What is your decision on a new requirement?

Gauss-Jordon Elimination

Solve a set of linear equations

 Gauss-Jordon Elimination (http://en.wikipedia.org/wiki/ Gauss%E2%80%93Jordan_elimination)

> $L_1: 2 x + y - z = 8$ $L_2: -3 x - y + 2 z = -11$ $L_3: -2 x + y + 2 z = -3$

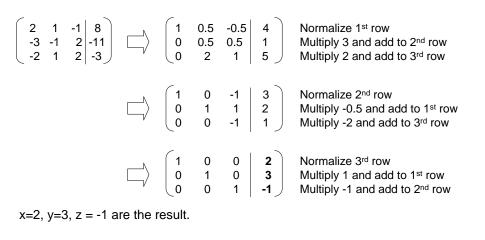
□ Basic algorithm:

- eliminate x from all equations except L1, then
- ${\mbox{ \bullet eliminate y from all equations except L}_{2^{\prime}}$ and
- eliminate z from all equations except L₃.

This will put the system into reduced row-echelon form. Each unknown variables can be solved.

Gauss-Jordon Elimination (cont'd)

> In matrix notation:



Write a C program to do this!

3

Matrix Inverse

$ \left(\begin{array}{rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr$	1 0 0 1 0 0	0 0 1	\Box	(1 0 0	0.5 0.5 2	-0.5 0.5 1	0.5 1.5 1	0 1 0	0 0 1
			\Box	(1 0 0	0 1 0	-1 1 -1	-1 3 -5	-1 2 -4	0 0 1
			\Box	$ \left(\begin{array}{c} 1\\ 0\\ 0 \end{array}\right) $	0 1 0	0 0 1	4 -2 5	3 -2 4	-1 1 -1
2 1 -3 -1 -2 1	-1 2 2	4 -2 5	3 -2 4	-1 1 -1	=	1 0 0	0 0 1 0 0 1		

Assume you have the previous program, write another C program to do this! How would you do it? 2

How would you do it?

- 1. Write a totally new program to do it?
 - Old program might still have bugs, write a new one might avoid that at all.
 - Old program is difficult to read and modify. Even if old program does not have bugs, there will be some bugs after the modification.

or

2. Adapt the old program to satisfy the new requirement?

5

or

3. Modularize suitably each components in the old program, test the modularized old program, and extract useful components from it to compose the new program?