

原

高斯消元算法

2018年08月30日 17:47:37

数学家是我理想

阅读数 614

一、基本描述

高斯消元主要用来求解线性方程组，也可以求解矩阵的秩、矩阵的逆。它的时间复杂度是 n^3 ，主要与方程组的个数， n 的个数有关

二、算法过程

其实高斯消元的过程就是手算解方程组的过程，分两步：1.加减消元，2.回代求未知数值。下面通过一个例子来具体说明

三、示例

有样一个三元一次方程组：

$$\begin{cases} 2x + y + z = 1 & \text{①} \\ 6x + 2y + z = -1 & \text{②} \\ -2x + 2y + z = 7 & \text{③} \end{cases}$$

1.消去x

① \times (-3)+②得到 $0x-y-2z=-4$

①+③得到 $0x+3y+2z=8$

从而得到

$$\begin{cases} 2x + y + z = 1 & \text{①} \\ 0x - y - 2z = -4 & \text{②} \\ 0x + 3y + 2z = 8 & \text{③} \end{cases}$$

2.消去y

② \times 3+③得到 $0x+0y-4z=-4$

进而得到

$$\begin{cases} 2x + y + z = 1 & \text{①} \\ 0x - y - 2z = -4 & \text{②} \\ 0x + 0y - 4z = -4 & \text{③} \end{cases}$$

至此，我们已经求出了 $z=1$ ，下一步进行回代

3.回代求解x

将 $z=1,y=2$ 带入①，求得 $x=-1$

最终得到

$$\begin{cases} x = -1 \\ y = 2 \\ z = 1 \end{cases}$$

四、再说算法


对于方程组，其系数是具体存在矩阵(数组)里的，下面在给出实际在矩阵中的表示

$$\begin{bmatrix} x & y & z & val \\ 2 & 1 & 1 & 1 \\ 6 & 2 & 1 & -1 \\ -2 & 2 & 1 & 7 \end{bmatrix}$$


1.消去x


$$\begin{bmatrix} x & y & z & val \\ 2 & 1 & 1 & 1 \\ 0 & -1 & -2 & -4 \\ 0 & 3 & 2 & 8 \end{bmatrix}$$

1











2.消去y

$$\begin{bmatrix} x & y & z & val \\ 2 & 1 & 1 & 1 \\ 0 & -1 & -2 & -4 \\ 0 & 0 & -4 & -4 \end{bmatrix}$$

3.回代求解y

回代的时候，记录各个变量的结果将保存在另外一个数组当中，故保存矩阵的数组值不会发生改变，该矩阵主要进行消元过程。

五、有几个问题

- Q1：系数不一定是整数啊？
A1：这时候数组就要用到浮点数了！不能是整数！
- Q2：什么时候无解啊？
A2：消元完了，发现有一行系数都为0，但是常数项不为0，当然无解啦！比如：

$$\begin{bmatrix} x & y & z & val \\ 2 & 1 & 1 & 1 \\ 0 & -1 & -2 & -4 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$

- Q3：什么时候多解啊？
A3：消元完了，发现有好几行系数为0，常数项也为0，这样就多解了！有几行为全为0，就有几个自由元，所谓自由元，就是这些变量的值可以随某种情况可以满足给出的方程组，比如：

$$\begin{bmatrix} x & y & z & val \\ 2 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- Q4：那什么时候解是唯一的啊！
A4：您做一下排除法，不满足2和3的，不就是解释唯一的嘛！其实也就是说我们的系数矩阵可以化成上三角阵。

六、代码实现

c/c++高斯消元模板

```
1  #include<stdio.h>
2  #include<algorithm>
3  #include<iostream>
4  #include<string.h>
5  #include<math.h>
6  using namespace std;
7  const int MAXN=50;
8  int a[MAXN][MAXN]; //增广矩阵
9  int x[MAXN]; //解集
10 bool free_x[MAXN]; //标记是否是不确定的变元
11 int gcd(int a,int b) {
12     if(b == 0)
13         return a;
14     else
15         return gcd(b,a % b);
16 }
17
18 inline int lcm(int a,int b) {
19     return a / gcd(a,b) * b; //先除后乘防溢出
20 }
21 //高斯消元法解方程组(Gauss-Jordan elimination).(-2表示有浮点数解，但无整数解，
22 //-1表示无解，0表示唯一解，大于0表示无穷解，并返回自由变元的个数)
23 //有equ个方程，var个变元。增广矩阵行数为equ,分别为0到equ-1,列数为var+1,分别为0到var.
24 int Gauss(int equ,int var) {
25     int i,j,k;
26     int max_r,max_c;
27     int x_lcm=lcm(1,2);
28     for(i=0;i<equ;i++)
29         for(j=0;j<var;j++)
30             a[i][j]=a[i][j]*x_lcm;
31     for(i=0;i<equ;i++)
32         for(j=0;j<var;j++)
33             a[i][j]=a[i][j]/x_lcm;
34     for(i=0;i<equ;i++)
35         for(j=0;j<var;j++)
36             a[i][j]=a[i][j];
37     for(i=0;i<equ;i++)
38         for(j=0;j<var;j++)
39             a[i][j]=a[i][j];
40     for(i=0;i<equ;i++)
41         for(j=0;j<var;j++)
42             a[i][j]=a[i][j];
43     for(i=0;i<equ;i++)
44         for(j=0;j<var;j++)
45             a[i][j]=a[i][j];
46     for(i=0;i<equ;i++)
47         for(j=0;j<var;j++)
48             a[i][j]=a[i][j];
49     for(i=0;i<equ;i++)
50         for(j=0;j<var;j++)
51             a[i][j]=a[i][j];
52     for(i=0;i<equ;i++)
53         for(j=0;j<var;j++)
54             a[i][j]=a[i][j];
55     for(i=0;i<equ;i++)
56         for(j=0;j<var;j++)
57             a[i][j]=a[i][j];
58     for(i=0;i<equ;i++)
59         for(j=0;j<var;j++)
60             a[i][j]=a[i][j];
61     for(i=0;i<equ;i++)
62         for(j=0;j<var;j++)
63             a[i][j]=a[i][j];
64     for(i=0;i<equ;i++)
65         for(j=0;j<var;j++)
66             a[i][j]=a[i][j];
67     for(i=0;i<equ;i++)
68         for(j=0;j<var;j++)
69             a[i][j]=a[i][j];
70     for(i=0;i<equ;i++)
71         for(j=0;j<var;j++)
72             a[i][j]=a[i][j];
73     for(i=0;i<equ;i++)
74         for(j=0;j<var;j++)
75             a[i][j]=a[i][j];
76     for(i=0;i<equ;i++)
77         for(j=0;j<var;j++)
78             a[i][j]=a[i][j];
79     for(i=0;i<equ;i++)
80         for(j=0;j<var;j++)
81             a[i][j]=a[i][j];
82     for(i=0;i<equ;i++)
83         for(j=0;j<var;j++)
84             a[i][j]=a[i][j];
85     for(i=0;i<equ;i++)
86         for(j=0;j<var;j++)
87             a[i][j]=a[i][j];
88     for(i=0;i<equ;i++)
89         for(j=0;j<var;j++)
90             a[i][j]=a[i][j];
91     for(i=0;i<equ;i++)
92         for(j=0;j<var;j++)
93             a[i][j]=a[i][j];
94     for(i=0;i<equ;i++)
95         for(j=0;j<var;j++)
96             a[i][j]=a[i][j];
97     for(i=0;i<equ;i++)
98         for(j=0;j<var;j++)
99             a[i][j]=a[i][j];
100    for(i=0;i<equ;i++)
101        for(j=0;j<var;j++)
102            a[i][j]=a[i][j];
103    for(i=0;i<equ;i++)
104        for(j=0;j<var;j++)
105            a[i][j]=a[i][j];
106    for(i=0;i<equ;i++)
107        for(j=0;j<var;j++)
108            a[i][j]=a[i][j];
109    for(i=0;i<equ;i++)
110        for(j=0;j<var;j++)
111            a[i][j]=a[i][j];
112    for(i=0;i<equ;i++)
113        for(j=0;j<var;j++)
114            a[i][j]=a[i][j];
115    for(i=0;i<equ;i++)
116        for(j=0;j<var;j++)
117            a[i][j]=a[i][j];
118    for(i=0;i<equ;i++)
119        for(j=0;j<var;j++)
120            a[i][j]=a[i][j];
121    for(i=0;i<equ;i++)
122        for(j=0;j<var;j++)
123            a[i][j]=a[i][j];
124    for(i=0;i<equ;i++)
125        for(j=0;j<var;j++)
126            a[i][j]=a[i][j];
127    for(i=0;i<equ;i++)
128        for(j=0;j<var;j++)
129            a[i][j]=a[i][j];
130    for(i=0;i<equ;i++)
131        for(j=0;j<var;j++)
132            a[i][j]=a[i][j];
133    for(i=0;i<equ;i++)
134        for(j=0;j<var;j++)
135            a[i][j]=a[i][j];
136    for(i=0;i<equ;i++)
137        for(j=0;j<var;j++)
138            a[i][j]=a[i][j];
139    for(i=0;i<equ;i++)
140        for(j=0;j<var;j++)
141            a[i][j]=a[i][j];
142    for(i=0;i<equ;i++)
143        for(j=0;j<var;j++)
144            a[i][j]=a[i][j];
145    for(i=0;i<equ;i++)
146        for(j=0;j<var;j++)
147            a[i][j]=a[i][j];
148    for(i=0;i<equ;i++)
149        for(j=0;j<var;j++)
150            a[i][j]=a[i][j];
151    for(i=0;i<equ;i++)
152        for(j=0;j<var;j++)
153            a[i][j]=a[i][j];
154    for(i=0;i<equ;i++)
155        for(j=0;j<var;j++)
156            a[i][j]=a[i][j];
157    for(i=0;i<equ;i++)
158        for(j=0;j<var;j++)
159            a[i][j]=a[i][j];
160    for(i=0;i<equ;i++)
161        for(j=0;j<var;j++)
162            a[i][j]=a[i][j];
163    for(i=0;i<equ;i++)
164        for(j=0;j<var;j++)
165            a[i][j]=a[i][j];
166    for(i=0;i<equ;i++)
167        for(j=0;j<var;j++)
168            a[i][j]=a[i][j];
169    for(i=0;i<equ;i++)
170        for(j=0;j<var;j++)
171            a[i][j]=a[i][j];
172    for(i=0;i<equ;i++)
173        for(j=0;j<var;j++)
174            a[i][j]=a[i][j];
175    for(i=0;i<equ;i++)
176        for(j=0;j<var;j++)
177            a[i][j]=a[i][j];
178    for(i=0;i<equ;i++)
179        for(j=0;j<var;j++)
180            a[i][j]=a[i][j];
181    for(i=0;i<equ;i++)
182        for(j=0;j<var;j++)
183            a[i][j]=a[i][j];
184    for(i=0;i<equ;i++)
185        for(j=0;j<var;j++)
186            a[i][j]=a[i][j];
187    for(i=0;i<equ;i++)
188        for(j=0;j<var;j++)
189            a[i][j]=a[i][j];
190    for(i=0;i<equ;i++)
191        for(j=0;j<var;j++)
192            a[i][j]=a[i][j];
193    for(i=0;i<equ;i++)
194        for(j=0;j<var;j++)
195            a[i][j]=a[i][j];
196    for(i=0;i<equ;i++)
197        for(j=0;j<var;j++)
198            a[i][j]=a[i][j];
199    for(i=0;i<equ;i++)
200        for(j=0;j<var;j++)
201            a[i][j]=a[i][j];
202    for(i=0;i<equ;i++)
203        for(j=0;j<var;j++)
204            a[i][j]=a[i][j];
205    for(i=0;i<equ;i++)
206        for(j=0;j<var;j++)
207            a[i][j]=a[i][j];
208    for(i=0;i<equ;i++)
209        for(j=0;j<var;j++)
210            a[i][j]=a[i][j];
211    for(i=0;i<equ;i++)
212        for(j=0;j<var;j++)
213            a[i][j]=a[i][j];
214    for(i=0;i<equ;i++)
215        for(j=0;j<var;j++)
216            a[i][j]=a[i][j];
217    for(i=0;i<equ;i++)
218        for(j=0;j<var;j++)
219            a[i][j]=a[i][j];
220    for(i=0;i<equ;i++)
221        for(j=0;j<var;j++)
222            a[i][j]=a[i][j];
223    for(i=0;i<equ;i++)
224        for(j=0;j<var;j++)
225            a[i][j]=a[i][j];
226    for(i=0;i<equ;i++)
227        for(j=0;j<var;j++)
228            a[i][j]=a[i][j];
229    for(i=0;i<equ;i++)
230        for(j=0;j<var;j++)
231            a[i][j]=a[i][j];
232    for(i=0;i<equ;i++)
233        for(j=0;j<var;j++)
234            a[i][j]=a[i][j];
235    for(i=0;i<equ;i++)
236        for(j=0;j<var;j++)
237            a[i][j]=a[i][j];
238    for(i=0;i<equ;i++)
239        for(j=0;j<var;j++)
240            a[i][j]=a[i][j];
241    for(i=0;i<equ;i++)
242        for(j=0;j<var;j++)
243            a[i][j]=a[i][j];
244    for(i=0;i<equ;i++)
245        for(j=0;j<var;j++)
246            a[i][j]=a[i][j];
247    for(i=0;i<equ;i++)
248        for(j=0;j<var;j++)
249            a[i][j]=a[i][j];
250    for(i=0;i<equ;i++)
251        for(j=0;j<var;j++)
252            a[i][j]=a[i][j];
253    for(i=0;i<equ;i++)
254        for(j=0;j<var;j++)
255            a[i][j]=a[i][j];
256    for(i=0;i<equ;i++)
257        for(j=0;j<var;j++)
258            a[i][j]=a[i][j];
259    for(i=0;i<equ;i++)
260        for(j=0;j<var;j++)
261            a[i][j]=a[i][j];
262    for(i=0;i<equ;i++)
263        for(j=0;j<var;j++)
264            a[i][j]=a[i][j];
265    for(i=0;i<equ;i++)
266        for(j=0;j<var;j++)
267            a[i][j]=a[i][j];
268    for(i=0;i<equ;i++)
269        for(j=0;j<var;j++)
270            a[i][j]=a[i][j];
271    for(i=0;i<equ;i++)
272        for(j=0;j<var;j++)
273            a[i][j]=a[i][j];
274    for(i=0;i<equ;i++)
275        for(j=0;j<var;j++)
276            a[i][j]=a[i][j];
277    for(i=0;i<equ;i++)
278        for(j=0;j<var;j++)
279            a[i][j]=a[i][j];
280    for(i=0;i<equ;i++)
281        for(j=0;j<var;j++)
282            a[i][j]=a[i][j];
283    for(i=0;i<equ;i++)
284        for(j=0;j<var;j++)
285            a[i][j]=a[i][j];
286    for(i=0;i<equ;i++)
287        for(j=0;j<var;j++)
288            a[i][j]=a[i][j];
289    for(i=0;i<equ;i++)
290        for(j=0;j<var;j++)
291            a[i][j]=a[i][j];
292    for(i=0;i<equ;i++)
293        for(j=0;j<var;j++)
294            a[i][j]=a[i][j];
295    for(i=0;i<equ;i++)
296        for(j=0;j<var;j++)
297            a[i][j]=a[i][j];
298    for(i=0;i<equ;i++)
299        for(j=0;j<var;j++)
300            a[i][j]=a[i][j];
301    for(i=0;i<equ;i++)
302        for(j=0;j<var;j++)
303            a[i][j]=a[i][j];
304    for(i=0;i<equ;i++)
305        for(j=0;j<var;j++)
306            a[i][j]=a[i][j];
307    for(i=0;i<equ;i++)
308        for(j=0;j<var;j++)
309            a[i][j]=a[i][j];
310    for(i=0;i<equ;i++)
311        for(j=0;j<var;j++)
312            a[i][j]=a[i][j];
313    for(i=0;i<equ;i++)
314        for(j=0;j<var;j++)
315            a[i][j]=a[i][j];
316    for(i=0;i<equ;i++)
317        for(j=0;j<var;j++)
318            a[i][j]=a[i][j];
319    for(i=0;i<equ;i++)
320        for(j=0;j<var;j++)
321            a[i][j]=a[i][j];
322    for(i=0;i<equ;i++)
323        for(j=0;j<var;j++)
324            a[i][j]=a[i][j];
325    for(i=0;i<equ;i++)
326        for(j=0;j<var;j++)
327            a[i][j]=a[i][j];
328    for(i=0;i<equ;i++)
329        for(j=0;j<var;j++)
330            a[i][j]=a[i][j];
331    for(i=0;i<equ;i++)
332        for(j=0;j<var;j++)
333            a[i][j]=a[i][j];
334    for(i=0;i<equ;i++)
335        for(j=0;j<var;j++)
336            a[i][j]=a[i][j];
337    for(i=0;i<equ;i++)
338        for(j=0;j<var;j++)
339            a[i][j]=a[i][j];
340    for(i=0;i<equ;i++)
341        for(j=0;j<var;j++)
342            a[i][j]=a[i][j];
343    for(i=0;i<equ;i++)
344        for(j=0;j<var;j++)
345            a[i][j]=a[i][j];
346    for(i=0;i<equ;i++)
347        for(j=0;j<var;j++)
348            a[i][j]=a[i][j];
349    for(i=0;i<equ;i++)
350        for(j=0;j<var;j++)
351            a[i][j]=a[i][j];
352    for(i=0;i<equ;i++)
353        for(j=0;j<var;j++)
354            a[i][j]=a[i][j];
355    for(i=0;i<equ;i++)
356        for(j=0;j<var;j++)
357            a[i][j]=a[i][j];
358    for(i=0;i<equ;i++)
359        for(j=0;j<var;j++)
360            a[i][j]=a[i][j];
361    for(i=0;i<equ;i++)
362        for(j=0;j<var;j++)
363            a[i][j]=a[i][j];
364    for(i=0;i<equ;i++)
365        for(j=0;j<var;j++)
366            a[i][j]=a[i][j];
367    for(i=0;i<equ;i++)
368        for(j=0;j<var;j++)
369            a[i][j]=a[i][j];
370    for(i=0;i<equ;i++)
371        for(j=0;j<var;j++)
372            a[i][j]=a[i][j];
373    for(i=0;i<equ;i++)
374        for(j=0;j<var;j++)
375            a[i][j]=a[i][j];
376    for(i=0;i<equ;i++)
377        for(j=0;j<var;j++)
378            a[i][j]=a[i][j];
379    for(i=0;i<equ;i++)
380        for(j=0;j<var;j++)
381            a[i][j]=a[i][j];
382    for(i=0;i<equ;i++)
383        for(j=0;j<var;j++)
384            a[i][j]=a[i][j];
385    for(i=0;i<equ;i++)
386        for(j=0;j<var;j++)
387            a[i][j]=a[i][j];
388    for(i=0;i<equ;i++)
389        for(j=0;j<var;j++)
390            a[i][j]=a[i][j];
391    for(i=0;i<equ;i++)
392        for(j=0;j<var;j++)
393            a[i][j]=a[i][j];
394    for(i=0;i<equ;i++)
395        for(j=0;j<var;j++)
396            a[i][j]=a[i][j];
397    for(i=0;i<equ;i++)
398        for(j=0;j<var;j++)
399            a[i][j]=a[i][j];
400    for(i=0;i<equ;i++)
401        for(j=0;j<var;j++)
402            a[i][j]=a[i][j];
403    for(i=0;i<equ;i++)
404        for(j=0;j<var;j++)
405            a[i][j]=a[i][j];
406    for(i=0;i<equ;i++)
407        for(j=0;j<var;j++)
408            a[i][j]=a[i][j];
409    for(i=0;i<equ;i++)
410        for(j=0;j<var;j++)
411            a[i][j]=a[i][j];
412    for(i=0;i<equ;i++)
413        for(j=0;j<var;j++)
414            a[i][j]=a[i][j];
415    for(i=0;i<equ;i++)
416        for(j=0;j<var;j++)
417            a[i][j]=a[i][j];
418    for(i=0;i<equ;i++)
419        for(j=0;j<var;j++)
420            a[i][j]=a[i][j];
421    for(i=0;i<equ;i++)
422        for(j=0;j<var;j++)
423            a[i][j]=a[i][j];
424    for(i=0;i<equ;i++)
425        for(j=0;j<var;j++)
426            a[i][j]=a[i][j];
427    for(i=0;i<equ;i++)
428        for(j=0;j<var;j++)
429            a[i][j]=a[i][j];
430    for(i=0;i<equ;i++)
431        for(j=0;j<var;j++)
432            a[i][j]=a[i][j];
433    for(i=0;i<equ;i++)
434        for(j=0;j<var;j++)
435            a[i][j]=a[i][j];
436    for(i=0;i<equ;i++)
437        for(j=0;j<var;j++)
438            a[i][j]=a[i][j];
439    for(i=0;i<equ;i++)
440        for(j=0;j<var;j++)
441            a[i][j]=a[i][j];
442    for(i=0;i<equ;i++)
443        for(j=0;j<var;j++)
444            a[i][j]=a[i][j];
445    for(i=0;i<equ;i++)
446        for(j=0;j<var;j++)
447            a[i][j]=a[i][j];
448    for(i=0;i<equ;i++)
449        for(j=0;j<var;j++)
450            a[i][j]=a[i][j];
451    for(i=0;i<equ;i++)
452        for(j=0;j<var;j++)
453            a[i][j]=a[i][j];
454    for(i=0;i<equ;i++)
455        for(j=0;j<var;j++)
456            a[i][j]=a[i][j];
457    for(i=0;i<equ;i++)
458        for(j=0;j<var;j++)
459            a[i][j]=a[i][j];
460    for(i=0;i<equ;i++)
461        for(j=0;j<var;j++)
462            a[i][j]=a[i][j];
463    for(i=0;i<equ;i++)
464        for(j=0;j<var;j++)
465            a[i][j]=a[i][j];
466    for(i=0;i<equ;i++)
467        for(j=0;j<var;j++)
468            a[i][j]=a[i][j];
469    for(i=0;i<equ;i++)
470        for(j=0;j<var;j++)
471            a[i][j]=a[i][j];
472    for(i=0;i<equ;i++)
473        for(j=0;j<var;j++)
474            a[i][j]=a[i][j];
475    for(i=0;i<equ;i++)
476        for(j=0;j<var;j++)
477            a[i][j]=a[i][j];
478    for(i=0;i<equ;i++)
479        for(j=0;j<var;j++)
480            a[i][j]=a[i][j];
481    for(i=0;i<equ;i++)
482        for(j=0;j<var;j++)
483            a[i][j]=a[i][j];
484    for(i=0;i<equ;i++)
485        for(j=0;j<var;j++)
486            a[i][j]=a[i][j];
487    for(i=0;i<equ;i++)
488        for(j=0;j<var;j++)
489            a[i][j]=a[i][j];
490    for(i=0;i<equ;i++)
491        for(j=0;j<var;j++)
492            a[i][j]=a[i][j];
493    for(i=0;i<equ;i++)
494        for(j=0;j<var;j++)
495            a[i][j]=a[i][j];
496    for(i=0;i<equ;i++)
497        for(j=0;j<var;j++)
498            a[i][j]=a[i][j];
499    for(i=0;i<equ;i++)
500        for(j=0;j<var;j++)
501            a[i][j]=a[i][j];
502    for(i=0;i<equ;i++)
503        for(j=0;j<var;j++)
504            a[i][j]=a[i][j];
505    for(i=0;i<equ;i++)
506        for(j=0;j<var;j++)
507            a[i][j]=a[i][j];
508    for(i=0;i<equ;i++)
509        for(j=0;j<var;j++)
510            a[i][j]=a[i][j];
511    for(i=0;i<equ;i++)
512        for(j=0;j<var;j++)
513            a[i][j]=a[i][j];
514    for(i=0;i<equ;i++)
515        for(j=0;j<var;j++)
516            a[i][j]=a[i][j];
517    for(i=0;i<equ;i++)
518        for(j=0;j<var;j++)
519            a[i][j]=a[i][j];
520    for(i=0;i<equ;i++)
521        for(j=0;j<var;j++)
522            a[i][j]=a[i][j];
523    for(i=0;i<equ;i++)
524        for(j=0;j<var;j++)
525            a[i][j]=a[i][j];
526    for(i=0;i<equ;i++)
527        for(j=0;j<var;j++)
528            a[i][j]=a[i][j];
529    for(i=0;i<equ;i++)
530        for(j=0;j<var;j++)
531            a[i][j]=a[i][j];
532    for(i=0;i<equ;i++)
533        for(j=0;j<var;j++)
534            a[i][j]=a[i][j];
535    for(i=0;i<equ;i++)
536        for(j=0;j<var;j++)
537            a[i][j]=a[i][j];
538    for(i=0;i<equ;i++)
539        for(j=0;j<var;j++)
540            a[i][j]=a[i][j];
541    for(i=0;i<equ;i++)
542        for(j=0;j<var;j++)
543            a[i][j]=a[i][j];
544    for(i=0;i<equ;i++)
545        for(j=0;j<var;j++)
546            a[i][j]=a[i][j];
547    for(i=0;i<equ;i++)
548        for(j=0;j<var;j++)
549            a[i][j]=a[i][j];
550    for(i=0;i<equ;i++)
551        for(j=0;j<var;j++)
552            a[i][j]=a[i][j];
553    for(i=0;i<equ;i++)
554        for(j=0;j<var;j++)
555            a[i][j]=a[i][j];
556    for(i=0;i<equ;i++)
557        for(j=0;j<var;j++)
558            a[i][j]=a[i][j];
559    for(i=0;i<equ;i++)
560        for(j=0;j<var;j++)
561            a[i][j]=a[i][j];
562    for(i=0;i<equ;i++)
563        for(j=0;j<var;j++)
564            a[i][j]=a[i][j];
565    for(i=0;i<equ;i++)
566        for(j=0;j<var;j++)
567            a[i][j]=a[i][j];
568    for(i=0;i<equ;i++)
569        for(j=0;j<var;j++)
570            a[i][j]=a[i][j];
571    for(i=0;i<equ;i++)
572        for(j=0;j<var;j++)
573            a[i][j]=a[i][j];
574    for(i=0;i<equ;i++)
575        for(j=0;j<var;j++)
576            a[i][j]=a[i][j];
577    for(i=0;i<equ;i++)
578        for(j=0;j<var;j++)
579            a[i][j]=a[i][j];
580    for(i=0;i<equ;i++)
581        for(j=0;j<var;j++)
582            a[i][j]=a[i][j];
583    for(i=0;i<equ;i++)
584        for(j=0;j<var;j++)
585            a[i][j]=a[i][j];
586    for(i=0;i<equ;i++)
587        for(j=0;j<var;j++)
588            a[i][j]=a[i][j];
589    for(i=0;i<equ;i++)
590        for(j=0;j<var;j++)
591            a[i][j]=a[i][j];
592    for(i=0;i<equ;i++)
593        for(j=0;j<var;j++)
594            a[i][j]=a[i][j];
595    for(i=0;i<equ;i++)
596        for(j=0;j<var;j++)
597            a[i][j]=a[i][j];
598    for(i=0;i<equ;i++)
599        for(j=0;j<var;j++)
600            a[i][j]=a[i][j];
601    for(i=0;i<equ;i++)
602        for(j=0;j<var;j++)
603            a[i][j]=a[i][j];
604    for(i=0;i<equ;i++)
605        for(j=0;j<var;j++)
606            a[i][j]=a[i][j];
607    for(i=0;i<equ;i++)
608        for(j=0;j<var;j++)
609            a[i][j]=a[i][j];
610    for(i=0;i<equ;i++)
611        for(j=0;j<var;j++)
612            a[i][j]=a[i][j];
613    for(i=0;i<equ;i++)
614        for(j=0;j<var;j++)
615            a[i][j]=a[i][j];
616    for(i=0;i<equ;i++)
617        for(j=0;j<var;j++)
618            a[i][j]=a[i][j];
619    for(i=0;i<equ;i++)
620        for(j=0;j<var;j++)
621            a[i][j]=a[i][j];
622    for(i=0;i<equ;i++)
623        for(j=0;j<var;j++)
624            a[i][j]=a[i][j];
625    for(i=0;i<equ;i++)
626        for(j=0;j<var;j++)
627            a[i][j]=a[i][j];
628    for(i=0;i<equ;i++)
629        for(j=0;j<var;j++)
630            a[i][j]=a[i][j];
631    for(i=0;i<equ;i++)
632        for(j=0;j<var;j++)
633            a[i][j]=a[i][j];
634    for(i=0;i<equ;i++)
635        for(j=0;j<var;j++)
636            a[i][j]=a[i][j];
637    for(i=0;i<equ;i++)
638        for(j=0;j<var;j++)
639            a[i][j]=a[i][j];
640    for(i=0;i<equ;i++)
641        for(j=0;j<var;j++)
642            a[i][j]=a[i][j];
643    for(i=0;i<equ;i++)
644        for(j=0;j<var;j++)
645            a[i][j]=a[i][j];
646    for(i=0;i<equ;i++)
647        for(j=0;j<var;j++)
648            a[i][j]=a[i][j];
649    for(i=0;i<equ;i++)
650        for(j=0;j<var;j++)
651            a[i][j]=a[i][j];
652    for(i=0;i<equ;i++)
653        for(j=0;j<var;j++)
654            a[i][j]=a[i][j];
655    for(i=0;i<equ;i++)
656        for(j=0;j<var;j++)
657            a[i][j]=a[i][j];
658    for(i=0;i<equ;i++)
659        for(j=0;j<var;j++)
660            a[i][j]=a[i][j];
661    for(i=0;i<equ;i++)
662        for(j=0;j<var;j++)
663            a[i][j]=a[i][j];
664    for(i=0;i<equ;i++)
665        for(j=0;j<var;j++)
666            a[i][j]=a[i][j];
667    for(i=0;i<equ;i++)
668        for(j=0;j<var;j++)
669            a[i][j]=a[i][j];
670    for(i=0;i<equ;i++)
671        for(j=0;j<var;j++)
672            a[i][j]=a[i][j];
673    for(i=0;i<equ;i++)
674        for(j=0;j<var;j++)
675            a[i][j]=a[i][j];
676    for(i=0;i<equ;i++)
677        for(j=0;j<var;j++)
678            a[i][j]=a[i][j];
679    for(i=0;i<equ;i++)
680        for(j=0;j<var;j++)
681            a[i][j]=a[i][j];
682    for(i=0;i<equ;i++)
683        for(j=0;j<var;j++)
684            a[i][j]=a[i][j];
685    for(i=0;i<equ;i++)
686        for(j=0;j<var;j++)
687            a[i][j]=a[i][j];
688    for(i=0;i<equ;i++)
689        for(j=0;j<var;j++)
690            a[i][j]=a[i][j];
691    for(i=0;i<equ;i++)
692        for(j=0;j<var;j++)
693            a[i][j]=a[i][j];
694    for(i=0;i<equ;i++)
695        for(j=0;j<var;j++)
696            a[i][j]=a[i][j];
697    for(i=0;i<equ;i++)
698        for(j=0;j<var;j++)
699            a[i][j]=a[i][j];
700    for(i=0;i<equ;i++)
701        for(j=0;j<var;j++)
702            a[i][j]=a[i][j];
703    for(i=0;i<equ;i++)
704        for(j=0;j<var;j++)
705            a[i][j]=a[i][j];
706    for(i=0;i<equ;i++)
707        for(j=0;j<var;j++)
708            a[i][j]=a[i][j];
709    for(i=0;i<equ;i++)
710        for(j=0;j<var;j++)
711            a[i][j]=a[i][j];
712    for(i=0;i<equ;i++)
713        for(j=0;j<var;j++)
714            a[i][j]=a[i][j];
715    for(i=0;i<equ;i++)
716        for(j=0;j<var;j++)
717            a[i][j]=a[i][j];
718    for(i=0;i<equ;i++)
719        for(j=0;j<var;j++)
720            a[i][j]=a[i][j];
721    for(i=0;i<equ;i++)
722        for(j=0;j<var;j++)
723            a[i][j]=a[i][j];
724    for(i=0;i<equ;i++)
725        for(j=0;j<var;j++)
726            a[i][j]=a[i][j];
727    for(i=0;i<equ;i++)
728        for(j=0;j<var;j++)
729            a[i][j]=a[i][j];
730    for(i=0;i<equ;i++)
731        for(j=0;j<var;j++)
732            a[i][j]=a[i][j];
733    for(i=0;i<equ;i++)
734        for(j=0;j<var;j++)
735            a[i][j]=a[i][j];
736    for(i=0;i<equ;i++)
737        for(j=0;j<var;j++)
738            a[i][j]=a[i][j];
739    for(i=0;i<equ;i++)
740        for(j=0;j<var;j++)
741            a[i][j]=a[i][j];
742    for(i=0;i<equ;i++)
743        for(j=0;j<var;j++)
744            a[i][j]=a[i][j];
745    for(i=0;i<equ;i++)
746        for(j=0;j<var;j++)
74
```

```

26  int max_r; // 当前这列绝对值最大的行.
27  int col; // 当前处理的列
28  int ta, tb;
29  int LCM;
30  int temp;
31  int free_x_num;
32  int free_index;
33
34  for(int i = 0; i <= var; i++){
35      x[i] = 0;
36      free_x[i] = true;
37  }
38
39  // 转换为阶梯阵.
40  col = 0; // 当前处理的列
41  for(k = 0; k < equ && col < var; k++, col++) { // 枚举当前处理的行.
42      // 找到该col列元素绝对值最大的那行与第k行交换.(为了在除法时减小误差)
43      max_r = k;
44      for(i = k + 1; i < equ; i++)
45          if(abs(a[i][col]) > abs(a[max_r][col]))
46              max_r = i;
47
48      if(max_r != k) // 与第k行交换.
49          for(j = k; j < var + 1; j++)
50              swap(a[k][j], a[max_r][j]);
51
52      if(a[k][col] == 0) { // 说明该col列第k行以下全是0了, 则处理当前行的下一列.
53          k--;
54          continue;
55      }
56  }
57  for(i = k + 1; i < equ; i++) { // 枚举要删去的行.
58      if(a[i][col] != 0) {
59          LCM = lcm(abs(a[i][col]), abs(a[k][col]));
60          ta = LCM / abs(a[i][col]);
61          tb = LCM / abs(a[k][col]);
62          if(a[i][col] * a[k][col] < 0)
63              tb = -tb; // 异号的情况是相加
64          for(j = col; j < var + 1; j++)
65              a[i][j] = a[i][j] * ta - a[k][j] * tb;
66      }
67  }
68  }
69  // 1. 无解的情况: 化简的增广阵中存在(0, 0, ..., a)这样的行(a != 0).
70  for (i = k; i < equ; i++) // 对于无穷解来说, 如果要判断哪些是自由变元, 那么初等行变换中的交换就会影响, 则要记录交换.
71      if (a[i][col] != 0)
72          return -1;
73
74  // 2. 无穷解的情况: 在var * (var + 1)的增广阵中出现(0, 0, ..., 0)这样的行, 即说明没有形成严格的上三角阵.
75  // 且出现的行数即为自由变元的个数.
76  if (k < var)
77      return var - k; // 自由变元有var - k个.
78
79  // 3. 唯一解的情况: 在var * (var + 1)的增广阵中形成严格的上三角阵.
80  // 计算出Xn-1, Xn-2 ... X0.
81  for (i = var - 1; i >= 0; i--) {
82      temp = a[i][var];
83      for (j = i + 1; j < var; j++)
84          if (a[i][j] != 0)
85              temp -= a[i][j] * x[j];
86      if (temp % a[i][i] != 0)
87          return -2; // 说明有浮点数解, 但无整数解.
88      x[i] = temp / a[i][i];
89  }
90  return 0;
91  }
92  }
93  int main(void) {
94      int i, j;
95      int equ, var;

```



1



1











```
97     memset(a, 0, sizeof(a));
98     for (i = 0; i < equ; i++)
99         for (j = 0; j < var + 1; j++)
100             scanf("%d", &a[i][j]);
101
102     int free_num = Gauss(equ,var);
103     if (free_num == -1)
104         printf("无解!\n");
105     else if (free_num == -2)
106         printf("有浮点数解, 无整数解!\n");
107     else if (free_num > 0) {
108         printf("无穷多解! 自由变元个数为%d\n", free_num);
109         for (i = 0; i < var; i++) {
110             if (free_x[i])
111                 printf("x%d 是不确定的\n", i + 1);
112             else
113                 printf("x%d: %d\n", i + 1, x[i]);
114         }
115     } else
116         for (i = 0; i < var; i++)
117             printf("x%d: %d\n", i + 1, x[i]);
118         printf("\n");
119     }
120     return 0;
121 }
```

白开水+它，喝完排出“巨便”，大肚子没了

舜飞



想对作者说点什么

高斯消元(and牛顿迭代法)详解及模板

阅读数 1113

高斯消元的最主要作用为求解线性方程组，说白了，就是解方程.....消元的过程就是模拟人手算解方程的过程...

博文 来自: 风灯记的博客

高斯消元

阅读数 739

题目背景Gauss消元题目描述给定一个线性方程组，对其求解输入输出格式输入格式： 第一行，一个正整数 n第二至 ...

博文 来自: sslz_fsy的博客

高斯消元快速入门

阅读数 2万+

高斯消元快速入门一、基本描述学习一个算法/技能，首先要知道它是干什么的，那么高斯消元是干啥的呢？高斯消...

博文 来自: Pengwill's Blog

高斯消元法

阅读数 1937

高斯消去法是一种常用的求解线性方程组的方法，通过逐次消元后，在回代求解，实际计算中常用的一种方法。顺序...

博文 来自: tyxr



网上商城项目完整源码

高斯消元法(python实现)

阅读数 1072

今天在看一篇论文是讲fastmultipolemethod的。好久之前就看过相关的知识，主要是用来加速矩阵和向量之间的乘...

博文 来自: bengapai的博客

Java的高斯消元法

阅读数 38

算法虽然会，但是用数组的形式进行求值是一件非常非常恶心的事情。更恶心的是，公司里面的逻辑居然要用到方程...

博文 来自: 柠檬不萌

计算行列式 (高斯消元? +Java+工具)

阅读数 860

功能：计算行列式并输出用法：首先输入行列式的阶数，然后以输入行列式内容。例如：输入：412-1323-12-1110...

博文 来自: ACM_Fish的博客

高斯消元法解线性方程--Java实现

阅读数 2188

我想当你看到这篇文章的时候，已经对高斯消元法进行了一些了解了，如果还有不明白的地方，请大家自行百度，我...

博文 来自: 郑小小小源的博客

高斯消元法求解方程组

阅读数 1611

华为云年中大促 消费送P30 [广告][关闭]







