第二次数值作业

陈晓宇 121110008

一. 题目要求

练习 6.6. 编制程序实现 Jacobi 迭代方法和 Gauss-Seidel 方法。对应不同的停机标准（例如残量，相邻差量，后验误差停机标准），比较迭代次数以及算法停止时的真实误差。

练习 6.7. 编写程序实现 SOR 迭代方法。以真实误差作为停机标准，数值观测 SOR 迭代方法中松弛因子！对迭代次数的影响，找到最佳迭代因子的取值。

练习 6.8. 对于 J 方法、GS 方法和（带有最佳松弛因子的）SOR 方法，分别绘制误差下降曲线以及残量的下降曲线（采用对数坐标系），绘制（按真实误差）迭代次数与矩阵阶数倒数的关系。

练习 6.9. 编制变系数 Richardson 迭代方法，绘制误差下降曲线以及残量的下降曲线。观测循环指标 m 对收敛速度的影响。

练习 6.10. 对 Jacobi 迭代进行半迭代加速（考虑 m=5 的循环迭代），绘制误差下降曲线以及残量的下降曲线。

练习 6.11. 共轭梯度算法的研究。比较 n = 100; 101 时 CG 算法与 SOR 算法的迭代次数；绘制误差下降曲线以及残量的下降曲线，绘制迭代次数与矩阵阶数的关系。

练习 6.12. 编制预处理 CG 算法，采用 SSOR 做为预处理因子。取定矩阵阶数，考察迭代次数与ω的关系。

二. 目标

6.6 实现 Jacobi 和 Gauss-Seidel 方法，对应不同的停机标准，比较两者的迭代次数，以及在相同阶数的情况下比较两者在算法停止时的真实误差大小。

6.7 实现 SOR 方法，以真实误差作为停机标准，观察松弛因子ω对迭代次数的影响，并且通过迭代次数的大小来确定最佳迭代因子。

6.8 比较上述 Jacobi，Gauss-Seidel 和 SOR 方法，分别绘制误差下降曲线和残量下降曲线（采用对数坐标系，最好采用 2-范数），并按照真实误差作为停机标准绘制迭代次数与矩阵阶数倒数的关系。

6.9 实现变系数的 Richardson 迭代方法。

6.10 对 Jacobi 方法进行半迭代加速。

6.11 实现 CG 算法，并就 n = 100 和 n = 101，将 CG 算法和 SOR 算法进行迭代次数的比较，以及误差下降曲线和残量下降曲线的比较，分别绘制两者迭代次数与矩阵阶数的关系图。

6.12 采用 SSOR 作为预处理因子，编制预处理 CG 算法，并在矩阵阶数一定的情况下，考虑迭代次数与ω的关系。

三. 实现过程

6.6 三种停机标准，这里采用的都是∞-范数，

1.残量停机标准：max(abs(A*x-b)) < TOL

2.相邻差量停机标准，再利用一个变量 xx 来储存迭代上一步产生的 x 值:max(abs(x-xx))<TOL

3.后验误差停机标准，要储存这一步的相邻误差量 del(k)和上一步的相邻误差量 del(k-1): max(abs(power(del(k),2)/(del(k-1)-del(k)))) < TOL

然后分别在这三种停机标准下做出Jacobi和Gauss-Seidel方法的迭代次数和算法停止时的真实误差。

6.7 实现 SOR 方法，迭代次数$\omega\in(0,2)$，先取步长为 0.1，做出迭代次数与ω之间的关系，由图中找出迭代次数极值点所在的区间，在此区间内去步长为 0.01，再次做出迭代次数与ω之间的关系，依次下去，找到最佳松弛因子和最小迭代次数的范围

6.8 利用$\omega_{\text{opt}}=2/(1+(1-(\rho(B))^2)^{1/2})$，计算出$\omega_{\text{opt}}$，作为 SOR 方法的最佳松弛因子，然后分别做出三种方法的误差下降曲线，残量下降曲线（这里取的都是∞-范数），以及迭代次数和矩阵阶数倒数的关系。

6.9 编写变系数的 Richardson 迭代方法，通过

$$\begin{cases} x_m = x_{m-1}+\tau_m(b-A*x_{m-1}) \\ 1/\tau_k=((b-a)*\cos\theta_k)/2+(b+a)/2 \text{ 在此处键入公式。} \\ \theta_k=(2k-1)*\pi/(2m) \end{cases}$$

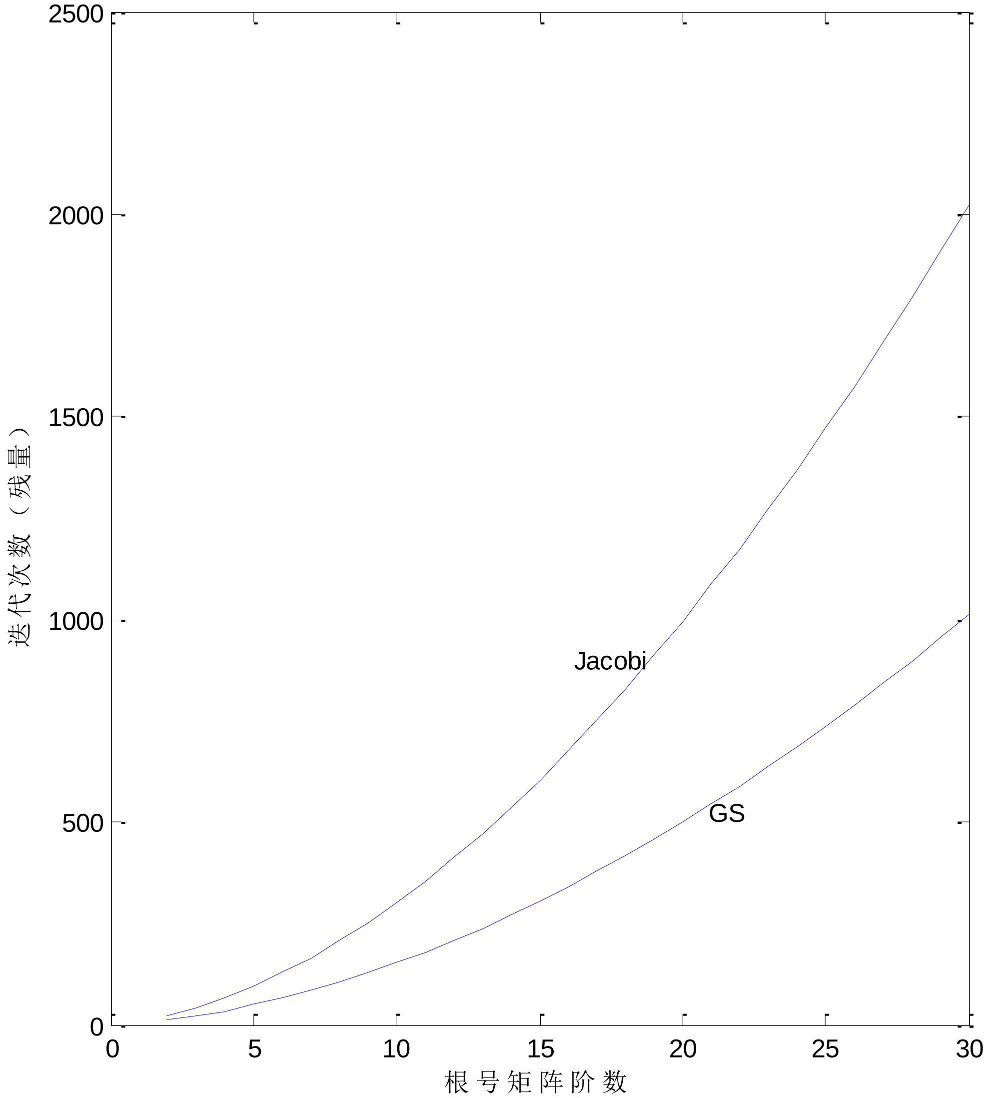
6.10 编写J 法的半迭代加速法的函数，取定矩阵阶数作图：数值误差e 的对数--迭代次数k，残量r 的对数--迭代次数k。

6.11 因为内存关系，SOR算法计算n = 100 和 n = 101 时，计算机内存不够，已经在和计算机的硬盘进行数据交互了，这样得到的数据和只利用计算机内存进行计算的CG算法所得的数据进行比较不大合理，故对于SOR算法只取n = 50时。

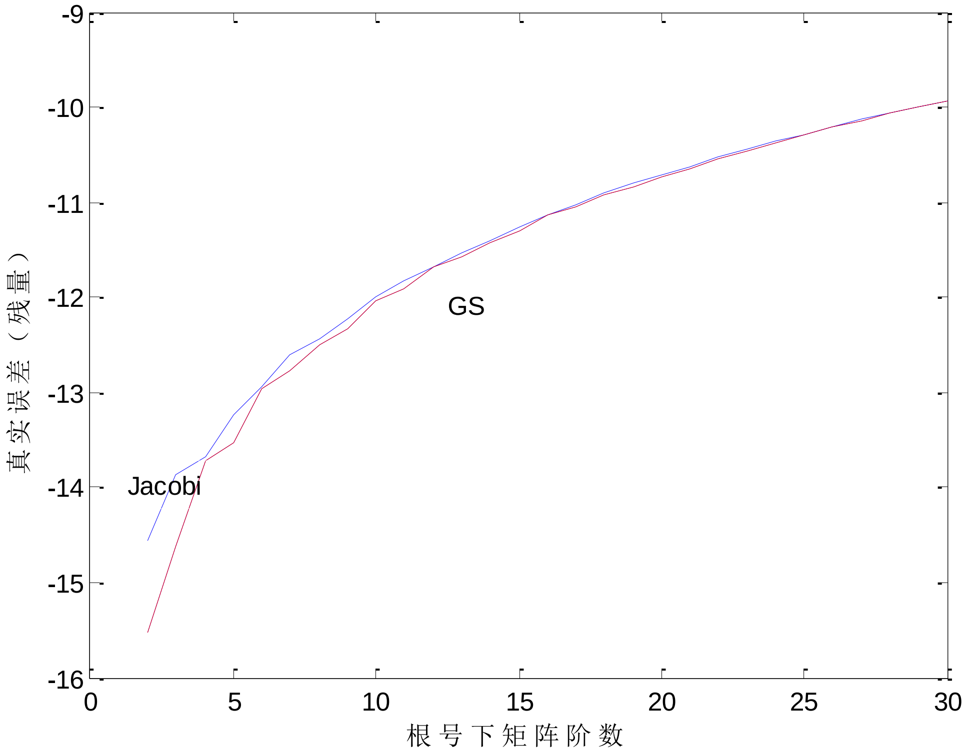
6.12 利用$(D+\omega L)D^{-1}(D+\omega L)^T$得出矩阵M，利用矩阵M作为条件预优矩阵。

四．结果与评价

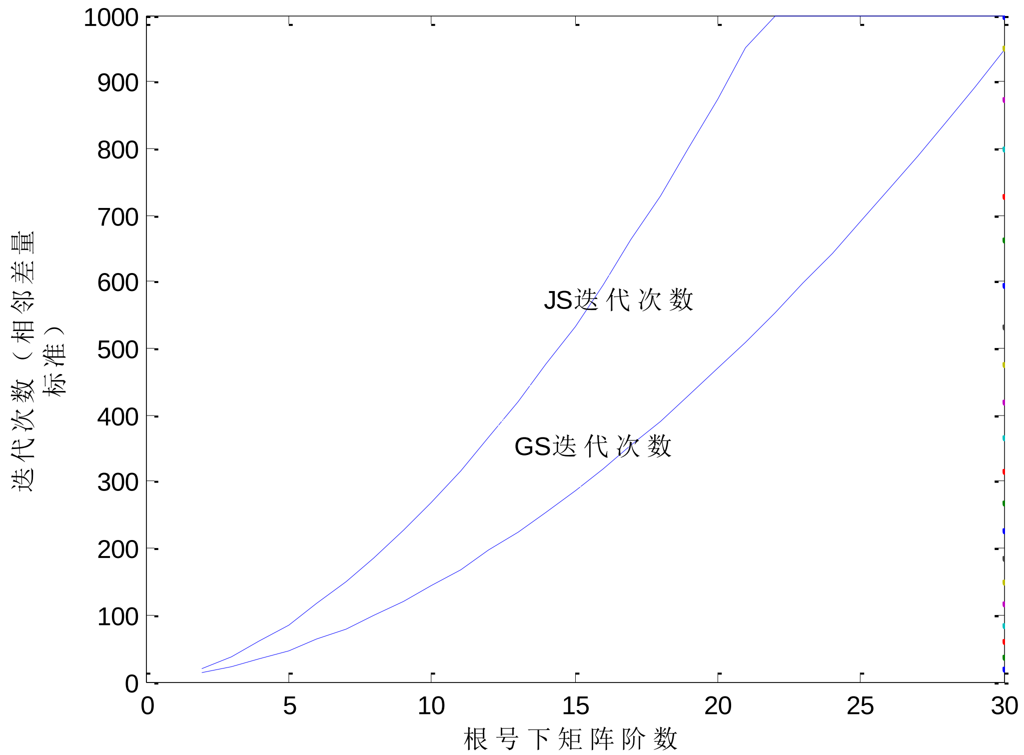
6.6



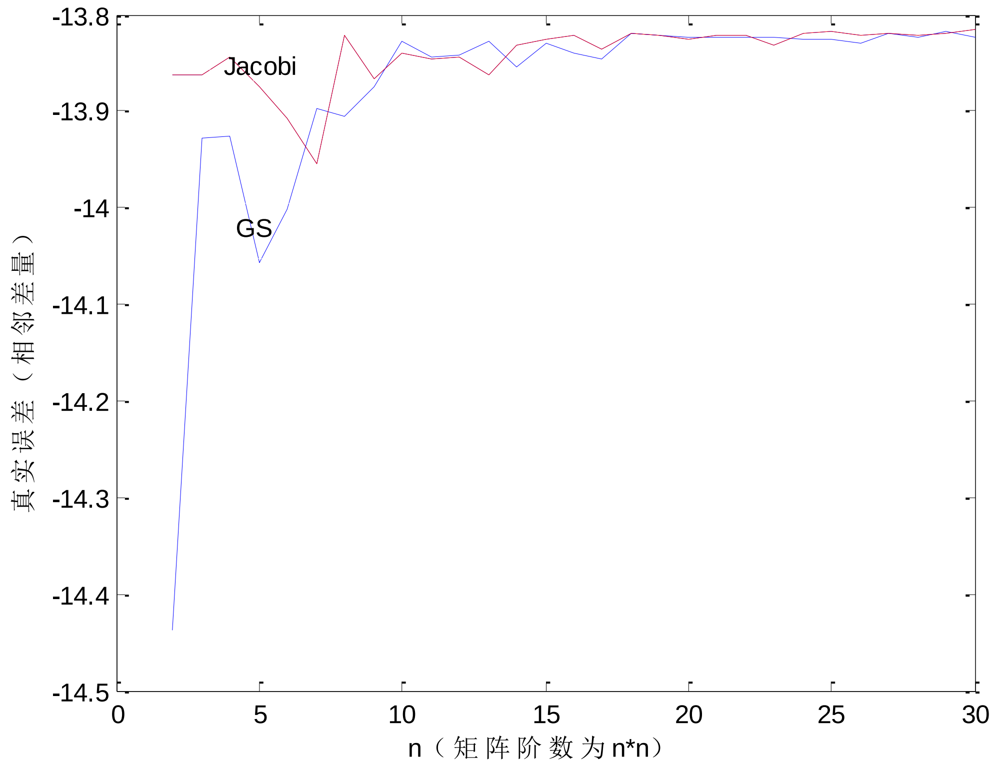
这是两种方法一残量为停机标准所得到的迭代次数的比较图像，由图上大致可以看出GS的收敛速度是Jacobi收敛速度的两倍



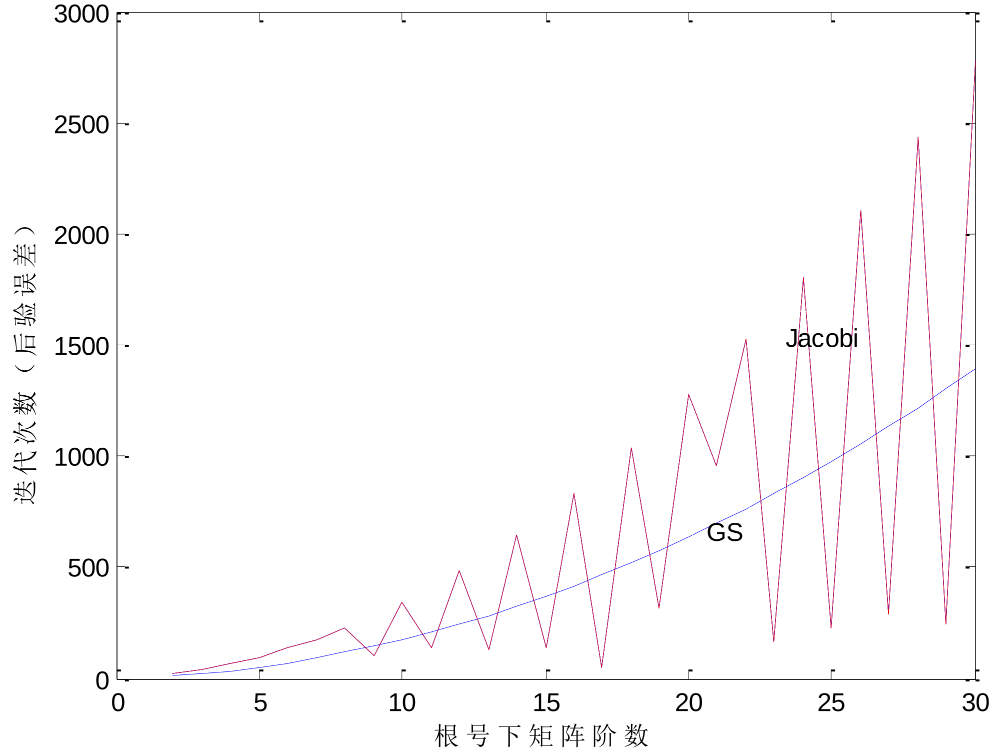
这是两种方法以残量为停机标准时，算法停止时的真实误差。而此时蓝色的Jacobi方法的误差，红色的GS方法的误差，都是在∞-范数下，两者的真实误差都随着矩阵的阶数增大而增大，最终趋于一致。



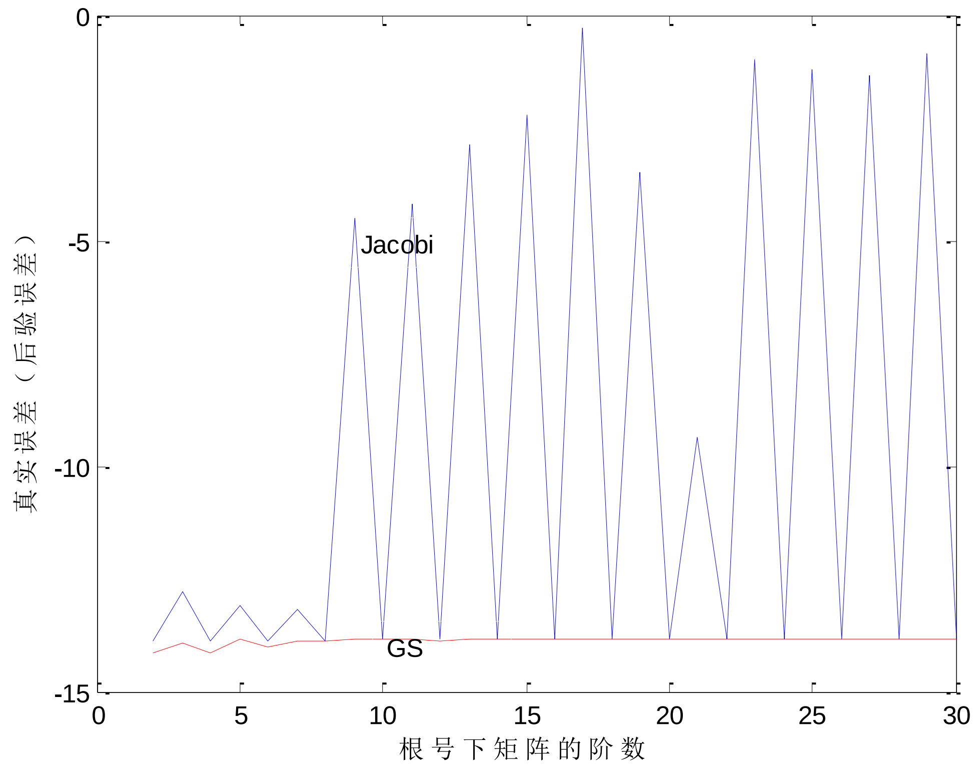
由上图发现，在相邻差量标准下，GS 收敛速度依旧是 Jacobi 收敛速度的两倍，由于取的预设迭代次数的限制，上图中 Jacobi 后来已经超出预设迭代次数了。



这是相邻差量标准下两者最后停机时的真实误差（∞-范数），红色的为 Jacobi，蓝色的为 GS，二者都递增，最后趋于一致。

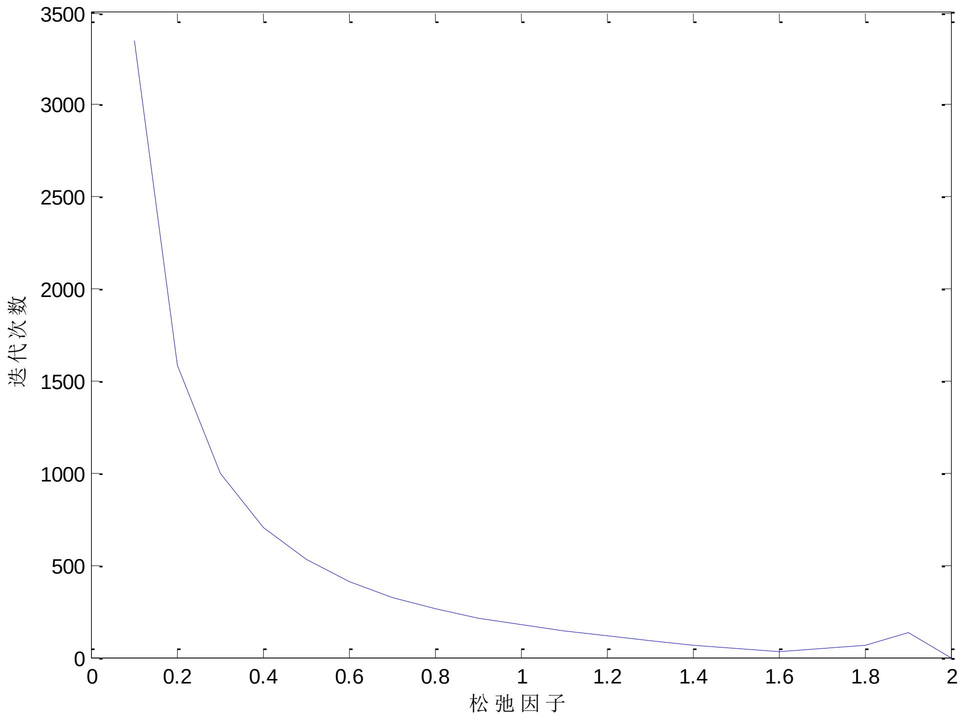


上图为二者在后验误差下的迭代次数的比较，红色的是 Jacobi，蓝色的是 GS，Jacobi 的波动比较大，总体趋势是递增的。

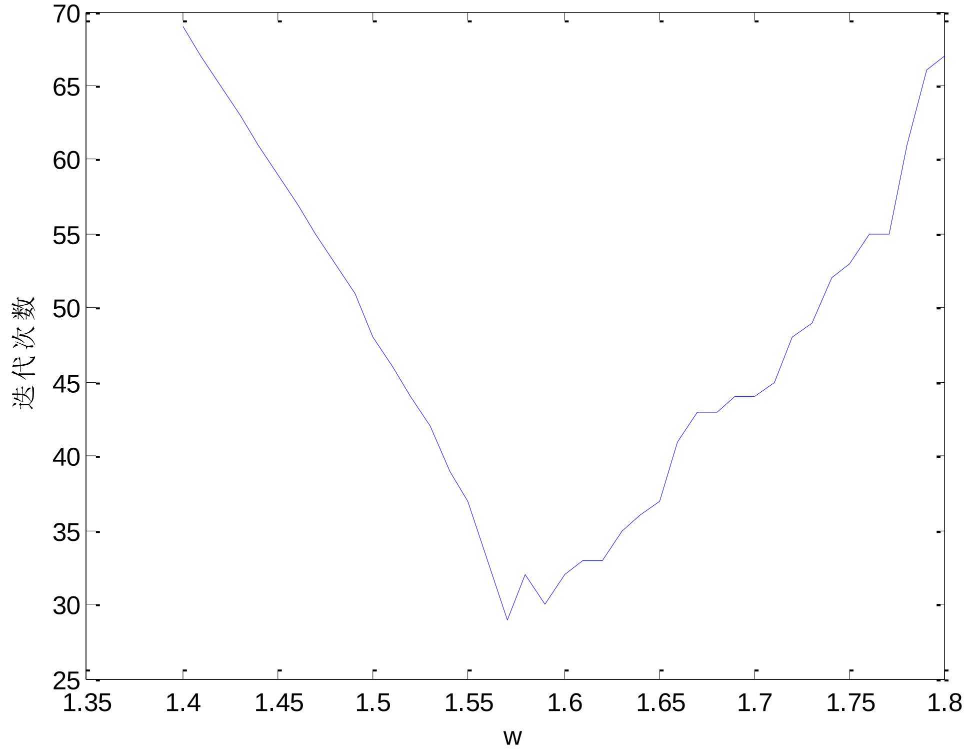


上图是二者在后验误差停机标准下的真实误差，Jacobi 波动大，而 GS 方法则稳定，则在矩阵阶数比较小的时候，后验误差与真实误差比较接近。

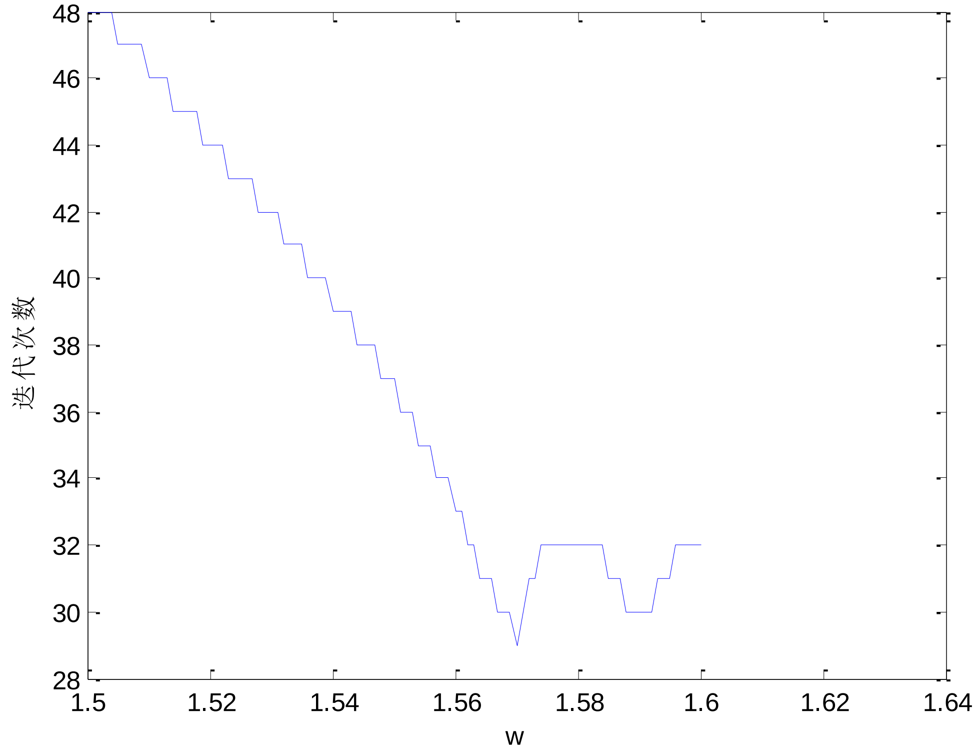
6.7



这是初始 0:0.1:2.0 时，松弛因子对迭代次数的影响，可以看出在[1.5，1.6]时，迭代次数可以取到极小值，因此，对此段区间进行放大。

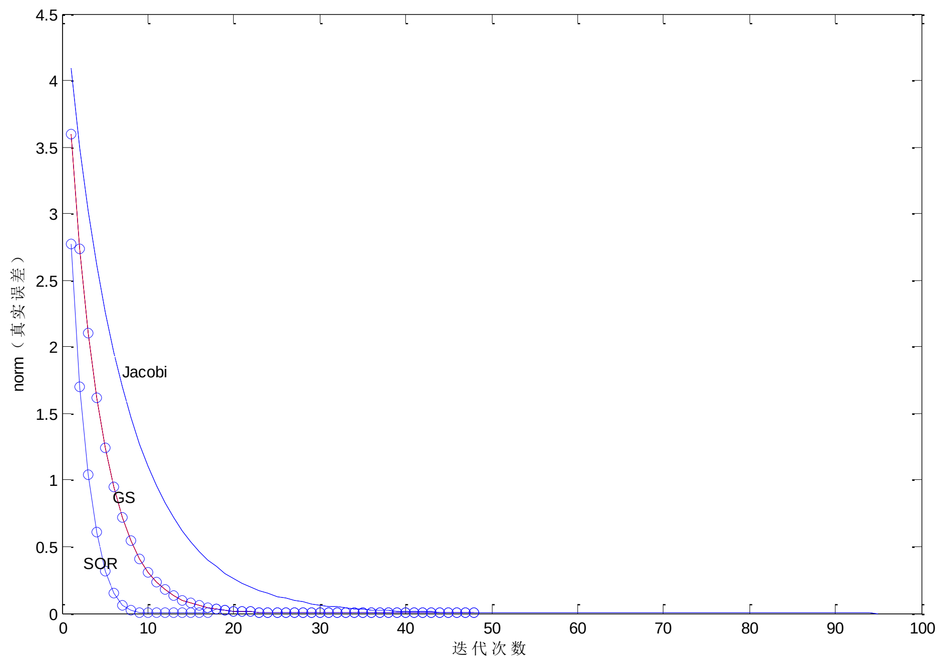


这是放大后区间上迭代次数的图，可以看出在[1.55,1.6]之间，有可能取到ω_{opt}，再次进行区间的放大，

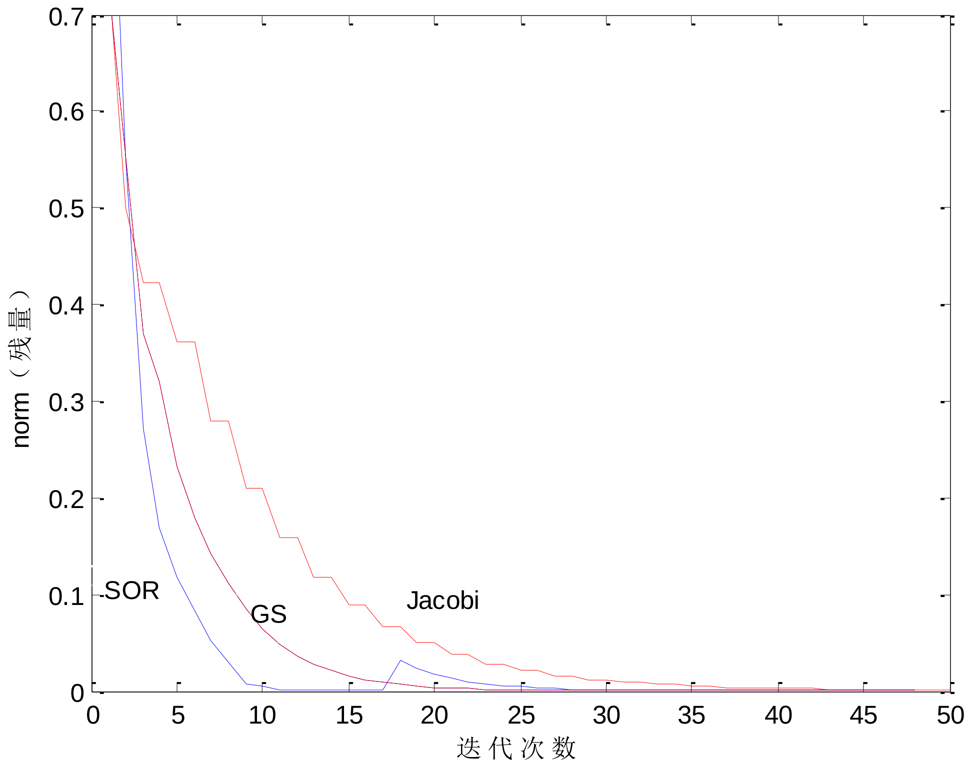


大致能够得到，ω_{opt} = 1.57.

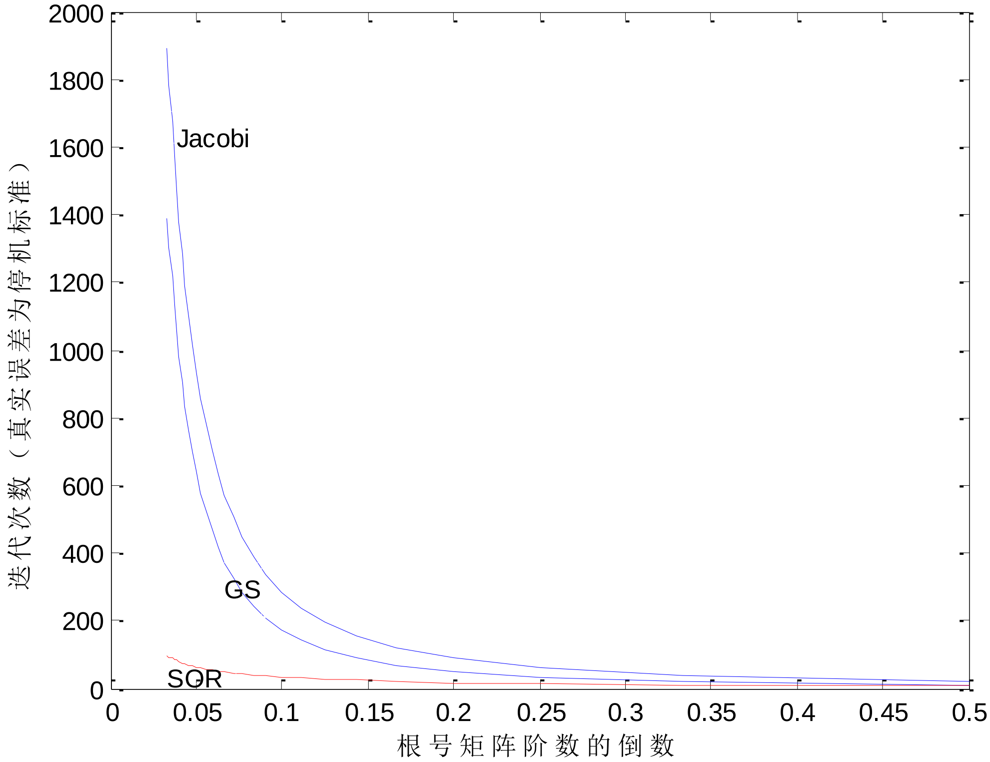
6.8 对于三种方法，取 n = 5



由此图看，这次取得是 2-范数，三种方法的迭代速度一个比一个快，而最后真实误差也是一个比一个小。

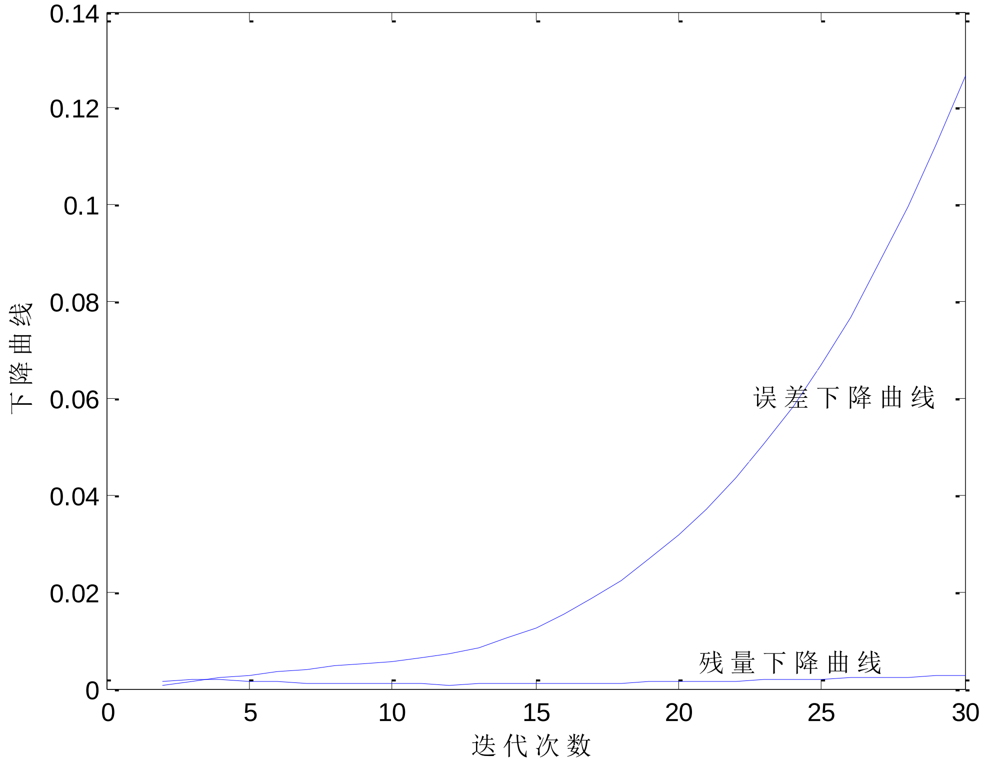


这是三者的残量下降曲线，结论如上图。



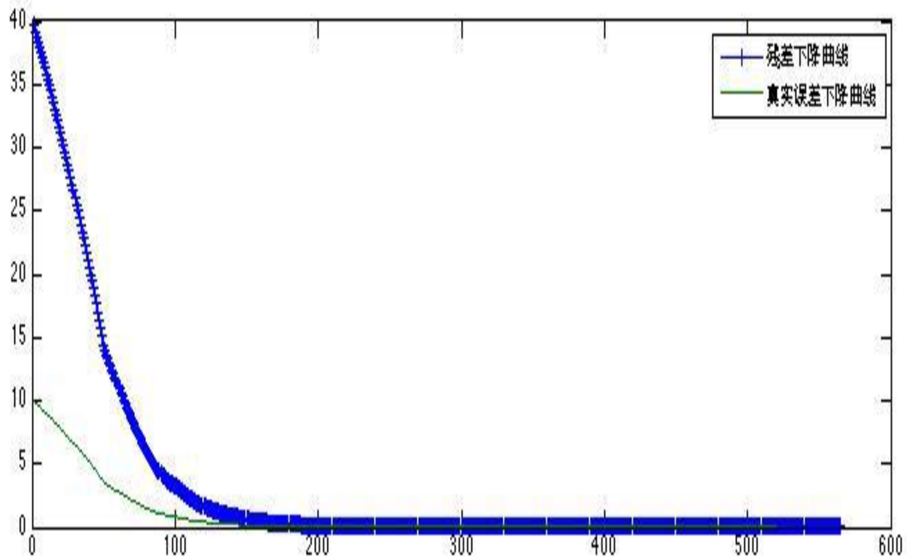
三者迭代次数与 1/（根号下矩阵阶数）之间的关系，SOR 方法收敛速度最快。

6.9



这是取得 2-范数下的变系数的 Richardson 迭代方法的两个下降曲线。真实误差下降曲线是递增的，随着迭代次数的增加，真实误差越来越大，而残量一直稳定。这个误差曲线是有问题的，不可能误差会越来越大。

6. 10

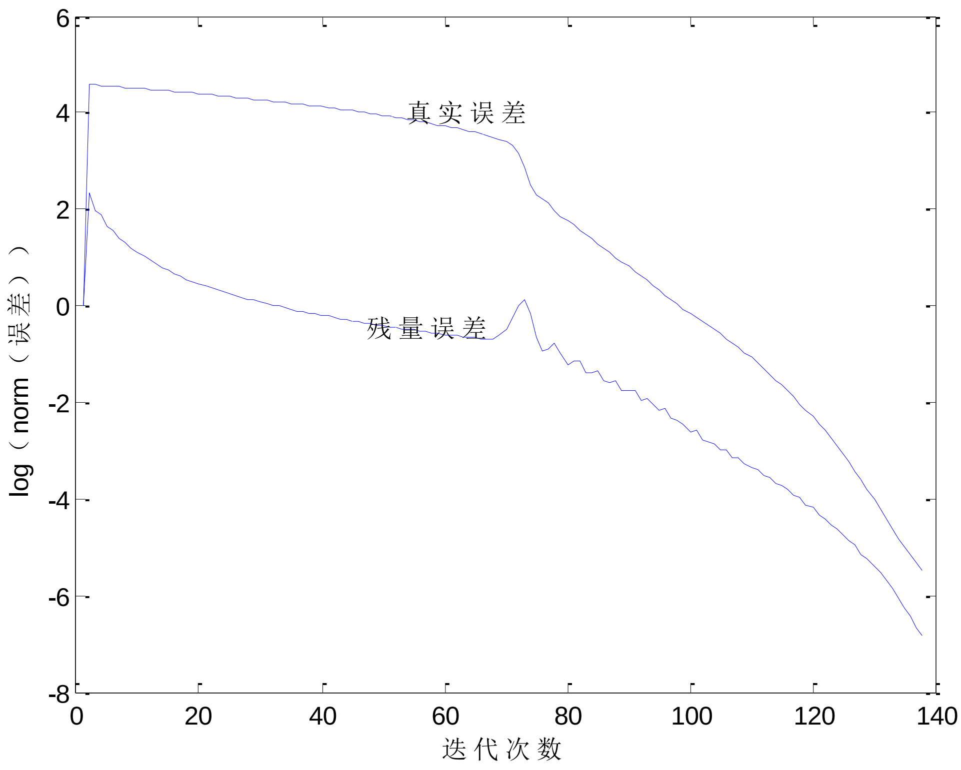


这一个是 Jacobi 方法的半迭代加速。

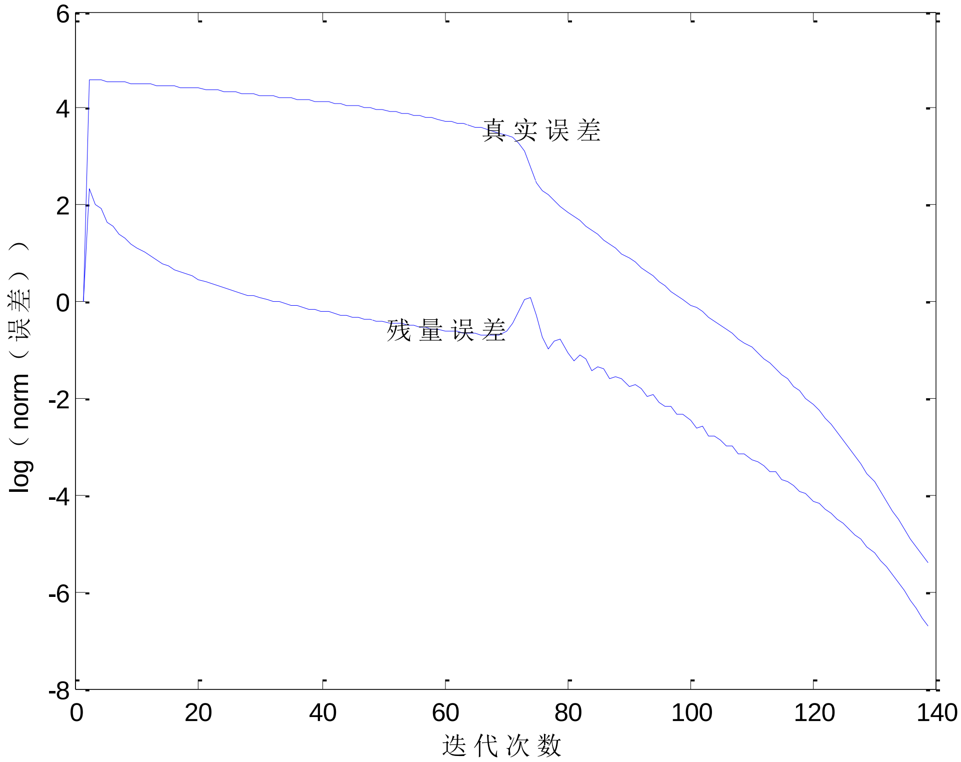
6.11

n=100，CG 算法的迭代次数为 166。SOR 算法迭代步数为 280。 n=101，CG 算法的迭代次数为 168。SOR 算法迭代步数为 288。 在方法论上 CG 方法的迭代次数应该小于 n*n，实际计算时，发现其实仅仅迭代步数小于 n*n，甚至小于 2n。算法的高效性令我震惊。虽然在计算时会有舍入误差，无法得到精确解，但是，CG 方法可以很好面对这种情况并且很 快的给出了近似解。

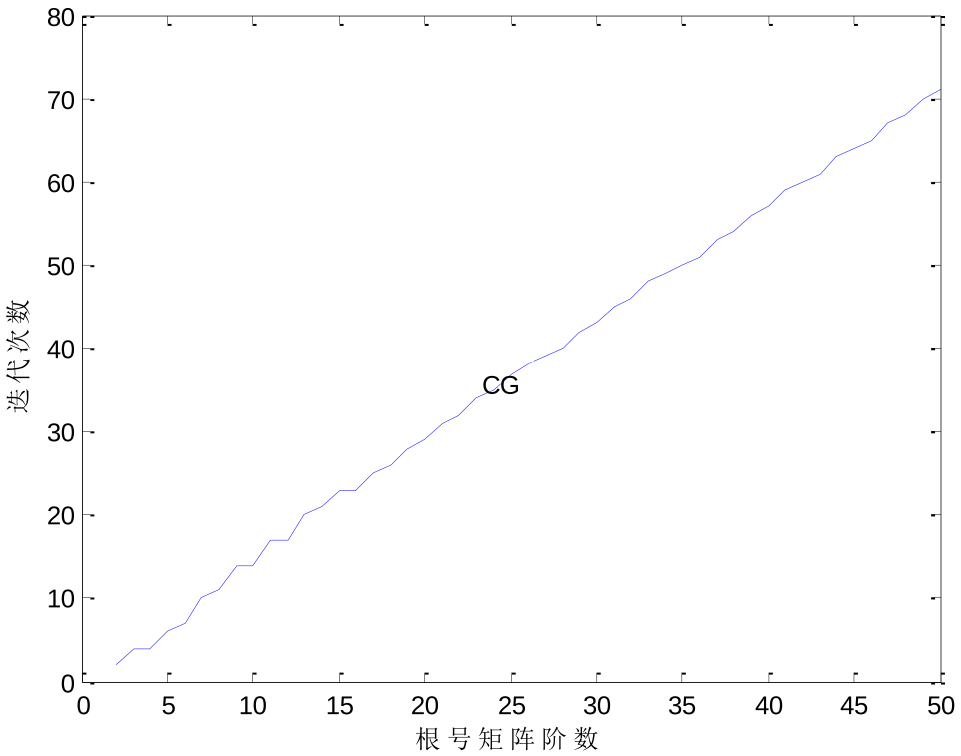
n=100 或 101 时，SOR 算法迭代次数的数量级不 CG 算法是相同的，均为 n 乘以一个常数左右。对比 SOR 方法不 CG 算法，CG 算法比 SOR 方法快，而 且最关键是 CG 算法没有涉及参数的选取，SOR 算法中当 n 比较大时松弛因子ω的选取是一个麻烦的事情。所以下面只给出了 CG 算法的答案，对于 SOR 算法，电脑内存已经不够使用了，只能从硬盘中进行数据的交换，这时，运算所得的结果和 CG 相比，已经没有可比性了。



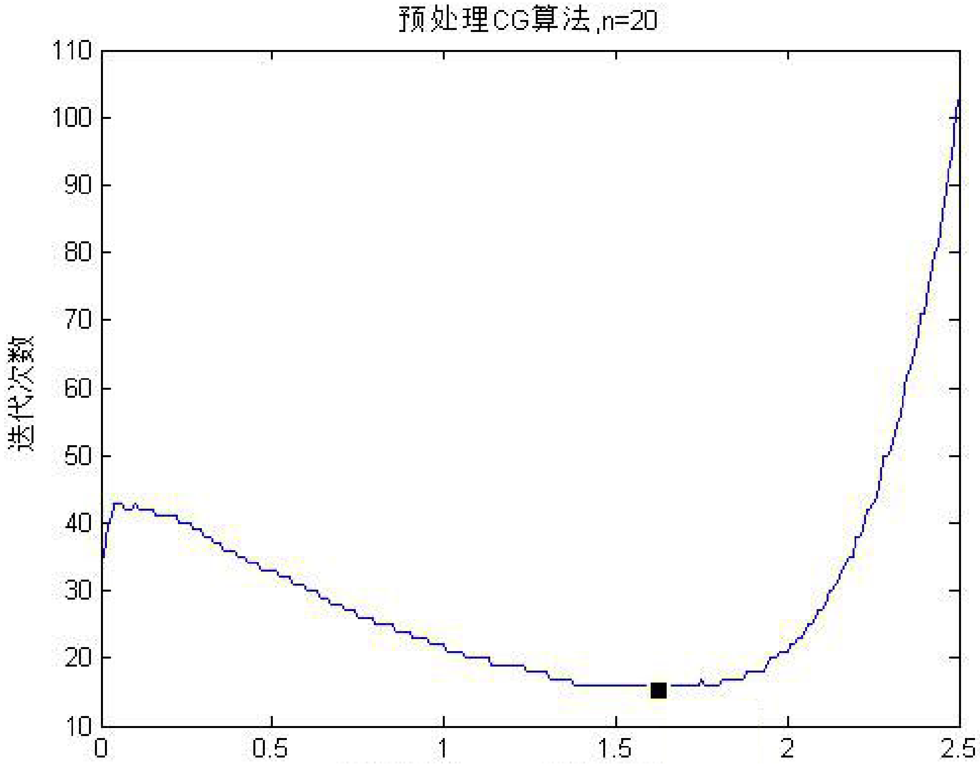
上图是 n = 100 时，两个误差下降曲线（2-范数下），真实误差最后依旧比残量误差大。



上图是 n = 101 的时候，比较上述两个图，发现 n = 101 和 n= 100 的时候，两者的下降趋势是相同的，下图就只是上图往左进行了平移。



6．12



如上图，随着ω的取值的增大，迭代次数呈现了先下降然后上升的趋势，而迭代次数的最低点就是ω_{opt}。

可以看出预处理 CG 算法比 CG 算法要快。

五．结论

1.本报告主要讨论了线性方程组Ax 与b的迭代解法，包括经典的Jacobi迭代解法，Gauss-seidel迭代法，分析了三种不同停机标准对迭代收敛的影响。

2.讨论了SOR超松弛迭代法以及最佳因子的选取，变系数Richardson迭代的推导与因子选取。

3.讨论了 CG 算法，以及一些算法间的比较。