1. Solve the following set of equations

$$A_i x_i = b_i, i = 1, 2, 3 \tag{1}$$

For A_1 and b_1(lineq1.dat), the solutuons for each method are

1-1. Using Gauss-Jordan Elimination(`gaussj()`)

- [nan, nan, nan, nan]

1-2. Using LU Decomposition(`ludcmp()`)

- [1.000000, -3.000000, 2.000000, 0.000000]
- In fact, the result should not be calculated, but the tiny epsilon value used in NR library does not stop the division by zero.

1-3. Using Singular Value Decomposition(`svdcmp()`)

- [1.733333, -1.533334, -0.200000, -0.733334]
- The result of singular value decomposition about any singular matrix can be interpreted as a solution of least square method, which is located to the minimum distance.

For A_2 and b_2(lineq2.dat), the solutuons for each method are

1-1. Using Gauss-Jordan Elimination(`gaussj()`)

- [-2.873567, -0.612357, 0.976277, 0.635819, -0.553441]

1-2. Using LU Decomposition(`ludcmp()`)

- [-2.873566, -0.612357, 0.976277, 0.635819, -0.553441]

1-3. Using Singular Value Decomposition(`svdcmp()`)

- [-2.873566, -0.612357, 0.976278, 0.635819, -0.553441]

All results are same, because it's non-singular.

For A_3 and b_3(lineq3.dat), the solutuons for each method are

1-1. Using Gauss-Jordan Elimination(`gaussj()`)

- [-0.326608, 1.532293, -1.044826, -1.587447, 2.928480, -2.218931]

1-2. Using LU Decomposition(`ludcmp()`)

- [-0.326608, 1.532292, -1.044826, -1.587447, 2.928480, -2.218930]

1-3. Using Singular Value Decomposition(`svdcmp()`)

- [-0.326609, 1.532292, -1.044825, -1.587447, 2.928479, -2.218929]

All results are same, because it's non-singular.

Discuss empirically the advantage/disadvantage of each method

Method	Advantage	Disadvantage
Gauss-Jordan Elimination	Good for obtaining an inverse	Computations are heavy O(n^3). Cannot be used for singular one
LU Decomposition	Good for getting a solution and a determinant. Able to know whether a matrix is ill-conditioned or not	Substitution is needed. Cannot be used for singular one
Singular Value Decomposition	Can also be used for singular one. Can be applied to over-/under-determined.	When the dimensions are large, calculation costs are heavy

2. Apply the method of iterative improvement(`mprove()`) to the above problem and discuss the results

For A_1 and b_1(lineq1.dat), the solutuon is

- [1.000000, -3.000000, 2.000000, 0.000000]

- Because the A_1 is singular, the inverse does not exist.
- The result of LU decomposition in `mprove()` cannot be used for calculating the correct inverse, therefore.

For A_2 and b_2(lineq2.dat), the solutuon is

- [-2.873566,   -0.612357,   0.976277,   0.635819,   -0.553441]
- The result is same as 'Section 1'

For A_3 and b_3(lineq3.dat), the solutuon is

- [-0.326608,   1.532292,   -1.044826,   -1.587447,   2.928480,   -2.218930]
- The result is same as 'Section 1'

3. Find the inverse and the determinant of the matrix A_i in the above problem

For A_1 and b_1(lineq1.dat), the inverse of A_1 and determinant of A_1 are

- Inverse:

$$\left(\begin{matrix} -1000000020040877734272.000000 & -3750000007515329003520.000000 & 375000007515329003520.000000 & 125000013500225945 \\ -2000000567847336017920.000000 & -7500001909525184512.000000 & 7500001029915882291.2000000 & 250000048990684446 \\ 30000004119663529164.8000000 & 1125000154487382343.68000000 & -1125000066526452121.60000000 & -37500005149579411 \\ 1000000020040877734272.000000 & 375000007515329003520.000000 & -3749999635348638924.8000000 & -125000013500225945 \end{matrix}\right.$$

- Determinant: -0.000000
- Since the determinant is 0, the matrix are singular.
- Which means its inverse must be very large or incomputable by the division by zero, because **some pivot chosen for Gauss-Jordan Elimination should be elinimated to a closed-zero value**, which incurs explosion of inverses.

For A_2(lineq2.dat), the inverse of A_2 and determinant of A_2 are

- Inverse:

$$\begin{pmatrix} 0.354536 & 0.766945 & 0.207769 & -0.595412 & 0.253128 \\ 0.035454 & 0.126695 & 0.195777 & -0.159541 & 0.050313 \\ -0.138686 & -0.098540 & -0.096715 & 0.124088 & 0.016423 \\ -0.052138 & -0.303962 & -0.023201 & 0.234619 & -0.044578 \\ 0.149114 & 0.459333 & 0.051356 & -0.171011 & 0.042492 \end{pmatrix} \tag{3}$$

- Determinant: 3835.999512

For A_3(lineq3.dat), the inverse of A_3 and determinant of A_3 are

- Inverse:

$$\begin{pmatrix} -0.162205 & 0.122801 & 0.024068 & -0.016431 & -0.022840 & 0.046132 \\ 0.169407 & -0.041117 & 0.228313 & -0.087624 & 0.180306 & -0.395655 \\ -0.011636 & 0.122745 & -0.117407 & -0.180981 & 0.015910 & 0.186766 \\ 0.105669 & -0.051726 & -0.108916 & 0.299774 & 0.000859 & -0.190541 \\ -0.053026 & -0.042362 & 0.160508 & -0.224034 & 0.161811 & 0.015024 \\ -0.062341 & -0.064694 & -0.234216 & 0.351126 & -0.364828 & 0.434633 \end{pmatrix} \tag{4}$$

- Determinant: 16178.401367

How to compile and run

1. `./build.sh`
2. `./main lineq1.dat` to run with 'lineq1.dat'
3. `./main lineq2.dat` to run with 'lineq2.dat'
4. `./main lineq3.dat` to run with 'lineq3.dat'