原创 【luogu 3389】高斯消元

2019-09-17 15:59:11 我是一只计算鸡 阅读数 8 更多

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。

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

我们可以将这个写成增广矩阵的形式

$$\begin{vmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ a_{21} & a_{22} & \ldots & a_{2n} & b_2 \\ \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} & b_n \end{vmatrix}$$

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。

$$\begin{vmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ a_{21} & a_{22} & \ldots & a_{2n} & b_2 \\ \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} & b_n \end{vmatrix} \Rightarrow \begin{vmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ 0 & a_{22} & \ldots & a_{2n} & b_2 \\ \ldots \\ 0 & a_{n2} & \ldots & a_{nn} & b_n \end{vmatrix} \Rightarrow \begin{vmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ 0 & a_{22} & \ldots & a_{2n} & b_2 \\ \ldots \\ 0 & 0 & \ldots & a_{nn} & b_n \end{vmatrix}$$

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。

```
#include<bits/stdc++.h>
using namespace std;
const int maxn = 100 + 7;
const double eps = 1e-8;
typedef double matrix[maxn][maxn];
int guass_elimination(matrix A, int n) // 高斯消元
{
	int i, j, k, r;
	for(i = 0; i < n; i++) { // 消元
		r = i;
		for(j = i + 1; j < n; j++) if(fabs(A[j][i]) > fabs(A[r][i])) r = j;
		if(fabs(A[r][i]) < eps) return 0;  // 无解
		if(r != i) for(int j = 0; j <= n; j++) swap(A[i][j], A[r][j]);
		for(k = i + 1; k < n; k++) {
			double f = A[k][i] / A[i][i];
			for(j = i; j <= n; j++) A[k][j] -= f * A[i][j];
		}
	}
	for(i = n - 1; i >= 0; i--) {  // 回代
		for(j = i + 1; j < n; j++) {
			A[i][n] -= A[j][n] * A[i][j];
		}
```

```
			A[i][n] /= A[i][i];
		}
	return 1;
}
int main()
{
	int n;
	matrix A;
	while(scanf("%d", &n) == 1) {
		for(int i = 0; i < n; i++) {
			for(int j = 0; j <= n; j++) {
				scanf("%lf", &A[i][j]);
			}
		}
		int flag = guass_elimination(A, n);
		if(flag == 0) printf("No Solution\n");
		else {
			for(int i = 0; i < n; i++) printf("%.2lf\n", A[i][n]);
		}
	}
	return 0;
}
```