实验报告 5

姓名：徐煜森　　学号：　PB16110173

1. 算法分析

本次实验使用使用 Jacobi 迭代法和 Gauss-Seidel 迭代法计算线性方程组的迭代解。使用自己写的 norm 函数计算两个列向量的无穷范数。若两次迭代的解向量 x 之差的无穷范数小于等于 10^{-5}，则认为迭代已收敛。

两种迭代法均使用迭代的分量形式，而没有对矩阵求逆，因为对于容易求逆的矩阵，可以直接求逆来求解方程组，而迭代法一般用于求解难以计算矩阵的逆的情况。

其中 Jacobi 迭代法源代码：

```
void Jacobi(vector<vector<int>> A, vector<int> b, int row, int col) {
    printf("Jacobi\nstep\t");
    for (int k = 0; k < col; k++) {
        printf("x%d\t", k + 1);
    }
    printf("\n");
    vector<double> x(col, 0.0);
    vector<double> last(col, 0.0);
    int step = 0;
    printf("%d\t", step++);
    for (auto xi : x) {
        printf("%lf\t", xi);
    }
    printf("\n");
    do {
        last.assign(x.begin(), x.end());
        for (int i = 0; i < col; i++) {
            x[i] = (double)b[i];
            for (int j = 0; j < col; j++) {
                if (i == j) continue;
```

```
                x[i] -= A[i][j] * last[j];
            }
            x[i] = x[i] / (double)A[i][i];
        }
        printf("%d\t", step++);
        for (auto xi : x) {
            printf("%lf\t", xi);
        }
        printf("\n");
    } while (norm(x, last) > 0.00001);

    return;
}
```

其中 Gauss-Seidel 迭代法源代码：

```
void GaussSeidel(vector<vector<int>> A, vector<int> b, int row, int col) {
    printf("GaussSeidel\nstep\t");
    for (int k = 0; k < col; k++) {
        printf("x%d\t", k + 1);
    }
    printf("\n");
    vector<double> x(col, 0.0);
    vector<double> last(col, 0.0);
    int step = 0;
    printf("%d\t", step++);
    for (auto xi : x) {
        printf("%lf\t", xi);
    }
    printf("\n");
    do {
        last.assign(x.begin(), x.end());
        for (int i = 0; i < col; i++) {
            x[i] = (double)b[i];
            for (int j = 0; j < col; j++) {
                if (i == j) continue;
                x[i] -= A[i][j] * x[j];
            }
            x[i] = x[i] / (double)A[i][i];
        }
        printf("%d\t", step++);
        for (auto xi : x) {
            printf("%lf\t", xi);
```

```
        }
        printf("\n");
    } while (norm(x, last) > 0.00001);

    return;
}
```

2. 计算结果

Jacobi 迭代法

表 1 Jacobi 迭代法

Jacobi step	x1	x2	x3	x4	x5	x6	x7	x8	x9
0	0	0	0	0	0	0	0	0	0
1	-0.483871	0.771429	-0.74194	0	-0.350877	0.255319	-0.17073	0.259259	0.344828
2	-0.078008	0.290646	-0.51797	-0.187876	-0.29678	0.094083	0.016087	0.219825	0.362708
3	-0.331638	0.515988	-0.71816	-0.136946	-0.418922	0.221386	-0.10189	0.231167	0.301681
4	-0.196074	0.331918	-0.63631	-0.215622	-0.375489	0.12789	-0.00874	0.204028	0.31827
5	-0.303425	0.416968	-0.71513	-0.186877	-0.430759	0.193815	-0.07715	0.2133	0.291981
6	-0.246492	0.339456	-0.68116	-0.220838	-0.406721	0.141916	-0.02892	0.201117	0.301541
7	-0.295739	0.376891	-0.71462	-0.206322	-0.432038	0.176287	-0.06689	0.206277	0.290162
8	-0.268953	0.34204	-0.69907	-0.221467	-0.419724	0.148277	-0.04174	0.200746	0.295023
9	-0.292604	0.359858	-0.71408	-0.214269	-0.43162	0.166164	-0.06224	0.203386	0.289941
10	-0.279362	0.343476	-0.70658	-0.221264	-0.425403	0.15131	-0.04915	0.200807	0.292357
11	-0.291023	0.352276	-0.71359	-0.217679	-0.431136	0.16059	-0.06002	0.202137	0.290008
12	-0.284339	0.344339	-0.70988	-0.221012	-0.427992	0.152799	-0.05323	0.200901	0.291212
13	-0.290181	0.348764	-0.71326	-0.219211	-0.430811	0.157601	-0.05893	0.201573	0.290093
14	-0.286776	0.34484	-0.7114	-0.220837	-0.429215	0.153542	-0.05541	0.200968	0.290698
15	-0.289731	0.347086	-0.71306	-0.219926	-0.430622	0.156023	-0.05838	0.201308	0.290152
16	-0.287989	0.345118	-0.71211	-0.220733	-0.429808	0.153918	-0.05657	0.201007	0.290458
17	-0.289493	0.346264	-0.71294	-0.220269	-0.430518	0.155198	-0.05811	0.201181	0.290187
18	-0.2886	0.345268	-0.71246	-0.220675	-0.430101	0.154109	-0.05717	0.201029	0.290343
19	-0.289369	0.345855	-0.71288	-0.220438	-0.430462	0.154768	-0.05797	0.201118	0.290206
20	-0.28891	0.345348	-0.71264	-0.220643	-0.430248	0.154206	-0.05749	0.201041	0.290286
21	-0.289304	0.345649	-0.71285	-0.220522	-0.430432	0.154546	-0.0579	0.201086	0.290217
22	-0.289068	0.34539	-0.71272	-0.220627	-0.430323	0.154256	-0.05765	0.201047	0.290258
23	-0.28927	0.345545	-0.71283	-0.220564	-0.430417	0.154431	-0.05786	0.20107	0.290223
24	-0.289149	0.345412	-0.71277	-0.220618	-0.430361	0.154282	-0.05773	0.20105	0.290244
25	-0.289252	0.345492	-0.71282	-0.220586	-0.430409	0.154372	-0.05784	0.201062	0.290225
26	-0.28919	0.345424	-0.71279	-0.220613	-0.43038	0.154295	-0.05778	0.201052	0.290236
27	-0.289243	0.345464	-0.71282	-0.220597	-0.430405	0.154341	-0.05783	0.201058	0.290227
28	-0.289211	0.345429	-0.7128	-0.220611	-0.43039	0.154302	-0.0578	0.201053	0.290233
29	-0.289239	0.34545	-0.71281	-0.220603	-0.430403	0.154325	-0.05783	0.201056	0.290228
30	-0.289222	0.345432	-0.71281	-0.22061	-0.430395	0.154305	-0.05781	0.201053	0.290231
31	-0.289236	0.345443	-0.71281	-0.220605	-0.430402	0.154317	-0.05783	0.201055	0.290228
32	-0.289228	0.345434	-0.71281	-0.220609	-0.430398	0.154307	-0.05782	0.201054	0.29023
33	-0.289235	0.34544	-0.71281	-0.220607	-0.430401	0.154313	-0.05782	0.201054	0.290228

从图中可以看出本次实验中 Jacobi 迭代法经过 33 步迭代后收敛，收敛结果经过验算后证实其在当前精度条件下正确。

Gauss-Seidel 迭代法

表 2 Gauss-Seidel 迭代法

GaussSeidel									
step	x1	x2	x3	x4	x5	x6	x7	x8	x9
0	0	0	0	0	0	0	0	0	0
1	-0.483871	0.591705	-0.57015	-0.072171	-0.388862	0.197404	-0.02629	0.187248	0.335343
2	-0.172058	0.438698	-0.63785	-0.190206	-0.410318	0.177427	-0.04091	0.208115	0.300151
3	-0.242666	0.388319	-0.69056	-0.209035	-0.42085	0.166528	-0.04888	0.203557	0.293993
4	-0.267309	0.362304	-0.70418	-0.215461	-0.425971	0.160676	-0.05316	0.202153	0.291902
5	-0.280106	0.352438	-0.70912	-0.218268	-0.42829	0.157597	-0.05542	0.201569	0.29099
6	-0.285237	0.348533	-0.71116	-0.219511	-0.429374	0.155997	-0.05659	0.2013	0.290586
7	-0.28739	0.346868	-0.71204	-0.22008	-0.429894	0.155173	-0.05719	0.201174	0.290401
8	-0.288354	0.34612	-0.71244	-0.22035	-0.430148	0.15475	-0.0575	0.201113	0.290313
9	-0.288805	0.345769	-0.71263	-0.22048	-0.430273	0.154534	-0.05766	0.201084	0.290271
10	-0.289021	0.345601	-0.71272	-0.220544	-0.430336	0.154423	-0.05774	0.201069	0.29025
11	-0.289128	0.345518	-0.71277	-0.220576	-0.430368	0.154367	-0.05778	0.201061	0.290239
12	-0.28918	0.345477	-0.71279	-0.220592	-0.430384	0.154338	-0.0578	0.201058	0.290234
13	-0.289207	0.345457	-0.7128	-0.2206	-0.430392	0.154324	-0.05781	0.201056	0.290231
14	-0.28922	0.345446	-0.71281	-0.220604	-0.430396	0.154316	-0.05782	0.201055	0.29023
15	-0.289227	0.345441	-0.71281	-0.220606	-0.430398	0.154313	-0.05782	0.201054	0.290229

从图中可以看出 Gauss-Seidel 迭代经过 15 步迭代后收敛，迭代步约为 Jacobi 迭代的 1/2，两者收敛结果有细微的差异 （小于等于 10^{-5}），验算后证实其在当前精度条件下为正确结果。

3. 结果分析与对比

对比表一和表二可以发现，Jacobi 迭代法经过 33 步迭代后收敛，而 Gauss-Seidel 迭代法经过 15 步就收敛了。这一实验结果符合预期，因一般情况下 Gauss-Seidel 迭代法收敛速度比 Jacobi 迭代法要快。理论上，收敛速度最终是由迭代矩阵的谱半径决定的。

4. 实验小结

通过本次实验掌握了解线性方程组的两种迭代法，在部分情况下

通过求解矩阵的逆或直接法来求解线性方程组十分困难，这种情况下可以使用迭代法求解方程组。理论上，Jacobi 迭代法是否收敛和 Gauss-Seidel 迭代法是否收敛没有联系。另外一般情况下，Gauss-Seidel 迭代法收敛速度比 Jacobi 迭代法要快。