Linear System Solver

I used Python as a programming language and the Gaussian elimination method to solve linear equations. Firstly I implement 3 elementary row operations as functions. Source code is well commented that every function, its parameters and return values are described elaborately.

After that, I wrote a function to parse the input and create the matrix. I use a 2 dimensional list (list of lists of floats) to implement the matrix. While parsing the data for the first time, I place Identity Matrix between A and b. I keep our matrix in the form (A | I | b) in order to get A^{-1} after applying elementary row operations to A matrix to turn it into Identity. After reaching the identity matrix in A, the values in the b column will be the values of x_1,x_2 …

In order to reach the solution, I wrote a function namely solve() which uses Gaussian elimination method and 3 elementary row operations. I pivot elements one by one and when I reach the identity matrix, b vector will be the solution set as Gaussian elimination is suggested.

In the case that there is a unique solution, I can get the injected matrix which was initially the identity matrix to retrieve A^{-1}. Injected matrix will be turned into A^{-1}. As told before, the values in the b column will be the values of the unique solution.

In the cases that there are infinitely many solutions and no solutions at all, I would encounter a situation where there were no appropriate pivot values. For that case, I wrote another function namely irregular() , which handles irregular cases.

First of all, I got rid of the injected matrix because I don’t need it anymore. Then I continue pivoting operation but in the same time, I started to store not-pivoted variables. While passing by non-pivoted variables, regarding indexing arrangements are done properly.

After traverse, I should determine whether I am faced with an inconsistent case or an infinitely many solution case. I stored the zero rows at the bottom of the matrix. The important thing is whether the value of the b vector of this row is also zero or not.

If all the zero rows have zero b values, that means I have infinitely many solutions with previously stored arbitrary variables. For the sake of the simplicity, I decided to choose arbitrary variables as zero and generate one arbitrary solution regarding this decision.

Or if all the zero rows don’t have zero b values, that means I have an inconsistent case. The previously stored variables means nothing anymore. I just did state that I have no solution at all and returned.

Our code is written in the most modular way and can be executed with any size of input matrix. No specific bugs are encountered. I haven’t used any additional linear algebra library. Just pure Python code is written.

I didn’t round the result values but it is a very easy operation.

Example output is below:

```
PS D:\Documents\Lectures\Ie310> python .\20201101_24_cakir_akar_atasoy_canfes_hw2.py
Dataset  1
Arbitrary variables:  3
Arbitrary solution: 6.6 1.8 0
--------------
Dataset  2
Unique solution: 1.0 -0.5 1.5
Inverted A:  0.5 0.16666666666666663 0.3333333333333333
             1.0 0.4 0.2
             0.0 0.13333333333333333 0.06666666666666667
--------------
Dataset  3
Inconsistent problem
--------------
Dataset  4
Unique solution: 1.5972501852833867 0.27672433978110167 0.767265904263867 -0.6920236736295844 0.9740575620299556 -1.0377488493082427
Inverted A:  0.21301962760112425 -0.09309971898521985 0.038048145645968826 0.0504653900111624 -0.045651773054406040 -0.1560727740229731 8
             -0.0006601892832170941 -0.0473208042147872 4 -0.04133053410971385 0.10472686840082174 -0.07131979023985952 0.16833326291840778
             -0.05323210430990666 -0.026077476687066004 0.06936765159379249 -0.02939145315479269 0.08669327692768754 -0.06338685788991341
             -0.06617963229194268 0.09192701433740019 -0.022300735961007782 -0.028136224846571106 -0.043478518380467235 0.07884760643874078
             0.08242376333951532 -0.059165121157763484 -0.02773624174615328 0.03025143656319358 -0.007899370095715591 0.051995433954024436
             -0.14055951041752623 0.12239561842796767 -0.01730035493070974 -0.07124397903030355 0.07023829595353721 -0.02415574149483121
--------------
PS D:\Documents\Lectures\Ie310>
```