Problem Name:

System of Linear Equations using Gauss Elimination Method.

Apparatus:

- Computing Device (e.g.; Computer)
- Programming Environment (MATLAB R2016b)

Method:

The Gauss Elimination method is a systematic approach for solving systems of linear equations by transforming the augmented matrix of the system into row-echelon form and then back-substituting to find the solution. The method involves a series of elementary row operations to simplify the matrix until it is in a triangular form.

Algorithm:

1. Start
2. Read Number of Unknowns: n
3. Read Augmented Matrix (A) of n by n+1 Size
4. Transform Augmented Matrix (A)

   to Upper Triangular Matrix by Row Operations.
5. Obtain Solution by Back Substitution.
6. Display Result.
7. Stop

Code:

```
 clc;

y = input('Please Enter the size of the equation system n = ');
C = input('Please Enter the elements of the Matrix C ');
b = input('Please Enter the elements of the Matrix b ');

dett = det(C);

if dett == 0
   disp('This system is unsolvable because det(C) = 0 ')
else
   b = b';
   a = [C b];
end

% Gauss elimination method
[m, n] = size(a);

for j = 1:m-1
   for z = 2:m
      if a(j, j) == 0
         t = a(j, :);
         a(j, :) = a(z, :);
         a(z, :) = t;
      end
   end
   for i = j+1:m
      a(i, :) = a(i, :) - a(j, :) * (a(i, j) / a(j, j));
   end
end

x = zeros(1, m);

% Back-substitution
for s = m:-1:1
   c = 0;
```

```
    for k = s+1:m
        c = c + a(s, k) * x(k);
    end
    x(s) = (a(s, n) - c) / a(s, s);
end

disp('Gauss elimination method:');
disp(a);
disp('Solution:');
disp(x');
```

Input:

Please Enter the size of the equation system n = 3

Please Enter the elements of the Matrix C

[1   1   1; 2 1 1; 1 1 2]

Please Enter the elements of the Matrix b

[1   -1   2]

Output:

GaussElimination_1039.m

/MATLAB Drive/GaussElimination_1039.m

```matlab
clc;
y = input('Please Enter the size of the equation system n = ');
C = input('Please Enter the elements of the Matrix C ');
b = input('Please Enter the elements of the Matrix b ');

dett = det(C);

if dett == 0
    disp('This system is unsolvable because det(C) = 0 ')
else
    b = b';
    a = [C b];
end

% Gauss elimination method
[m, n] = size(a);

for j = 1:m-1
    for z = 2:m
        if a(j, j) == 0
            t = a(j, :);
            a(j, :) = a(z, :);
            a(z, :) = t;
        end
    end
    for i = j+1:m
        a(i, :) = a(i, :) - a(j, :) * (a(i, j) / a(j, j));
    end
end

x = zeros(1, m);

% Back-substitution
for s = m:-1:1
    c = 0;
    for k = s+1:m
        c = c + a(s, k) * x(k);
    end
    x(s) = (a(s, n) - c) / a(s, s);
end

disp('Gauss elimination method:');
disp(a);
disp('Solution:');
disp(x');
```

Command Window

```
Please Enter the size of the equation system n =
3
Please Enter the elements of the Matrix C
[1 1 1; 2 1 1; 1 1 2]
Please Enter the elements of the Matrix b
[1 -1 2]
Gauss elimination method:
     1     1     1     1
     0    -1    -1    -3
     0     0     1     1

Solution:
    -2
     2
     1

>> 
```

Code Issues

Discussion:

In this report, we implemented the Gauss Elimination method for solving systems of linear equations using MATLAB. The Gauss Elimination method is a systematic approach that transforms the augmented matrix of a system into row-echelon form and then uses back-substitution to find the solution. It is a widely-used numerical technique known for its simplicity and efficiency.

During the implementation, we encountered a challenge when using MATLAB online. The command window did not display the output as expected, requiring multiple attempts to troubleshoot and successfully observe the intermediate steps and final solution. This experience highlighted the importance of understanding the intricacies of the chosen development environment and the potential hurdles associated with online platforms.

Despite the technical issues faced, the implementation of the Gauss Elimination method provided valuable insights into the numerical solution of linear systems. By navigating through the steps of row reduction and back-substitution, we gained a deeper understanding of the method's mechanics and its role in solving real-world problems. This hands-on experience underscored the significance of robust coding practices and adaptability in overcoming challenges during algorithmic implementations.