

北京航空航天大学数学科学学院实验报告

课程名称：科学计算通识实验课		实验名称：线性方程组的迭代求解与最小二乘	
实验类型： 演示性实验 <input type="checkbox"/> 验证性实验 <input type="checkbox"/> 综合性实验 <input checked="" type="checkbox"/> 设计性实验 <input type="checkbox"/>			
班级：180924	姓名：陈乐宇	学号：18377471	
实验日期：2021.7.12	指导教师：冯成亮	实验成绩：	
实验环境：（所用仪器设备及软件） Windows + Visual Studio 2019, Ubuntu 18.04.1 + g++			
<p>实验目的与实验内容：</p> <p>通过本实验使学生进一步熟悉个人电脑上 C++ 代码的编写与调试，服务器上的代码编译与运行；熟悉求解线性方程组的 Jacobi 迭代法、Gauss-Seidel 迭代法和 SOR 松弛迭代法；了解以上方法的算法的稳定性与收敛速度特点；熟悉超定线性方程组不可容问题与最小二乘问题的求解算法。</p> <p>实验 1.1：（Jacobi 迭代法求解线性方程组 1）</p> <p>针对方程组</p> $\begin{cases} 8x_1 - 3x_2 + 2x_3 = 20 \\ 4x_1 + 11x_2 - x_3 = 33 \\ 6x_1 + 3x_2 + 12x_3 = 36 \end{cases}$ <p>采用 Jacobi 迭代法进行迭代求解，记录 $\ x^{k+1} - x^k\ _2 < 10^{-3}$ 收敛终止条件下的迭代数据。</p> <p>实验 1.2：（Jacobi 迭代法求解线性方程组 2）</p> <p>针对方程组</p> $\begin{cases} 10x_1 + 3x_2 + 1x_3 = 14 \\ 2x_1 - 10x_2 + 3x_3 = -5 \\ x_1 + 3x_2 + 10x_3 = 14 \end{cases}$ <p>采用 Jacobi 迭代法进行迭代求解，记录 $\ x^{k+1} - x^k\ _2 < 10^{-3}$ 收敛终止条件下的迭代数据。</p> <p>实验 1.3：（Jacobi 迭代法求解线性方程组 3）</p> <p>针对方程组</p> $\begin{cases} 4x_1 - 2x_2 - 4x_3 = 10 \\ -2x_1 + 17x_2 + 10x_3 = 3 \\ -4x_1 + 10x_2 + 9x_3 = -7 \end{cases}$ <p>采用 Jacobi 迭代法进行迭代求解，记录 $\ x^{k+1} - x^k\ _2 < 10^{-5}$ 收敛终止条件下的迭代数据。</p> <p>实验 2.1：（Gauss-Seidel 迭代法求解线性方程组 1）</p> <p>针对方程组</p> $\begin{cases} 8x_1 - 3x_2 + 2x_3 = 20 \\ 4x_1 + 11x_2 - x_3 = 33 \\ 6x_1 + 3x_2 + 12x_3 = 36 \end{cases}$ <p>采用 Gauss-Seidel 迭代法进行迭代求解，记录 $\ x^{k+1} - x^k\ _2 < 10^{-3}$ 收敛终止条件下的迭代数据。</p>			

实验 2.2: (Gauss-Seidel 迭代法求解线性方程组 2)

针对方程组

$$\begin{cases} 10x_1 + 3x_2 + 1x_3 = 14 \\ 2x_1 - 10x_2 + 3x_3 = -5 \\ x_1 + 3x_2 + 10x_3 = 14 \end{cases}$$

采用 Gauss-Seidel 迭代法进行迭代求解, 记录 $\|x^{k+1} - x^k\|_2 < 10^{-3}$ 收敛终止条件下的迭代数据。

实验 2.3: (Gauss-Seidel 迭代法求解线性方程组 3)

针对方程组

$$\begin{cases} 4x_1 - 2x_2 - 4x_3 = 10 \\ -2x_1 + 17x_2 + 10x_3 = 3 \\ -4x_1 + 10x_2 + 9x_3 = -7 \end{cases}$$

采用 Gauss-Seidel 迭代法进行迭代求解, 记录 $\|x^{k+1} - x^k\|_2 < 10^{-5}$ 收敛终止条件下的迭代数据。

实验 3.1: (SOR 松弛迭代法求解线性方程组 1)

针对方程组

$$\begin{cases} 8x_1 - 3x_2 + 2x_3 = 20 \\ 4x_1 + 11x_2 - x_3 = 33 \\ 6x_1 + 3x_2 + 12x_3 = 36 \end{cases}$$

采用 SOR 松弛迭代法 ($\omega = 1.46$) 进行迭代求解, 记录 $\|x^{k+1} - x^k\|_2 < 10^{-3}$ 收敛终止条件下的迭代数据。

实验 3.2: (SOR 松弛迭代法求解线性方程组 2)

针对方程组

$$\begin{cases} 10x_1 + 3x_2 + 1x_3 = 14 \\ 2x_1 - 10x_2 + 3x_3 = -5 \\ x_1 + 3x_2 + 10x_3 = 14 \end{cases}$$

采用 SOR 松弛迭代法 ($\omega = 1.46$) 进行迭代求解, 记录 $\|x^{k+1} - x^k\|_2 < 10^{-3}$ 收敛终止条件下的迭代数据。

实验 3.3: (SOR 松弛迭代法求解线性方程组 3)

针对方程组

$$\begin{cases} 4x_1 - 2x_2 - 4x_3 = 10 \\ -2x_1 + 17x_2 + 10x_3 = 3 \\ -4x_1 + 10x_2 + 9x_3 = -7 \end{cases}$$

采用 SOR 松弛迭代法 ($\omega = 1.46$) 进行迭代求解, 记录 $\|x^{k+1} - x^k\|_2 < 10^{-5}$ 收敛终止条件下的迭代数据。

实验 4.1: (超定线性方程组不可容问题的求解)

针对超定线性方程组

$$\begin{cases} x_1 + 2x_2 + 4x_3 = -1 \\ 2x_1 + x_2 + x_3 = 4 \\ x_1 + x_2 + 2x_3 = 2 \\ x_1 - x_2 - 2x_3 = 1 \end{cases}$$

采用最小二乘方法进行求解。

实验 4.2: (线性最小二乘问题的求解)

针对数据组

i	x	y
1	25	110
2	27	115
3	31	155
4	33	160
5	35	180

采用线性最小二乘方法进行求解其线性拟合函数 $y = f(x) = ax + b$ 。

实验过程与结果:

```
work1@ws1:~/ChenLeyu/class3$ ./test1
use Jacobi Iterative Method to solve linear equations:
test1.1
k      x[1]      x[2]      x[3]      error
0      0.0000      0.0000      0.0000
1      2.5000      3.0000      3.0000      4.924429
2      2.8750      2.3636      1.0000      2.132037
3      3.1364      2.0455      0.9716      0.412744
4      3.0241      1.9478      0.9205      0.157282
5      3.0003      1.9840      1.0010      0.091419
6      2.9938      2.0000      1.0038      0.017518
7      2.9990      2.0026      1.0031      0.005946
8      3.0002      2.0006      0.9998      0.004024
9      3.0003      1.9999      0.9997      0.000736

test1.2
k      x[1]      x[2]      x[3]      error
0      0.0000      0.0000      0.0000
1      1.4000      0.5000      1.4000      2.042058
2      1.1100      1.2000      1.1100      0.811295
3      0.9290      1.0550      0.9290      0.294189
4      0.9906      0.9645      0.9906      0.125616
5      1.0116      0.9953      1.0116      0.042776
6      1.0003      1.0058      1.0003      0.019165
7      0.9982      1.0001      0.9982      0.006345
8      1.0001      0.9991      1.0001      0.002873
9      1.0003      1.0001      1.0003      0.000964

test1.3
k      x[1]      x[2]      x[3]      error
0      0.000000      0.000000      0.00000000
1      2.500000      0.176471      -0.777778      2.62413417
2      1.810458      0.928105      0.137255      1.37029460
3      3.101307      0.308727      -1.004357      1.83117431
4      1.650006      1.132129      0.257551      2.09205041
5      3.323615      0.219089      -1.302362      2.46331795
6      1.307182      1.333580      0.455953      2.89823437
7      3.622742      0.062049      -1.678563      3.39628702
8      0.852462      1.590066      0.763386      3.99654918
9      4.058419      -0.172291      -2.165646      4.68650073
10     0.248209      1.927841      1.217398      5.51119164
11     4.681318      -0.510445      -2.809508      6.46634852
12     -0.564730      2.379866      1.869969      7.60081729
13     5.559902      -0.989950      -3.673064      8.92143438
14     -1.668039      2.991203      2.793234      10.48359291
15     6.788835      -1.662848      -4.842687      12.30797342
16     -3.174111      3.823797      4.087092      14.46044677
17     8.498990      -2.601126      -6.437157      16.97942027
18     -5.237720      4.962914      5.889691      19.94650485
19     10.871148      -3.904256      -8.620003      23.42335415
20     -8.072130      6.526019      8.391905      27.51442235

150      -12323880264.042952      6784761893.537353      11082053146.740456 3
3156668200.67671967
```

由截图可看出使用 Jacobi 迭代法，方程 1 与方程 2 均可解出，方程 3 的误差则会越来越大。

```
work1@ws1:~/ChenLeyu/class3$ ./test2
use Gauss-Seidel Iterative Method to solve linear equations:
test2.1
```

k	x[1]	x[2]	x[3]	error
0	0.0000	0.0000	0.0000	
1	2.5000	2.0909	1.2273	3.482542
2	2.9773	2.0289	1.0041	0.530493
3	3.0098	1.9968	0.9959	0.046459
4	2.9998	1.9997	1.0002	0.011236
5	2.9998	2.0001	1.0001	0.000397

```

use Gauss-Seidel Iterative Method to solve linear equations:
test2.2
```

k	x[1]	x[2]	x[3]	error
0	0.0000	0.0000	0.0000	
1	1.4000	0.7800	1.0260	1.902913
2	1.0634	1.0205	0.9875	0.415465
3	0.9951	0.9953	1.0019	0.074207
4	1.0012	1.0008	0.9996	0.008565
5	0.9998	0.9998	1.0001	0.001785
6	1.0000	1.0000	1.0000	0.000316

39	2.004437	0.996858	-0.994537	0.00107984
40	2.003892	0.997244	-0.995209	0.00094714
41	2.003414	0.997583	-0.995797	0.00083074
42	2.002994	0.997880	-0.996314	0.00072865
43	2.002626	0.998141	-0.996767	0.00063911
44	2.002303	0.998369	-0.997164	0.00056057
45	2.002020	0.998570	-0.997513	0.00049168
46	2.001772	0.998745	-0.997818	0.00043126
47	2.001554	0.998900	-0.998086	0.00037826
48	2.001363	0.999035	-0.998322	0.00033178
49	2.001196	0.999153	-0.998528	0.00029100
50	2.001049	0.999257	-0.998709	0.00025524
51	2.000920	0.999349	-0.998867	0.00022388
52	2.000807	0.999429	-0.999007	0.00019636
53	2.000708	0.999499	-0.999129	0.00017223
54	2.000621	0.999560	-0.999236	0.00015107
55	2.000544	0.999615	-0.999330	0.00013250
56	2.000478	0.999662	-0.999412	0.00011622
57	2.000419	0.999703	-0.999484	0.00010194
58	2.000367	0.999740	-0.999548	0.00008941
59	2.000322	0.999772	-0.999603	0.00007842
60	2.000283	0.999800	-0.999652	0.00006879
61	2.000248	0.999824	-0.999695	0.00006033
62	2.000217	0.999846	-0.999732	0.00005292
63	2.000191	0.999865	-0.999765	0.00004641
64	2.000167	0.999882	-0.999794	0.00004071
65	2.000147	0.999896	-0.999819	0.00003571
66	2.000129	0.999909	-0.999842	0.00003132
67	2.000113	0.999920	-0.999861	0.00002747
68	2.000099	0.999930	-0.999878	0.00002410
69	2.000087	0.999939	-0.999893	0.00002113
70	2.000076	0.999946	-0.999906	0.00001854
71	2.000067	0.999953	-0.999918	0.00001626
72	2.000059	0.999959	-0.999928	0.00001426
73	2.000051	0.999964	-0.999937	0.00001251
74	2.000045	0.999968	-0.999944	0.00001097
75	2.000040	0.999972	-0.999951	0.00000962

```

use Gauss-Seidel Iterative Method to solve linear equations:
test2.3
k      x[1]      x[2]      x[3]      error
0      0.000000  0.000000  0.00000000  0.00000000
1      2.500000  0.470588  -0.189542    2.55095661
2      2.545752  0.587466  -0.299073    0.16658524
3      2.494660  0.645885  -0.386690    0.11704656
4      2.436252  0.690553  -0.462280    0.10545418
5      2.382996  0.728753  -0.528394    0.09309332
6      2.335983  0.762112  -0.586354    0.08174670
7      2.294702  0.791350  -0.637188    0.07171492
8      2.258487  0.816991  -0.681774    0.06290398
9      2.226722  0.839481  -0.720881    0.05517403
10     2.198860  0.859207  -0.755182    0.04839375
11     2.174422  0.876509  -0.785267    0.04244665
12     2.152987  0.891685  -0.811656    0.03723039
13     2.134187  0.904996  -0.834801    0.03265515
14     2.117697  0.916671  -0.855103    0.02864216
15     2.103233  0.926911  -0.872909    0.02512233
16     2.090547  0.935893  -0.888527    0.02203505
17     2.079419  0.943771  -0.902226    0.01932716
18     2.069659  0.950681  -0.914242    0.01695205
19     2.061099  0.956742  -0.924780    0.01486881
20     2.053591  0.962058  -0.934024    0.01304159
21     2.047005  0.966721  -0.942132    0.01143891
22     2.041228  0.970810  -0.949243    0.01003318
23     2.036162  0.974397  -0.955481    0.00880021
24     2.031718  0.977544  -0.960952    0.00771875
25     2.027820  0.980303  -0.965750    0.00677019
26     2.024401  0.982724  -0.969959    0.00593820
27     2.021403  0.984847  -0.973651    0.00520846
28     2.018772  0.986709  -0.976889    0.00456839
29     2.016466  0.988342  -0.979729    0.00400698
30     2.014442  0.989775  -0.982220    0.00351457
31     2.012667  0.991032  -0.984405    0.00308266
32     2.011111  0.992134  -0.986322    0.00270383
33     2.009745  0.993100  -0.988003    0.00237156
34     2.008548  0.993948  -0.989477    0.00208012
35     2.007497  0.994692  -0.990770    0.00182449
36     2.006576  0.995344  -0.991904    0.00160028
37     2.005768  0.995916  -0.992899    0.00140362
38     2.005059  0.996418  -0.993772    0.00123113

```

使用 Gauss-Seidal 迭代法，三个方程均可解出。

```

work1@ws1:~/ChenLeyu/class3$ ./test3
use SOR Method to solve linear equations:
test3.1
k      x[1]      x[2]      x[3]      error
0      0.0000  0.0000  0.0000    0.00000000
1      3.6500  2.4422  0.8241    4.468322
2      3.0073  1.7694  1.1598    0.989142
3      2.8121  2.2271  0.9808    0.528790
4      3.2178  1.7774  0.9311    0.607689
5      2.8031  2.1978  1.1032    0.615091
6      3.1612  1.8371  0.8943    0.549520
7      2.8753  2.1271  1.0933    0.453246
8      3.0929  1.9046  0.9241    0.354278
9      2.9327  2.0695  1.0587    0.266412
10     3.0476  1.9505  0.9563    0.194499
11     2.9670  2.0345  1.0316    0.138659
12     3.0226  1.9764  0.9776    0.096894
13     2.9848  2.0160  1.0155    0.066538
14     3.0100  1.9894  0.9894    0.044978
15     2.9934  2.0070  1.0071    0.029962
16     3.0042  1.9955  0.9953    0.019681
17     2.9973  2.0029  1.0031    0.012751
18     3.0017  1.9982  0.9980    0.008146
19     2.9990  2.0011  1.0013    0.005129
20     3.0006  1.9993  0.9992    0.003180
21     2.9996  2.0004  1.0005    0.001938
22     3.0002  1.9998  0.9997    0.001158
23     2.9999  2.0001  1.0002    0.000676

```

use SOR Method to solve linear equations:

test3.2

k	x[1]	x[2]	x[3]	error
0	0.0000	0.0000	0.0000	
1	2.0440	1.3268	1.1644	2.700801
2	0.3526	0.7326	1.1360	1.792975
3	1.3951	1.2979	0.7493	1.247339
4	0.7244	0.6727	1.2989	1.069064
5	1.2265	1.3477	0.6771	1.046133
6	0.7907	0.6375	1.3378	1.063364
7	1.2057	1.3748	0.6504	1.090112
8	0.7923	0.6138	1.3603	1.119796
9	1.2121	1.3974	0.6292	1.150946
10	0.7825	0.5913	1.3813	1.183184
11	1.2234	1.4202	0.6079	1.216408
12	0.7704	0.5679	1.4031	1.250596
13	1.2360	1.4442	0.5855	1.285757
14	0.7574	0.5433	1.4261	1.321912
15	1.2494	1.4696	0.5619	1.359084
16	0.7435	0.5172	1.4504	1.397303
17	1.2637	1.4964	0.5369	1.436597
18	0.7289	0.4897	1.4761	1.476996
19	1.2787	1.5247	0.5105	1.518531
20	0.7135	0.4606	1.5033	1.561234
21	1.2946	1.5546	0.4826	1.605138
22	0.6971	0.4298	1.5320	1.650276
23	1.3114	1.5862	0.4531	1.696684
24	0.6798	0.3973	1.5623	1.744397
25	1.3292	1.6197	0.4219	1.793451
26	0.6616	0.3629	1.5944	1.843885
27	1.3479	1.6550	0.3889	1.895738
28	0.6423	0.3266	1.6283	1.949048
29	1.3678	1.6924	0.3540	2.003858
30	0.6219	0.2882	1.6641	2.060209

140	-6.9893	-14.0402	15.0324	43.529948
141	9.2139	16.4631	-13.4270	44.754064
142	-7.4449	-14.8980	15.8327	46.012604
143	9.6824	17.3450	-14.2498	47.306536
144	-7.9266	-15.8047	16.6786	48.636855
145	10.1776	18.2772	-15.1195	50.004584
146	-8.4357	-16.7631	17.5728	51.410776
147	10.7010	19.2626	-16.0389	52.856511
148	-8.9738	-17.7762	18.5180	54.342902
149	11.2543	20.3042	-17.0107	55.871092
150	-9.5427	-18.8471	19.5171	57.442257

```

use SOR Method to solve linear equations:
test3.3
k      x[1]      x[2]      x[3]      error
0      0.000000    0.000000    0.00000000    0.00000000
1      3.650000    0.884588    -0.202110    3.76109621
2      2.321669    0.423094    -0.222432    1.40636161
3      2.566140    0.694826    -0.495259    0.45611295
4      2.253720    0.750477    -0.662763    0.35883236
5      2.193503    0.858390    -0.799845    0.18456111
6      2.099840    0.910391    -0.881921    0.13495705
7      2.061054    0.950298    -0.934071    0.07626600
8      2.031889    0.971719    -0.963757    0.04680537
9      2.017601    0.984906    -0.980765    0.02583250
10     2.008968    0.991964    -0.989993    0.01447400
11     2.004619    0.995896    -0.994948    0.00767595
12     2.002255    0.997936    -0.997513    0.00404130
13     2.001087    0.999000    -0.998817    0.00204818
14     2.000498    0.999529    -0.999457    0.00101867
15     2.000219    0.999788    -0.999764    0.00048849
16     2.000089    0.999910    -0.999904    0.00022688
17     2.000033    0.999965    -0.999966    0.00010026
18     2.000009    0.999988    -0.999991    0.00004160
19     2.000000    0.999998    -1.000000    0.00001549
20     1.999998    1.000001    -1.000003    0.00000465

```

使用 SOR 迭代法，方程 1 和方程 3 均可解出，方程 2 不能收敛。

```

work1@ws1:~/ChenLeyu/class3$ ./test4
use Least Squares Method to solve linear equations:
test4.1
the solution is:
x1= 1.000000
x2= 4.500000
x3=-2.500000

use Least Squares Method to solve linear equations:
test4.2
the solution is:
a= 7.209302
b=-73.720930

```

使用最小二乘法解决超定线性方程组和线性最小二乘问题。

实验分析与总结：

经过这一次的实验，我更加熟悉了个人电脑上 C++ 代码的编写与调试，以及服务器上的代码编译与运行。进一步熟悉了线性方程组的各种迭代法：Jacobi 迭代法、Gauss-Seidel 迭代法、SOR 迭代法，以及超定线性方程组不可容问题与最小二乘问题的求解算法；也了解了以上方法的适用性与稳定性。对于用于矩阵运算的 Armadillo 库的使用也更加熟练了。

注：若填写内容较多，可在背面继续填写。