢，而其它两种算法无显著差别。

此时目标函数值约为 21. 其中 $F1 \approx 0.21$， $F2 \approx 23.84$， $\|x\|_1 \approx 21$，多次运行计算平均值 $\frac{\|x\|_1}{\|u\|_1} \approx 1$。

此时由于没有噪声，Ax-b 的极值点对应 x 更稀疏，因此可增大 λ 以提高收敛速度。若增大 λ 至 1.5，梯度下降法收敛变快，此时依然能确保 err 下降至约 10ˆ(-4)。牛顿法、BB 算法、梯度下降法分别约迭代 24 次、236 次、796 次，收敛速度与噪声为 0.01 时相似。

多次计算求平均值，得 $\frac{\|x-u\|_2^2}{\|u\|_2^2} \approx 0.002$，与噪声程度为 0.01 时相似。

2.噪声水平取 0.5 时:

此时为确保 err 下降至约 10^(-4)，需将 λ 降至 0.8

牛顿法约迭代 69 次，而 BB 算法、梯度下降法分别约为 821 次、2977 次。与噪声为 0.01 时相比收敛速度均变慢。

此时目标函数值约为 21.其中 $F1 \approx 0.336$， $F2 \approx 20.4$， $\|x\|_1 \approx 24.4$，多次运行计算平均值 $\frac{\|x\|_1}{\|u\|_1} \approx 1.15$。因此噪声程度为 0.01 时得到的 x 更稀疏。

多次计算求平均值，得 $\frac{\|x-u\|_2^2}{\|u\|_2^2} \approx 0.02$，比噪声为 0.01 时得到的 x 还原度更低。