线性方程组部分实验题1实习报告

姓名:刘群 学号:2014211591 院系: 地学中心
Email: liu-q14@mails.tsinghua.edu.cn
(本文档由LaTeX编写)

October 14, 2014

1 问题的描述

设$H_n = [h_{ij}] \in \mathbb{R}^{n\times n}$是Hilbert矩阵,即

$$h_{ij} = \frac{1}{i+j-1}.$$

对$n = 2, 3, 4, \cdots$ (根据你的计算机性能选取合适的 n, 建议算到 $n = 15$左右)

(a) 取$x = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^n$, 及$b_n = H_n x$. 再用Gauss消去法和Cholesky分解法来求解$H_n y = b_n$, 看看误差有多大.

(b) 计算条件数: $\mathrm{cond}(H_n)_2$.

(c) 使用某种正则化方法改善(a)中的结果.

2 Gauss消去法和Cholesky分解(没有规则化)

2.1 原理介绍

Gauss消去法是一种较为简单的求解线性代数方程组的方法, 但是它有一个要求, 就是要求在消元的过程中, 对角线元素不能出现0或者很小的数, 否则会造成结果的不稳定. 采用列主元消去法可以很大程度上减少这个问题, 列主元消去法会在每一列中选择一个绝对值最大的数来作为主元, 其实质就是在方程的两边同时乘以初等排列矩阵. 现将该方法介绍如下:

对于矩阵$H \in \mathbb{R}^{n\times n}$,如果$det(H) \neq 0$, 则存在一个排列阵$P(PP^T = PP = I)$,使得$PA = LU$,其中$L$是单位下三角矩阵,$U$ 是上三角矩阵.如果记

$$I_{ij} = \begin{pmatrix} 1 &&&&&&&& \\ & 1 &&&&&&& \\ && \ddots &&&&&& \\ &&& 0 &&& 1 && \\ &&&& \ddots &&&& \\ &&&&& 1 &&& \\ &&&&&& \ddots && \\ &&& 1 &&& 0 && \\ &&&&&&& \ddots & \\ &&&&&&&& 1 \end{pmatrix}$$

表示第i行和第j行进行交换的初等排列阵, 则有在第k步进行消去变换计算$A^{(k+1)}$时, 我们先在$A^{(k)}$的第k列中,从第k个到第n个元素中选取$a_{i_k k}^{(k)}$, 使

$$a_{i_k k}^{(k)} = \max_{k \le i \le n} |a_{ik}^{(k)}|$$

由于A非奇异, 因此有$a_{i_k k}^{(k)} \neq 0$. 我们将$A^{(k)}$的第i_k行与第k行对调, 换行的过程相当于左乘$I_{i_k k}$,然后按照正常的Gauss消元法进行计算. 从而有经过$n-1$步的换行和消去后, A可以化成上三角矩阵U, 有

$$L_{n-1}^{-1} I_{i_{n-1} n-1} \cdots L_2^{-1} I_{i_2 2} L_1^{-1} I_{i_1 1} A = U$$

$$A = I_{i_1 1} L_1 I_{i_2 2} L_2 \cdots I_{i_{n-1} n-1} L_{n-1} U$$

可以证明,

$$P = I_{i_{n-1} n-1} \cdots I_{i_2 2} I_{i_1 1}$$

从而我们可以通过解两个线性方程组$Ly = Pb = b'$和$Ux = y$即可求出x, 其公式如下:

$$\begin{cases} y_1 = b'_1 \\ y_i = b'_i - \sum_{k=1}^{i-1} l_{ik} y_k, & i = 2, \cdots, n, \\ x_n = y_n / u_{nn}, \\ x_i = \left(y_i - \sum_{k=i+1}^{n} u_{ki} x_k \right) \Big/ u_{ii}\ , & i = n-1, \cdots, 1. \end{cases} \tag{1}$$

Cholesky分解的定义如下:如果矩阵$H \in \mathbb{R}^{n \times n}$是对称正定矩阵, 则存在唯一的对角元素为正的下三角矩阵L,使

$$H = LL^T.$$

这称为矩阵的Cholesky分解.并且, 根据矩阵相乘的规则, 即

$$H = LL^T = \begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & \cdots & l_{n1} \\ & l_{22} & \cdots & l_{n2} \\ & & \ddots & \vdots \\ & & & l_{nn} \end{pmatrix}$$

我们可以得出L的计算公式为:

对于$j = 1, 2, \cdots, n$,有

$$\begin{cases} l_{11} = a_{11}^{\frac{1}{2}}, & j = 1 \\ l_{i1} = a_{1i} / l_{11}, & i = 2, 3, \cdots, n \\ l_{jj} = \left(a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 \right)^{\frac{1}{2}}, & j = 2, 3, \cdots, n \\ l_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right) \Big/ l_{jj}\ , & i = j+1, \cdots, n \end{cases} \tag{2}$$

从而我们可以通过解两个线性方程组$Ly = b$和$L^T x = y$即可求出x, 其公式如下:

$$\begin{cases} y_1 = b_1 / l_{11} \\ y_i = \left(b_i - \sum_{k=1}^{i-1} l_{ik} y_k \right) \Big/ l_{ii}\ , & i = 2, 3, \cdots, n, \\ x_n = y_n / l_{nn}, \\ x_i = \left(y_i - \sum_{k=i+1}^{n} l_{ki} x_k \right) \Big/ l_{ii}\ , & i = n-1, n-2, \cdots, 1. \end{cases} \tag{3}$$

2.2 方案设计

这里我们首先采用自己编写的函数lu_pivot_in_col.m对矩阵进行列主元的三角分解, 该函数需要一个方阵作为输入参数, 会返回两个参数, 一个是LU矩阵, 一个是p 向量. 其中LU矩阵存放的是LU分解的L矩阵和U矩阵, 这里我们采用了紧凑存储的方法, 原因是L的对角线元素全是1, 因此我们可以不必存储, 只存储其对角线下方的部分, 而在计算时考虑到L的对角线为1这一性质即可. U由于是上三角矩阵, 对角线元素实际上是U的对角线的结果. 返回值p是一个n维向量, 它告诉我们A的哪些行进行了交换, 利用$b(p)$可以很容易得到交换后的b向量.

之后我们编写了另一个函数Gauss.m来进行求解线性代数方程组$Ax = b$, 该函数有两个参数, 分别是A和b, 返回值是x. 我们利用了公式(1)进行求解.

对于Cholesky分解, 我编写了函数cholesky_llt.m进行求解, 该函数需要一个对称正定的矩阵A作为输入参数, 返回值是下三角矩阵L, 满足$LL^T = A$,在程序中,我们主要采用了公式(2)求解L. 然后我们用另一个函数Cholesky.m来求解$Ax = b$, 该程序主要是用了公式(3)来求解x.

最后我们编写了一个函数Gauss_Cholesky_hw_1.m来调用上面的函数, 并将结果写入了一个Excel表格里.

2.3 计算结果及分析

从Table 1 和Table 2中我们以看出, 当n比较小时,无论是用列主元的Gauss消去法还是Cholesky分解的方法, 计算的结果都比较准确,基本都在1左右, 但是当n比较大时, 可以发现,二者的误差都逐渐增大. 特别是当$n = 14, 15$时,采用Cholesky分解的方法计算出的x与真值的差距是巨大的, 此时计算结果的某些分量已经达到了-1386.8, 是真值的1000多倍, 这个误差是相当大的. 同时, 我们可以看到, 即使采用列主元的Gauss消去法, 当$n = 15$时, 计算结果的误差也已经达到了6.8, 是真值的近7倍, 误差也是相当大的.

Table 1: n=2-15时Gauss消去法计算所得的结果

	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11	n=12	n=13	n=14	n=15
x_1	1	1	1	1	1	1	1	1	0.999999999	0.999999984	0.999999921	0.999999975	0.999999841	0.999999944
x_2	1	1	1	1	1	1.000000001	0.999999999	1.000000001	1.000000055	1.000001675	1.000010096	1.000003249	1.000019848	1.000009542
x_3		1	1	1	1	0.999999995	1.000000012	0.999999976	0.999998824	0.999956045	0.999681244	0.999895713	0.999424804	0.999600436
x_4			1	1	1	1.00000002	0.999999936	1.000000177	1.000010779	1.000496008	1.004358381	1.001470786	1.00630048	1.007183282
x_5				1	1	0.999999964	1.000000172	0.99999935	0.999948201	0.99702249	0.967949318	0.988670157	0.975819634	0.930952619
x_6					1	1.000000031	0.999999757	1.000001324	1.000143329	1.010535419	1.141212086	1.05322407	0.927171589	1.395123538
x_7						0.99999999	1.000000173	0.999998487	0.999763483	0.976936364	0.60559056	0.838019841	2.122456722	-0.415367935
x_8							0.999999951	1.000000908	1.000229733	1.031589063	1.715462185	1.329438053	-4.190702077	4.182171234
x_9								0.999999777	0.999878845	0.973654119	0.159620838	0.548198707	14.58369777	-3.139483841
x_{10}									1.000026752	1.012232893	1.616512671	1.412437251	-21.42182258	2.776223143
x_{11}										0.997575928	0.743287496	0.759926419	24.83668839	4.474986798
x_{12}											1.046315261	1.080571363	-14.87182213	-5.997088687
x_{13}												0.988144421	7.032767487	6.789625421
x_{14}													0	-1.424390455
x_{15}														1.420455005

Table 2: n=2-15时Cholesky分解法计算所得的结果

	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11	n=12	n=13	n=14	n=15
x_1	1	1	1	1	1	1	1	1	1	1.000000003	0.999999956	1.000000024	1.000002451	1.000017654
x_2	1	1	1	1	1	1	1.000000004	1.000000032	0.999999994	0.999999743	1.000005452	0.999995851	0.99961627	0.997243176
x_3		1	1	1	1	0.999999998	0.99999995	0.999999462	1.00000014	1.00000646	0.999831576	1.00017182	1.014866935	1.106468084
x_4			1	1	1.000000001	1.000000009	1.00000027	1.000003838	0.999998708	0.999929629	1.002260478	0.996990027	0.749868125	-0.784817299
x_5				1	0.999999999	0.999999983	0.999999277	0.999985939	1.000006214	1.000409988	0.983640437	1.028072151	3.281402001	17.2130681
x_6					1	1.000000015	1.000001019	1.000028682	0.999982858	0.998586045	1.071085628	0.84320811	-11.63990446	-88.4284312
x_7						0.999999995	0.999999277	0.999967089	1.000028139	1.003027305	0.803852299	1.559881072	46.3531319	320.3718586
x_8							1.000000204	1.000019863	0.999972846	0.995933591	1.352021842	-0.323020933	-108.1650276	-764.200155
x_9								0.999995096	1.000014218	1.003333667	0.590433724	3.092833447	179.8737222	1250.307986
x_{10}									0.999996884	0.998475602	1.297916548	-1.191906961	-197.7776513	-1386.759858
x_{11}										1.000297968	0.876895003	2.458332867	147.0800464	1029.755894
x_{12}											1.02205708	0.441917381	-65.99397825	-487.6930875
x_{13}												1.09352518	17.93676655	140.5586004
x_{14}													-0.712859484	-21.56977351
x_{15}														3.125

下面我们看一下剩余向量和误差向量的情况, y为我们求得的结果, x为方程的真值, 记剩余向量为$r_n = b - Hy$, 误差向量为$\Delta x = y - x$.从Table 3, Table 4, Table 5和Table 6中我们可以看出, 当不采用正则化方法, 单纯采用Gauss消元法和Cholesky分解法时虽然残差向量比较小,但是由于矩阵的条件数很大, 因此误差向量仍然比较大, 特别是Cholesky分解法, 当$n = 14, 15$时, 其误差向量变得非常大, 偏差是真值的1000多倍.

Table 3: n=2-15时Gauss消去法计算结果的残差向量$r = b - Hy$

	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11	n=12	n=13	n=14	n=15
r_1	0	0	0	0	0	4.44089E-16	4.44089E-16	0	4.44089E-16	-4.44089E-16	4.44089E-16	4.44089E-16	4.44089E-16	0
r_2	0	0	2.22045E-16	0	0	0	2.22045E-16	0	0	0	0	0	0	0
r_3		0	0	0	-2.22045E-16	0	2.22045E-16	2.22045E-16	0	2.22045E-16	2.22045E-16	2.22045E-16	8.88178E-16	-2.22045E-16
r_4			1.11022E-16	0	0	2.22045E-16	0	2.22045E-16	-2.22045E-16	2.22045E-16	4.44089E-16	-2.22045E-16	4.44089E-16	0
r_5				-2.22045E-16	0	2.22045E-16	-2.22045E-16	0	0	-2.22045E-16	6.66134E-16	0	0	-2.22045E-16
r_6					0	0	2.22045E-16	0	0	-2.22045E-16	-2.22045E-16	0	2.22045E-16	0
r_7						0	0	-2.22045E-16	-1.11022E-16	0	2.22045E-16	0	-2.22045E-16	-2.22045E-16
r_8							0	1.11022E-16	0	-1.11022E-16	0	0	-2.22045E-16	0
r_9								0	1.11022E-16	1.11022E-16	1.11022E-16	1.11022E-16	1.11022E-16	0
r_{10}									1.11022E-16	-1.11022E-16	0	1.11022E-16	-3.33067E-16	-1.11022E-16
r_{11}										0	1.11022E-16	0	1.11022E-16	0
r_{12}											1.11022E-16	0	0	0
r_{13}												1.11022E-16	-1.11022E-16	-2.22045E-16
r_{14}													0	-1.11022E-16
r_{15}														-1.11022E-16

Table 4: n=2-15时Cholesky分解法计算结果的残差向量$r = b - Hy$

	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11	n=12	n=13	n=14	n=15
r_1	0	2.22045E-16	0	0	0	4.44089E-16	0	0	-4.44089E-16	4.44089E-16	-4.44089E-16	0	-4.44089E-16	2.93099E-14
r_2	0	0	0	0	2.22045E-16	-2.22045E-16	-2.22045E-16	2.22045E-16	0	0	4.44089E-16	-4.44089E-16	-2.22045E-15	4.08562E-14
r_3		0	0	0	0	-2.22045E-16	-2.22045E-16	2.22045E-16	-2.22045E-16	-2.22045E-16	2.22045E-16	0	-5.77316E-15	2.75335E-14
r_4			0	0	-1.11022E-16	0	0	2.22045E-16	0	2.22045E-16	4.44089E-16	2.22045E-16	6.66134E-16	3.01981E-14
r_5				0	0	-1.11022E-16	2.22045E-16	0	0	0	0	2.22045E-16	-4.21885E-15	1.28786E-14
r_6					1.11022E-16	-1.11022E-16	-1.11022E-16	0	-2.22045E-16	2.22045E-16	0	0	6.66134E-16	2.64233E-14
r_7						0	-1.11022E-16	1.11022E-16	-1.11022E-16	-2.22045E-16	2.22045E-16	0	-2.22045E-15	1.11022E-15
r_8							-2.22045E-16	1.11022E-16	0	-2.22045E-16	-2.22045E-16	-2.22045E-16	-2.44249E-15	4.88498E-15
r_9								-1.11022E-16	-1.11022E-16	0	0	-1.11022E-16	-2.55351E-15	2.88658E-15
r_{10}									-1.11022E-16	0	0	0	-4.44089E-16	9.10383E-15
r_{11}										-1.11022E-16	0	2.22045E-16	1.11022E-15	1.40998E-14
r_{12}											1.11022E-16	0	0	6.43929E-15
r_{13}												2.22045E-16	5.55112E-16	9.76996E-15
r_{14}													-6.73905E-14	-4.91607E-13
r_{15}														-2.61735E-12

Table 5: n=2-15时Gauss消去法计算结果的误差$\Delta x = y - x$

	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11	n=12	n=13	n=14	n=15
Δx_1	4.44089E-16	4.44089E-16	-7.10543E-15	-2.73115E-14	-7.32525E-13	-1.3022E-11	1.71303E-11	-1.87108E-11	-6.26144E-10	-1.57617E-08	-7.93263E-08	-2.54816E-08	-1.59191E-07	-5.56988E-08
Δx_2	-6.66134E-16	-9.99201E-16	8.85958E-14	2.16049E-13	2.11939E-11	5.22731E-10	-9.09995E-10	1.39681E-09	5.46692E-08	1.67488E-06	1.00961E-05	3.24905E-06	1.98477E-05	9.54183E-06
Δx_3		0	-2.27041E-13	-1.59872E-13	-1.44818E-10	-5.05746E-09	1.18254E-08	-2.44254E-08	-1.17566E-06	-4.3955E-05	-0.000318756	-0.000104287	-0.000575196	-0.000399564
Δx_4			1.52989E-13	-5.53668E-13	3.79525E-10	1.97236E-08	-6.38286E-08	1.7658E-07	1.07791E-05	0.000496008	0.004358381	0.001470786	0.00630048	0.007183282
Δx_5				5.54667E-13	-4.21393E-10	-3.62458E-08	1.71606E-07	-6.49577E-07	-5.17989E-05	-0.00297751	-0.032050682	-0.011329843	-0.024180366	-0.069047381
Δx_6					1.66833E-10	3.1379E-08	-2.42657E-07	1.3236E-06	0.000143329	0.010535419	0.141212086	0.05322407	-0.072828411	0.395123538
Δx_7						-1.03186E-08	1.72648E-07	-1.51301E-06	-0.000236517	-0.023063636	-0.39440944	-0.161980159	1.122456722	-1.415367935
Δx_8							-4.87136E-08	9.08402E-07	0.000229733	0.031589063	0.715462185	0.329438053	-5.190702077	3.182171234
Δx_9								-2.22957E-07	-0.000121155	-0.026345881	-0.840379162	-0.451801293	13.58369777	-4.139483841
Δx_{10}									2.6752E-05	0.012232893	0.616512671	0.412437251	-22.42182258	1.776223143
Δx_{11}										-0.002424072	-0.256712504	-0.240073581	23.83668839	3.474986798
Δx_{12}											0.046315261	0.080571363	-15.87182213	-6.997088687
Δx_{13}												-0.011855579	6.032767487	5.789625421
Δx_{14}													-1	-2.424390455
Δx_{15}														0.420455005

Table 6: n=2-15时Cholesky分解法计算结果的误差向量$\Delta x = y - x$

	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11	n=12	n=13	n=14	n=15
Δx_1	4.44089E-16	2.22045E-16	-7.99361E-15	3.37508E-14	-1.57718E-12	-5.39546E-12	-7.30218E-11	-4.61957E-10	7.032E-11	2.54623E-09	-4.39606E-08	2.37187E-08	2.45052E-06	1.76542E-05
Δx_2	-7.77156E-16	-7.77156E-16	9.30367E-14	-6.94222E-13	4.43434E-11	2.279E-10	3.87268E-09	3.18366E-08	-6.38709E-09	-2.56847E-07	5.45222E-06	-4.14859E-06	-0.00038373	-0.002756824
Δx_3		2.22045E-16	-2.32703E-13	3.14349E-12	-2.96859E-10	-2.27604E-09	-5.01765E-08	-5.38188E-07	1.39941E-07	6.46042E-06	-0.000168424	0.00017182	0.014866935	0.106468084
Δx_4			1.55209E-13	-4.89664E-12	7.66126E-10	9.07199E-09	2.69923E-07	3.83755E-06	-1.29232E-06	-7.03708E-05	0.002260478	-0.003009973	-0.250131875	-1.784817299
Δx_5				2.44804E-12	-8.40536E-10	-1.69396E-08	-7.2327E-07	-1.4061E-05	6.21446E-06	0.000409988	-0.016359563	0.028072151	2.281402001	16.2130681
Δx_6					3.29605E-10	1.48451E-08	1.01948E-06	2.86823E-05	-1.71424E-05	-0.001413955	0.071085628	-0.15679189	-12.63990446	-89.4284312
Δx_7						-4.92899E-09	-7.23219E-07	-3.29114E-05	2.81389E-05	0.003027305	-0.196147701	0.559881072	45.3531319	319.3718586
Δx_8							2.03509E-07	1.98632E-05	-2.71544E-05	-0.004066409	0.352021842	-1.323020933	-109.1650276	-765.200155
Δx_9								-4.9042E-06	1.42184E-05	0.003333667	-0.409566276	2.092833447	178.8737222	1249.307986
Δx_{10}									-3.11622E-06	-0.001524398	0.297916548	-2.191906961	-198.7776513	-1387.759858
Δx_{11}										0.000297968	-0.123104997	1.458332867	146.0800464	1028.755894
Δx_{12}											0.02205708	-0.558082619	-66.99397825	-488.6930875
Δx_{13}												0.09352518	16.93676655	139.5586004
Δx_{14}													-1.712859484	-22.56977351
Δx_{15}														2.125

3 矩阵的条件数

3.1 原理简介

矩阵的条件数是一种矩阵病态程度的度量, 条件数越大, 矩阵的病态程度越严重.其定义如下:
设矩阵$H \in \mathbb{R}^{n\times n}$,$\det H \neq 0$, 对于矩阵的任意一种从属范数$\|\cdot\|$, 称

$$cond(H) = \|H\|\|H^{-1}\|$$

为H的条件数, 其中常见的有$cond(H)_p = \|H\|_p\|H^{-1}\|_p, p = 1, 2, \infty$.

对于$p = 2$时,其定义如下:

$$\|H\|_2 = \sqrt{\lambda_1},\ 其中\lambda_1为H^*H的最大的特征值, H^*为H的共轭转置.$$

3.2 实验方法及结果分析

我们这里采用了Matlab内置的函数cond来进行计算,其调用格式如下$cond(A, p)$, p表示的是范数的类型, 这里求的是2范数, 因此有$p = 2$. 计算结果如Table 7所示.

从Table 7中我们可以看出,当n很大时,H_n的条件数已经变得很大, 从而说明当n很大时,矩阵H_n已经变得相当病态.因此对此时$H_n x = b$进行求解时, x 的误差会非常大, 为此, 我们需要引入正则化方法来解决病态矩阵的求解问题.

Table 7: n=2-15时的条件数

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15
条件数	19.28	524.06	1.55e4	4.77e5	1.50e7	4.75e8	1.52e10	4.93e11	1.60e13	5.23e14	1.68e16	1.76e18	3.08e17	4.43e17

4 正则化方法的引入

4.1 Tikhonov正则化方法介绍

从上面的分析我们可以看出, 当n变大时,Hilbert矩阵的条件数是非常大的, 因此那些线性方程组是病态方程组. 为了解决Hilbert矩阵条件数过大的问题, 我们就要引入一些正则化方法加以改善. 其中Tikhonov正则化方法是一种非常常用的正则化方法,现将该方法简单介绍如下:

我们要解$Hx = b$,可以转化为求方程

$$(H^*H + \alpha I)x = H^*b \tag{4}$$

其中H^*是H的共轭转置, 即我们只需要在H^*H的对角线上添加α即可.

4.2 实验方法及结果分析

在编写函数时,取$\alpha = 10^{-4}$, 我们将$Hx = b$改写为$(H^*H+\alpha I)x = H^*b$, 然后调用Gauss.m和Cholesky.m两个函数进行求解,具体过程可以参看函数Gauss_Cholesky_hw_1.m.

Table 8 和Table 9是采用正则化方法之后的计算结果,从Table 8 和Table 9可以看出, 引入正则化方法之后, 极大地改善了含有病态矩阵的线性代数方程组的稳定性, 计算结果与真值的差距已经非常小了, 基本都在1附近.

Table 8: n=2-15时Gauss消去法采用正则化方法之后的计算结果

	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11	n=12	n=13	n=14	n=15
x_1	1.000438967	0.992619358	0.995134696	0.998359323	1.001138087	1.003598269	1.005585626	1.00691644	1.007546531	1.007575603	1.007166775	1.006475747	1.005620933	1.004682094
x_2	0.99916193	1.042289352	1.015987988	0.994796648	0.979290524	0.967852149	0.960817279	0.958382388	0.959923433	0.964275489	0.970255341	0.97696141	0.983817512	0.990497701
x_3		0.95870273	1.014589791	1.027876821	1.029851394	1.026392831	1.019618896	1.011230634	1.002699462	0.995006491	0.988594933	0.983530217	0.979690176	0.97689327
x_4			0.967952515	1.007969035	1.027758701	1.037385065	1.039734941	1.0367165	1.030291224	1.022230845	1.013806481	1.005755418	0.998422593	0.991922188
x_5				0.964670288	0.997867221	1.019208575	1.03185877	1.037449063	1.037708282	1.034448448	1.029200914	1.023041803	1.016632629	1.010342294
x_6					0.956378774	0.986882212	1.008586463	1.022730057	1.030572461	1.033650406	1.033463281	1.031212646	1.027744038	1.023608233
x_7						0.948592762	0.977613818	0.99920894	1.014133406	1.023542683	1.028769494	1.031018489	1.031226391	1.030062172
x_8							0.943298706	0.971039677	0.992101907	1.007247589	1.017590121	1.024271344	1.02825881	1.030291863
x_9								0.94074089	0.966933662	0.987046046	1.001960815	1.012724318	1.020297258	1.025455445
x_{10}									0.94021977	0.964528187	0.983411179	0.997806068	1.008644042	1.016714064
x_{11}										0.94078177	0.963049662	0.980610451	0.994352896	1.005063413
x_{12}											0.941666107	0.961952085	0.978240955	0.991306252
x_{13}												0.942429097	0.960928865	0.976071193
x_{14}													0.942882682	0.959842976
x_{15}														0.942992049

Table 9: n=2-15时Cholesky分解法采用正则化方法之后的计算结果

	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11	n=12	n=13	n=14	n=15
x_1	1.000438967	0.992619358	0.995134696	0.998359323	1.001138087	1.003598269	1.005585626	1.00691644	1.007546531	1.007575603	1.007166775	1.006475747	1.005620933	1.004682094
x_2	0.99916193	1.042289352	1.015987988	0.994796648	0.979290524	0.967852149	0.960817279	0.958382388	0.959923433	0.964275489	0.970255341	0.97696141	0.983817512	0.990497701
x_3		0.95870273	1.014589791	1.027876821	1.029851394	1.026392831	1.019618896	1.011230634	1.002699462	0.995006491	0.988594933	0.983530217	0.979690176	0.97689327
x_4			0.967952515	1.007969035	1.027758701	1.037385065	1.039734941	1.0367165	1.030291224	1.022230845	1.013806481	1.005755418	0.998422593	0.991922188
x_5				0.964670288	0.997867221	1.019208575	1.03185877	1.037449063	1.037708282	1.034448448	1.029200914	1.023041803	1.016632629	1.010342294
x_6					0.956378774	0.986882212	1.008586463	1.022730057	1.030572461	1.033650406	1.033463281	1.031212646	1.027744038	1.023608233
x_7						0.948592762	0.977613818	0.99920894	1.014133406	1.023542683	1.028769494	1.031018489	1.031226391	1.030062172
x_8							0.943298706	0.971039677	0.992101907	1.007247589	1.017590121	1.024271344	1.02825881	1.030291863
x_9								0.94074089	0.966933662	0.987046046	1.001960815	1.012724318	1.020297258	1.025455445
x_{10}									0.94021977	0.964528187	0.983411179	0.997806068	1.008644042	1.016714064
x_{11}										0.94078177	0.963049662	0.980610451	0.994352896	1.005063413
x_{12}											0.941666107	0.961952085	0.978240955	0.991306252
x_{13}												0.942429097	0.960928865	0.976071193
x_{14}													0.942882682	0.959842976
x_{15}														0.942992048

Table 10: n=2-15时Gauss消去法正则化后计算结果的残差向量$r = b - Hy$

	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11	n=12	n=13	n=14	n=15
r_1	-0.000193354	-0.000152626	-3.64793E-05	7.70014E-05	0.00014631	0.000182718	0.000199694	0.000204546	0.000200949	0.000190971	0.000176033	0.000157302	0.000135818	0.000112523
r_2	0.000587568	0.00013927	-0.000371582	-0.00068654	-0.000787298	-0.000779593	-0.00072272	-0.000640672	-0.000543539	-0.000436759	-0.000324321	-0.000209697	-9.59872E-05	1.41608E-05
r_3		0.000545975	0.000362803	8.39121E-05	-0.000129102	-0.000266985	-0.0003556	-0.000412883	-0.000448295	-0.000466766	-0.000471208	-0.000463757	-0.000446293	-0.000420624
r_4			0.000797577	0.000610746	0.000382212	0.000192663	4.2481E-05	-7.95925E-05	-0.000181557	-0.000267697	-0.000340064	-0.000399663	-0.000447156	-0.00048322
r_5				0.000918313	0.000708711	0.00051282	0.00034731	0.000205575	8.05956E-05	-3.17128E-05	-0.000133124	-0.000224124	-0.000304669	-0.000374652
r_6					0.000909257	0.00072536	0.000565257	0.000425387	0.000299295	0.000182805	7.40483E-05	-2.73646E-05	-0.000121096	-0.000206596
r_7						0.00086386	0.000717598	0.000589033	0.000472024	0.000362172	0.000257329	0.000156962	6.14409E-05	-2.84932E-05
r_8							0.000822634	0.000708931	0.000605259	0.000506954	0.000411523	0.000318167	0.00022714	0.000139212
r_9								0.000795754	0.000706693	0.000621761	0.000538112	0.000454636	0.000371389	0.000289071
r_{10}									0.000783126	0.000711804	0.000640622	0.000568156	0.000494227	0.000419407
r_{11}										0.000781798	0.000722877	0.000661583	0.000597495	0.000531026
r_{12}											0.000788378	0.000737895	0.000683587	0.0006257
r_{13}												0.000799822	0.000754911	0.000705469
r_{14}													0.000813676	0.00077233
r_{15}														0.000828107

Table 11: n=2-15时Cholesky分解法正则化后计算结果的残差向量$r = b - Hy$

	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11	n=12	n=13	n=14	n=15
r_1	-0.000193354	-0.000152626	-3.64793E-05	7.70014E-05	0.00014631	0.000182718	0.000199694	0.000204546	0.000200949	0.000190971	0.000176033	0.000157302	0.000135818	0.000112523
r_2	0.000587568	0.00013927	-0.000371582	-0.00068654	-0.000787298	-0.000779593	-0.00072272	-0.000640672	-0.000543539	-0.000436759	-0.000324321	-0.000209697	-9.59872E-05	1.41608E-05
r_3		0.000545975	0.000362803	8.39121E-05	-0.000129102	-0.000266985	-0.0003556	-0.000412883	-0.000448295	-0.000466766	-0.000471208	-0.000463757	-0.000446293	-0.000420624
r_4			0.000797577	0.000610746	0.000382212	0.000192663	4.2481E-05	-7.95925E-05	-0.000181557	-0.000267697	-0.000340064	-0.000399663	-0.000447156	-0.00048322
r_5				0.000918313	0.000708711	0.00051282	0.00034731	0.000205575	8.05956E-05	-3.17128E-05	-0.000133124	-0.000224124	-0.000304669	-0.000374652
r_6					0.000909257	0.00072536	0.000565257	0.000425387	0.000299295	0.000182805	7.40483E-05	-2.73646E-05	-0.000121096	-0.000206596
r_7						0.00086386	0.000717598	0.000589033	0.000472024	0.000362172	0.000257329	0.000156962	6.14409E-05	-2.84932E-05
r_8							0.000822634	0.000708931	0.000605259	0.000506954	0.000411523	0.000318167	0.00022714	0.000139212
r_9								0.000795754	0.000706693	0.000621761	0.000538112	0.000454636	0.000371389	0.000289071
r_{10}									0.000783126	0.000711804	0.000640622	0.000568156	0.000494227	0.000419407
r_{11}										0.000781798	0.000722877	0.000661583	0.000597495	0.000531026
r_{12}											0.000788378	0.000737895	0.000683587	0.0006257
r_{13}												0.000799822	0.000754911	0.000705469
r_{14}													0.000813676	0.00077233
r_{15}														0.000828107

Table 12: n=2-15时Gauss消去法正则化后计算结果的误差$\Delta x = y - x$

	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11	n=12	n=13	n=14	n=15
Δx_1	0.004298821	-0.009991912	-0.019417952	-0.019488126	-0.015355294	-0.010422874	-0.005647367	-0.001217068	0.002834614	0.006488677	0.009722412	0.012515934	0.014859627	0.016757701
Δx_2	-0.008210934	0.066041208	0.08115782	0.058334086	0.029012539	0.003561422	-0.017241849	-0.034167283	-0.047847811	-0.058672041	-0.06689584	-0.072732553	-0.076399644	-0.07813359
Δx_3		-0.068628197	0.004347947	0.03793107	0.047284045	0.04635532	0.041139992	0.033958205	0.025835772	0.01733812	0.008843097	0.00062883	-0.00709854	-0.014194716
Δx_4			-0.090295179	-0.02033764	0.018198443	0.038706139	0.049461902	0.054445166	0.055638232	0.054159257	0.050738743	0.045916888	0.040124269	0.033713715
Δx_5				-0.086575995	-0.029497785	0.006022157	0.029001798	0.04416527	0.053958481	0.059765514	0.062474449	0.062731848	0.061055444	0.057881672
Δx_6					-0.082252121	-0.036714928	-0.004960087	0.018003781	0.034815755	0.04696769	0.055400269	0.06078449	0.063652814	0.06445756
Δx_7						-0.082278974	-0.044375911	-0.015622288	0.006670544	0.024018468	0.037361094	0.047353433	0.054507111	0.059256218
Δx_8							-0.085155772	-0.052176377	-0.025691405	-0.004204446	0.013196704	0.027132485	0.038077667	0.046432213
Δx_9								-0.089229706	-0.059580679	-0.03484679	-0.014155129	0.003072943	0.017269504	0.028794211
Δx_{10}									-0.093488959	-0.066235936	-0.042908224	-0.022970389	-0.006033655	0.008220257
Δx_{11}										-0.097394675	-0.071974731	-0.049827927	-0.030606819	-0.014032963
Δx_{12}											-0.100695522	-0.076762076	-0.055651422	-0.03712218
Δx_{13}												-0.103310933	-0.0806477	-0.060480945
Δx_{14}													-0.105260158	-0.08372986
Δx_{15}														-0.106617096

我们进一步分析, 如Table 10, Table 11,Table 12和Table 13 所示, 采用正则化方法之后, 虽然残差向量相比于从前变大了一些, 但是误差向量变得很小了, 因此可以说采用正则化方法之后, 结果更加准确和稳定.

5 实验小结

通过这个实验, 我了解到了含有病态矩阵的线性代数方程组数值求解方法的不稳定性, 此时, 我们可以采用某些正则化方法来使得这种线性代数方程组的求解变得稳定.

6 Matlab源程序

(1) **lu_pivot_in_col.m**

```
function [LU, p] = lu_pivot_in_col(A)
% FUNCTION [LU, P] = lu_pivot_in_col(A)
% LU Decomposition by chosing column pivot
% LU is a matrix consists of L, a unit lower triangular matrix, and U,
    the
% upper triangular matrix, because the diagonal of L is 1, so we don't
% store the diagonal of U at all.
```

Table 13: n=2-15时Cholesky分解法正则化后计算结果的误差向量$\Delta x = y - x$

	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10	n=11	n=12	n=13	n=14	n=15
Δx_1	0.004298821	-0.009991912	-0.019417952	-0.019488126	-0.015355294	-0.010422874	-0.005647367	-0.001217068	0.002834614	0.006488677	0.009722412	0.012515934	0.014859627	0.016757701
Δx_2	-0.008210934	0.066041208	0.08115782	0.058334086	0.029012539	0.003561422	-0.017241849	-0.034167283	-0.047847811	-0.058672041	-0.06689584	-0.072732553	-0.076399644	-0.07813359
Δx_3		-0.068628197	0.004347947	0.03793107	0.047284045	0.04635532	0.041139992	0.033958205	0.025835772	0.01733812	0.008843097	0.00062883	-0.00709854	-0.014194716
Δx_4			-0.090295179	-0.02033764	0.018198443	0.038706139	0.049461902	0.054445166	0.055638232	0.054159257	0.050738743	0.045916888	0.040124269	0.033713715
Δx_5				-0.086575995	-0.029497785	0.006022157	0.029001798	0.04416527	0.053958481	0.059765514	0.062474449	0.062731848	0.061055444	0.057881672
Δx_6					-0.082252121	-0.036714928	-0.004960087	0.018003781	0.034815755	0.04696769	0.055400269	0.06078449	0.063652814	0.06445756
Δx_7						-0.082278974	-0.044375911	-0.015622288	0.006670544	0.024018468	0.037361094	0.047353433	0.054507111	0.059256218
Δx_8							-0.085155772	-0.052176377	-0.025691405	-0.004204446	0.013196704	0.027132485	0.038077667	0.046432213
Δx_9								-0.089229706	-0.059580679	-0.03484679	-0.014155129	0.003072943	0.017269504	0.028794211
Δx_{10}									-0.093488959	-0.066235936	-0.042908224	-0.022970389	-0.006033655	0.008220257
Δx_{11}										-0.097394675	-0.071974731	-0.049827927	-0.030606819	-0.014032963
Δx_{12}											-0.100695522	-0.076762076	-0.055651422	-0.03712218
Δx_{13}												-0.103310933	-0.0806477	-0.060480945
Δx_{14}													-0.105260158	-0.08372986
Δx_{15}														-0.106617096

```
% P is a permutation matrix, so that L*U = P * A.
%
% Author: Qun LIU, via liu-q14@mails.tsinghua.edu.cn
% Time: 2014-10-06

[m, n] = size(A);
% Check if A a square matrix.
if m ~= n
    disp('A must be a square matrix, please check again.')
    return
end

% LU composition and permutation
p = 1:n;
LU = A;
for j = 1:n
    [Max, index] = max(abs(LU(j:n, j)));
    if Max == 0
        disp('The matrix is singular.')
        return
    end
    index = index + j - 1;
    if index ~= j
        tmp = p(j);
        p(j) = p(index);
        p(index) = tmp;

        % Exchange the two rows of the matrix
        tmp = LU(j,:);
        LU(j, :) = LU(index, :);
        LU(index,:) = tmp;
    end
    if j == n
        break;
    end

    LU(j+1:n, j) = LU(j+1:n, j) / LU(j, j);
    for k = j+1:n
        LU(k, j+1:n) = LU(k, j+1:n) - LU(k, j) * LU(j,j+1:n);
    end
    %LU(j+1:n, j+1:n) = LU(j+1:n, j+1:n) - LU(j+1:n, j) * LU(j,j+1:n);
end
```

(2) **Gauss.m**

```
function x = Gauss(A, b)
% Slove Ax=b using Gauss Elimination
% A = (aij)nxn
```

```
% b is a nx1 vector

n = length(b);
%[L, U, P] = lu(A);
[LU, p] = lu_pivot_in_col(A);
b = b(p);
% Ly = P*b
y = zeros(n, 1);
y(1) = b(1);
for i = 2:n
    y(i) = b(i) - LU(i, 1:i-1) * y(1:i-1);
end
% Ux = y
x = zeros(n, 1);
x(n) = y(n) / LU(n, n);
for i = n-1:-1:1
    x(i) = (y(i) - LU(i, i+1:n)* x(i+1:n)) / LU(i, i);
end
end
```

(3) **cholesky_llt.m**

```
function L = cholesky_llt(A)
% Cholesky Decomposition
% Author: Qun LIU via liu-q14@mails.tsinghua.edu.cn
% Time: 2014-10-06

n = length(A);
L = zeros(n);
L(1,1) = A(1,1)^0.5;
L(2:n, 1) = A(1,2:n)/L(1,1);
for j = 2:n-1
    L(j,j) = (A(j,j) - sum(L(j,1:j-1).^2))^0.5;
    for i = j+1:n
        L(i,j) = (A(i,j)- L(i, 1:j-1)*L(j, 1:j-1)') / L(j,j);
    end
end
L(n,n) = (A(n,n) - sum(L(n,1:n-1).^2))^0.5;
```

(4) **Cholesky.m**

```
function x = Cholesky(A, b)
n = length(b);
L = cholesky_llt(A);
%L = chol(A,'lower');
y = zeros(n, 1);
x = zeros(n, 1);
% solve Ly = b
y(1) = b(1) / L(1,1);
for i = 2:n
    y(i) = (b(i) - L(i, 1:i-1) * y(1:i-1)) / L(i, i);
end
% solve Ux = y
U = L';
x(n) = y(n) / U(n, n);
for i = n-1:-1:1
    x(i) = (y(i) - U(i, i+1:n)* x(i+1:n)) / U(i, i);
end
```

(5) **Gauss_Cholesky_hw_1.m**

```
% 线性方程组部分实验题1
% 采用消元法和分解法分别对含有矩阵的方程进行求解GaussCholeskyHilbert
% Author: Qun Liu 刘群 contact via <liu-q14@mails.tsinghua.edu.cn>
% Time: 2014-10-02 14:05
clear;
clc;
% Excels possible range cells
range = {'A','B','C','D','E','F','G','H','I', 'J','K','L','M','N','O','P'
    ,...
        'Q','R','S','T','U','V','W','X','Y','Z','AA','AB','AC','AD'};
file_gauss = 'Results_Gauss.xlsx';
file_chol = 'Results_Chol.xlsx';
file_r_gauss = 'Results_r_Gauss.xlsx';
file_r_chol = 'Results_r_Chol.xlsx';
file_dx_gauss = 'Results_delta_x_Gauss.xlsx';
file_dx_chol ='Results_delta_x_Chol.xlsx';
for n = 2:15
    H = hilb(n);
    x = ones(n, 1);
    b = H * x;

    % solve H*y_gauss = b
    %y_gauss = H \ b;
    y_gauss = Gauss(H, b);
    % solve H*y_Chol = b
    y_chol = Cholesky(H, b);
    % 残差向量
    r_gauss = b - H*y_gauss;
    r_chol = b - H*y_chol;
    % 误差向量
    delta_x_gauss = y_gauss - x;
    delta_x_chol = y_chol - x;
    % Write results to Excels
    xlswrite(file_gauss,y_gauss, [range{n-1},'2:', range{n-1}, num2str(n
        +1)])
    xlswrite(file_gauss,{['n=' num2str(n)]}, [range{n-1} '1:' range{n-1}
         '1']);
    xlswrite(file_chol,y_chol, [range{n-1},'2:', range{n-1}, num2str(n
        +1)])
    xlswrite(file_chol,{['n=' num2str(n)]}, [range{n-1} '1:' range{n-1}
        '1']);

    xlswrite(file_r_gauss, r_gauss, [range{n-1},'2:', range{n-1},
        num2str(n+1)])
    xlswrite(file_r_gauss,{['n=' num2str(n)]}, [range{n-1} '1:' range{n
        -1} '1']);
    xlswrite(file_r_chol, r_chol, [range{n-1},'2:', range{n-1}, num2str(
        n+1)])
    xlswrite(file_r_chol,{['n=' num2str(n)]}, [range{n-1} '1:' range{n
        -1} '1']);

    xlswrite(file_dx_gauss, delta_x_gauss, [range{n-1},'2:', range{n-1},
         num2str(n+1)])
    xlswrite(file_dx_gauss,{['n=' num2str(n)]}, [range{n-1} '1:' range{n
        -1} '1']);
    xlswrite(file_dx_chol, delta_x_chol, [range{n-1},'2:', range{n-1},
```

```
        num2str(n+1)])
    xlswrite(file_dx_chol,{['n=' num2str(n)]}, [range{n-1} '1:' range{n
        -1} '1']);
    % 计算条件数cond(H)2
    condH(n-1) = cond(H, 2);
end

file_gauss_regu = 'Results_Gauss_regu.xlsx';
file_chol_regu = 'Results_Chol_regu.xlsx';
file_r_gauss_regu = 'Results_r_Gauss_regu.xlsx';
file_r_chol_regu = 'Results_r_Chol_regu.xlsx';
file_dx_gauss_regu = 'Results_delta_x_Gauss_regu.xlsx';
file_dx_chol_regu ='Results_delta_x_Chol_regu.xlsx';


% Tikhonov Regularization
alpha = 1e-4;
for n = 2:15
    H = hilb(n);
    x = ones(n, 1);
    b = H'* H * x;
    A = H' * H + alpha * diag(ones(n, 1));
    % Solve A*y_gauss = b
    y_gauss = Gauss(A, b);
    % Solve A*y_chol = b
    y_chol = Cholesky(A, b);
    r_gauss = H*x - H*y_gauss;
    r_chol = H*x - H*y_chol;
    delta_x_gauss = y_gauss - x;
    delta_x_chol = y_chol - x;

    % Write results to Excels
    xlswrite(file_gauss_regu, y_gauss, [range{n-1},'2:', range{n-1},
        num2str(n+1)])
    xlswrite(file_gauss_regu,{['n=' num2str(n)]}, [range{n-1} '1:' range
        {n-1} '1']);
    xlswrite(file_chol_regu, y_chol, [range{n-1},'2:', range{n-1},
        num2str(n+1)])
    xlswrite(file_gauss_regu,{['n=' num2str(n)]}, [range{n-1} '1:' range
        {n-1} '1']);

    xlswrite(file_r_gauss_regu, r_gauss, [range{n-1},'2:', range{n-1},
        num2str(n+1)])
    xlswrite(file_r_gauss_regu,{['n=' num2str(n)]}, [range{n-1} '1:'
        range{n-1} '1']);
    xlswrite(file_r_chol_regu, r_chol, [range{n-1},'2:', range{n-1},
        num2str(n+1)])
    xlswrite(file_r_chol_regu,{['n=' num2str(n)]}, [range{n-1} '1:'
        range{n-1} '1']);

    xlswrite(file_dx_gauss_regu, delta_x_gauss, [range{n-1},'2:', range{
        n-1}, num2str(n+1)])
    xlswrite(file_dx_gauss_regu,{['n=' num2str(n)]}, [range{n-1} '1:'
        range{n-1} '1']);
    xlswrite(file_dx_chol_regu, delta_x_chol, [range{n-1},'2:', range{n
        -1}, num2str(n+1)])
    xlswrite(file_dx_chol_regu,{['n=' num2str(n)]}, [range{n-1} '1:'
        range{n-1} '1']);
```

end