2019-09-17 15:59:11 我是一只计算鸡 阅读数 8 更多

版权声明:本文为博主原创文章,遵循 CC 4.0 BY-SA 版权协议,转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/giftedpanda/article/details/100929252

我们小学学过二元一次方程组,

$$\begin{cases} x + y = 3 \\ x + 2y = 4 \end{cases}$$

很明显,我们用下面的方程减去上面的方程就能求出y,然后再回代第一个方程我们就能求出x,高斯消元也是基于这样的思想。

```
\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2 \\ & \dots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n = b_n \end{cases}
```

我们可以将这个写成增广矩阵的形式

```
\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{bmatrix}
```

为了求出解,我们只有化解为上三角行列式,怎么化简呢可以这样,可以这样考虑,对于第 i 行来说,我们用第 i 行下面的每一行减去倍数关系使得第行的第 i 列都为0。

先不管系数, 我们来看一下过程。

```
b_1
                                             |a_{11}|
                                                                               b_1
                                                                                           |a_{11} \quad a_{12} \quad \dots \quad a_{1n}
a_{11} a_{12} ... a_{1n}
                                                      a_{12} ... a_{1n}
                                                0
                                                                                             0
a_{21}
         a_{22} ... a_{2n}
                                  b_2
                                                       a_{22} ... a_{2n}
                                                                                b_2
                                                                                                    a_{22} ... a_{2n}
                                                                                     \Rightarrow
                                        \Rightarrow
\begin{vmatrix} a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{vmatrix}
                                             0
                                                       a_{n2} \dots a_{nn} b_n
                                                                                           0
```

我们取枚举每一行,然后用每一行下面的行,减去这一行的倍数,就能化简为上三角行列式。

接下来怎么求出方程的解呢,回代。

最后一行只有一个未知数的系数不为0,一次除法就能得到方程的解了,倒数第二行只有两个未知数系数不为0,将最后一行的解带入倒数第二行,就见知数,一次除法就能得到倒数第二行的解了,依次类推。

为了提高精度,我们按第i行系数绝对值从大到小进行枚举。

```
1 | #include<bits/stdc++.h>
2 using namespace std;
    const int maxn = 100 + 7;
    const double eps = 1e-8;
    typedef double matrix[maxn][maxn];
6
    int guass_elimination(matrix A, int n) // 高斯消元
7
8
            int i, j, k, r;
            for(i = 0; i < n; i++) { // 消元
9
10
                    r = i;
                    for(j = i + 1; j < n; j++) if(fabs(A[j][i]) > fabs(A[r][i])) r = j;
11
                    if(fabs(A[r][i]) < eps) return 0; // 无解
12
                    if(r != i) for(int j = 0; j <= n; j++) swap(A[i][j], A[r][j]);
13
                    for(k = i + 1; k < n; k++) {
14
15
                            double f = A[k][i] / A[i][i];
16
                            for(j = i; j \le n; j++) A[k][j] -= f * A[i][j];
17
18
19
            for(i = n - 1; i >= 0; i--) { // 回代
20
                    for(j = i + 1; j < n; j++) {
21
                            A[i][n] -= A[j][n] * A[i][j];
22
```

102 程序员节,为程序员加油! 闭

```
12 程序员节,为程序员加油! 矧
```

```
A[i][n] /= A[i][i];
24 |
23
25
            return 1;
26 }
27 int main()
28
    {
29
            int n;
            matrix A;
30
            while(scanf("%d", &n) == 1) {
31
32
                    for(int i = 0; i < n; i++) {
33
                            for(int j = 0; j \leftarrow n; j++) {
                                     scanf("%lf", &A[i][j]);
34
35
36
                    }
                    int flag = guass_elimination(A, n);
37
                    if(flag == 0) printf("No Solution\n");
38
                    else {
39
                            for(int i = 0; i < n; i++) printf("%.21f\n", A[i][n]);</pre>
40
41
42
43
            return 0;
44 }
```

有 0 个人打赏 文章最后发布于: 201

©2019 CSDN 皮肤主题: 终极编程指南 设计师: CSDN官方博客