电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

数值分析第二次作业

The second assignment of numerical analysis

课程题目：	数值分析第二次作业
专　　业：	电子信息
姓　　名：	郭元洪
学　　号：	202122140307

目录

1 问题

求解系数矩阵由 16 阶 Hilbert 方程组构成的线性方程组，右端项为 [2877 / 851, 3491 / 1431, 816 / 409, 2035 / 1187, 2155 / 1423, 538 / 395, 1587 / 1279, 573 / 502, 947 / 895,1669 / 1691, 1589 / 1717, 414 / 475, 337 / 409, 905 / 1158, 1272 / 1711, 173 / 244]

要求：

1、Gauss_Sedel 方法；

2、最速下降法

3、共轭梯度法；

4、结果分析与对比。

2 高斯赛德尔方法

设方程组 $Ax = f$ 有唯一解 x^*，将 $Ax = f$ 变形为等价的方程组 $x = B\,x + f$ 由此建立迭代公式

$$x^{(k+1)} = Bx^{(k)} + f \qquad (k = 0,1,2,...)$$

给定初始向量 $x^{(0)}$，按此公式计算的近似解向量序列 $\{x^{(k)}\}$，称此方法为迭代法。若 $\lim\limits_{k\to\infty} x^{(k)} = x^*$，显然有 $x^{(*)} = Bx^{(*)} + f$ 则称迭代法收敛，否则迭代法发散。

对于线性方程组 $\sum\limits_{j=1}^{n} a_{ij}x_j = b_j\,(i = 1,2,...,n)$ Gauss_Sedel 的迭代格式如下所示：

$$x_i^{(k+1)} = \frac{1}{a_{ii}}[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}] \quad (i = 1,2,...n; k = 1,2,...)$$

将上述分量形式转化为矩阵的形式：

$$x^{(k+1)} = D^{-1}(Lx^{(k+1)} + Ux^{(k)} + b)$$

$$= D^{-1}Lx^{(k+1)} + D^{-1}Ux^{(k)} + D^{-1}b$$

整理上面的式子得：

$$x^{(k+1)} = (D-L)^{-1}Ux^{(k)} + (D-L)^{-1}b$$

$$= B_{G-S}x^{(k)} + f_{G-S} \qquad (k = 1,2,...,n)$$

其中 $B_{G-S} = (D-L)^{-1}U$ 称为 Gauss_Sedel 迭代矩阵。

Gauss_Sedel 迭代算法可表述为如下几步：

第一步：输入 $A,b,x^{(0)}$，维数 n，精度 ε，最大容许迭代次数 N；

第二步：置 $k=1$；

第三步：计算 $x_1=\frac{1}{a_{11}}(b_1-\sum_{j=2}^{n}a_{1j}x_j^{(0)})$

$$x_i=\frac{1}{a_{ii}}[b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k)}]\quad(i=2,...n-1)$$

$$x_n=\frac{1}{a_{nn}}(b_n-\sum_{j=1}^{n-1}a_{ij}x_j)$$

第四步：若 $\|x-x^{(0)}\|<\varepsilon$，输出 x，结束；否则转到第五步；

第五步：若 $k<N$，则置 $k\leftarrow k+1,x_i^{(0)}\leftarrow x_i(i=1,2,...,n)$ 转到第三步；否则，输出失败信息，结束。

3 最速下降法

最速下降法是求函数 $f(x)$ 的极小点的一种方法，其基本思想是从 $x^{(0)}$ 出发，寻找一个点 $x^{(1)}$，该点的函数值比 $f(x^{(0)})$ 更小。根据多元微积分理论，$f(x)$ 在点 x 下降最快的方向是该点的负梯度方向，即 $-gradf(x)$

对于线性方程组 $Ax=b$，构造二次函数 $f(x)=\frac{1}{2}x^TAx-b^Tx$。其负梯度可表示为 $-gradf(x^{(0)})=-(Ax^{(0)}-b)$。记 $r_0=b-Ax^{(0)}$，它既是二次函数 $f(x)$ 的负梯度方向也表示线性方程组的残差向量。如果 $r_0=0$ 则 $x^{(0)}$ 是方程组的解。如果 $r_0\neq 0$ 则构造迭代式 $x^{(k+1)}=x^{(k)}+t_kr_k$，其中 $r_k=b-Ax^{(k)}$ 为 $f(x)$ 在 $x^{(k)}$ 处的负梯度方向也是第 k 次近似解的残差向量，t_k 是沿 r_k 方向求函数 $f(x^{(k)}+tr_k)$ 极小点所确定的步长 $t_k=\frac{r_k^Tr_k}{r_k^TAr_k}$。

求解对称正定方程组的最速下降法算法分为如下三步。

第一步：给定初值解 $x^{(0)}$，$\varepsilon>0$，计算 $r_0=b-Ax^{(0)}$，置 $k=0$；

第二步：计算$t_k=\frac{r_k^T r_k}{r_k^T A r_k}$，$x^{(k+1)}=x^{(k)}+t_k r_k$，$r_{k+1}=b-Ax^{(k+1)}$；

第三步：$k \leftarrow k+1$，如果$\| x^{(k+1)}-x^{(k)} \| > \varepsilon$，则转到第二步；否则输出$x^{(k)}$结束。

4 共轭梯度法

最速下降法选择负梯度方向从局部来看是比较好的，但是从整体来看并不是最好的，对于正定矩阵A，共轭梯度法考虑选用关于A共轭的向量$p_1, p_2, \ldots$代替最速下降法的负梯度方向，使得迭代法经过有限步就可以收敛。共轭梯度法的算法步骤如下。

第一步：给定初值解$x^{(0)}$，$\varepsilon > 0$，计算$r_0=b-Ax^{(0)}$，若$|r_0| \le \varepsilon$转到第二步。

第二步：计算$t_k=\frac{(p_k, r_{k-1})}{(Ap_k, p_k)}$，$x^{(k)}=x^{(k-1)}+t_k p_k$；

第三步：如果$k=n$则结束；否则计算$r_k=b-Ax^{(k)}$，转到第四步；

第四步：若$|r_k| \le \varepsilon$则结束；否则计算$b_{kj}=\frac{(Ap_j, r_k)}{(Ap_j, p_j)}(j=1,\ldots,k)$，计算$p_{k+1}=r_k-\sum_{j=1}^{k} b_{kj} p_j$，$k \leftarrow k+1$转到第二步。

5 结果分析与对比

在本次实验中我们选择$\varepsilon=10^{-4}$，初值$x^{(0)}=0$，分别使用高斯赛德尔方法，最速下降法和共轭梯度法对线性方程组求解。使用高斯赛德尔方法总共迭代了 459 次才达到设定的精度，结果如表 1 所示。

表 1 高斯赛德尔迭代法

迭代次数	迭代解
1	[3.38072855 2.24756545 1.53154572 1.15117252 0.90879679 0.7398471 0.61549044 0.52048644 0.44588059 0.38606243 0.33723522 0.29683751 0.26304948 0.23439823 0.21000843 0.18900277]
2	[0.79424896 2.54430081 1.92393589 1.53357581 1.25767003 1.05180736 0.89287424 0.76709402 0.66559277 0.58244485 0.51335377 0.45534872

	0.40624951 0.36413148 0.32797755 0.29657202]
3	[0.11912302 2.24108377 1.9290251 1.63709384 1.39389832 1.19557393 1.03371094 0.9006372 0.79021277 0.69782651 0.61973784 0.55327974 0.49640963 0.44710875 0.40448108 0.36716953]
…	…
458	[0.99502725 1.01665689 1.05004218 0.93277613 0.93152019 0.96724726 1.00759641 1.03891417 1.05257947 1.06255555 1.04880228 1.02858709 1.01953175 0.97151586 0.95183517 0.91620387]
459	[0.99506547 1.01635981 1.05040055 0.93293212 0.93151627 0.9671532 1.00747234 1.03879807 1.05248177 1.06250527 1.0487739 1.02858416 1.01958979 0.97155618 0.95192259 0.91630412]

对于最速下降法，其迭代次数比高斯赛德尔方法少一些，但还是需要 399 次迭代才能得到方程的解，如表 2 所示。

表 2 最速下降法

迭代次数	迭代解
1	[1.8518977 1.33633981 1.09288267 0.93911848 0.82956241 0.74609112 0.67969343 0.62525561 0.57960702 0.540654 0.50694436 0.47743409 0.45134983 0.42810146 0.40723378 0.38838546]
2	[0.74155322 1.15164602 1.18930441 1.15498636 1.10257439 1.04709046 0.99346078 0.9433 0.89699629 0.85446785 0.81544795 0.77962125 0.74667479 0.71630386 0.688247 0.66226196]
3	[0.85442961 1.21600434 1.24126908 1.20142945 1.14565737 1.08770923 1.03207179 0.98017821 0.93233035 0.88840109 0.84809546 0.81108026 0.7770305 0.74562993 0.71661068 0.68972328]
…	…
398	[0.99877904 1.01501102 0.98254611 0.9801729 0.99124081 1.00378251 1.01351264 1.01928533 1.02085758 1.01929648 1.01443601 1.00702385 0.99836183 0.98737247 0.97573615 0.96317324]
399	[0.99883538 1.01507463 0.98258909 0.98019146 0.99124517 1.00378022 1.01350884 1.01928338 1.0208591 1.01930314 1.01444777 1.00704059 0.99838445 0.9873994 0.97576781 0.96320915]

将每一步的范数取对数绘制范数变化曲线如图 1 所示。

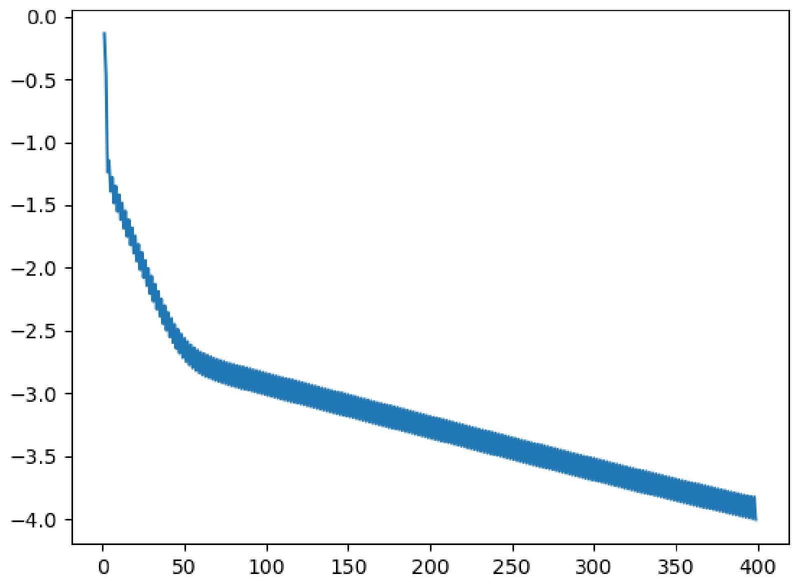

图 1 最速下降法范数变化

对于共轭梯度法，其迭代次数明显小于前两种方法，只需要四次迭代就能得到方程的解，如表 3 所示。

表 3 共轭梯度法

迭代次数	迭代解
1	[1.77731237 1.50717497 1.09171215 0.8615167 0.70718557 0.59484494 0.50916369 0.44174823 0.38746807 0.3430055 0.30601233 0.27488378 0.24845434 0.22570839 0.20610286 0.18900277]
2	[0.63749478 1.46987238 1.385843 1.29129194 1.19728942 1.11042319 1.03237828 0.96293706 0.90125545 0.84640727 0.79742983 0.75355047 0.71411386 0.6784088 0.64608816 0.61662816]
3	[1.02116892 0.94002653 0.94078135 1.00553705 1.04866748 1.0685258 1.07171629 1.06391583 1.04908645 1.0299516 1.00820762 0.98507122 0.96138915 0.93750916 0.91396838 0.89086828]
4	[0.99406736 1.04828064 0.94916611 0.96138405 0.98792565 1.00978433 1.02374632 1.03035731 1.03084882 1.02680855 1.01906235 1.00862343 0.99652786 0.98267694 0.96811509 0.95286562]

上述结果显示，共轭梯度法收敛速度最快只需要四次迭代即可得到方程组的解。究其原因，该方程组的条件数比较大，使得高斯赛德尔方法和最速下降法收敛较慢。同时，共轭梯度法是在最速下降法的基础上改进而来的，使用关于 A 的共轭向量代替最速下降法中的负梯度方向，很好的改善了收敛速度。