**Experiment: 02**

**Date:02.03.2022 & 09.03.2022**

**Title: Gauss elimination for solving system of linear equations & Basic Solution.**

**Program Name:**

1. **Write a MATLAB code for solving a system of linear equations using Gauss elimination with partial pivoting.**
2. **Use this above code as a function and solve simultaneous equations with unknowns to obtain basic solutions.**
3. **Check your program for the following examples and count the number of basic solutions:**

**Algorithm:**

* **a) Gauss Elimination with Partial Pivoting Algorithm:**

1. Start
2. Read the coefficients matrix
3. Read the right hand side vector for
4. Find the order of
5. If number of row is equal to the number of column of , then

Goto step 6

Otherwish print an error , “”

1. Create an augmented matrix A by appending as a column to

For

Find the maximum absolute value of -th column and identify the row index of it say

Swap -th row with -th row

For

Set factor

For

factor

Repeat

Repeat

Repeat

Initialize for

For

Set

For

Repeat

Repeat

**Program Code:**

1. **Program Code :(For solving a system of linear equations using Gauss elimination with partial pivoting.):**

**%ANINDYA NAG**

**%UG/02/BTCSE/2018/005**

**clc;**

**clearvars;**

**A= input('Enter the coefficients matrix : ');**

**[m,n] = size(A);**

**b= input('Enter the right hand side vector : ');**

**if ( m == n)**

**Aug =[A b];**

**for k = 1 : (n-1)**

**[big ip]=max(abs(Aug(k:m,k)));**

**ipr= ip +k-1 ;**

**if (ipr ~= k)**

**Aug([k ipr],:) =Aug([ipr k],:);**

**end**

**for i = (k +1) : n**

**factor= Aug(i,k)/Aug(k,k);**

**for j=1 : (n+1)**

**Aug(i,j) =(Aug(i,j)-(factor\*Aug(k,j)));**

**end**

**end**

**end**

**x=[];**

**x(n)= Aug(n,n+1)/ Aug(n,n);**

**for i=(n-1):-1: 1**

**Sum=0;**

**for j = n:-1:(i+1)**

**Sum =Sum + Aug(i,j)\*x(j) ;**

**end**

**x(i) =( Aug(i,n+1) -Sum)/Aug(i,i);**

**end**

**else**

**disp('This system of equations have no solution...');**

**end**

**A**

**b**

**Aug**

**x**

**Output: For**

**>> EXP\_2**

**Enter the coefficients matrix : [0 1 2;1 1 1;2 -1 2]**

**Enter the right hand side vector : [12;15;14]**

**A =**

**0 1 2**

**1 1 1**

**2 -1 2**

**b =**

**12**

**15**

**14**

**Aug =**

**2.0000 -1.0000 2.0000 14.0000**

**0 1.5000 0 8.0000**

**0 0 2.0000 6.6667**

**x =**

**6.3333 5.3333 3.3333**

**>>**

**Output: For**

**>> EXP\_2**

**Enter the coefficients matrix : [0 1 2;1 1 1]**

**Enter the right hand side vector : [12;15]**

**This system of equations have no solution...**

**>>**

**b)Program Code :( Use this above code as a function and solve simultaneous equations with unknowns to obtain basic solutions. )**

* **Program Code : For Build a function gauss\_elimination**

**(gauss\_elimination.m)**

**%ANINDYA NAG**

**%UG/02/BTCSE/2018/005**

**function x = gauss\_elimination(A,b)**

**[m,n] = size(A);**

**if ( m == n)**

**Aug =[A b];**

**for k = 1 : (n-1)**

**[big ip] = max(abs(Aug(k:m,k)));**

**ipr = ip +k-1 ;**

**if (ipr ~= k)**

**Aug([k ipr],:) =Aug([ipr k],:);**

**end**

**for i = (k +1) : n**

**factor = Aug(i,k)/Aug(k,k);**

**for j=1 : (n+1)**

**Aug(i,j) =(Aug(i,j)-(factor\*Aug(k,j)));**

**end**

**end**

**end**

**x = [];**

**x(n) = Aug(n,n+1)/ Aug(n,n);**

**for i=(n-1):-1: 1**

**Sum = 0;**

**for j = n:-1:(i+1)**

**Sum = Sum + Aug(i,j)\*x(j) ;**

**end**

**x(i) =( Aug(i,n+1) -Sum)/Aug(i,i);**

**end**

**end**

**end**

* **Program Code : For obtaining basic solutions**

**%ANINDYA NAG**

**%UG/02/BTCSE/2018/005**

**clc;**

**clearvars;**

**A = input('Enter the coefficients matrix : ');**

**b = input('Enter the right hand side vector : ');**

**[m,n] = size(A);**

**combo = nchoosek(1:n,m);**

**[m1 n1] = size(combo);**

**nbs = nchoosek(n,m);**

**for i = 1 : nbs**

**x = zeros(n,1);**

**bas\_mat = [ ];**

**for j = 1 :n1**

**bas\_mat = [bas\_mat A(:,combo(i,j))];**

**end**

**bas\_mat;**

**y = gauss\_elimination(bas\_mat,b);**

**if( max(y) ~= Inf )**

**for j = 1 : n1**

**x(combo(i,j),1) = y(j);**

**end**

**disp('Basic solution is: ');**

**x**

**if( min(x) >= 0 )**

**disp(' This is a Basic Feasible solution.');**

**else**

**disp (' This is a Basic Infeasible solution.');**

**end**

**if(y == 0)**

**disp('This is Degenerate Basic solution.');**

**else**

**disp ('This is Non Degenerate Basic solution.' );**

**end**

**else**

**disp ( 'In this case no Basic solution exist. ' );**

**end**

**end**

**Output: For**

**Enter the coefficients matrix : [0 1 2;1 1 1;2 -1 2]**

**Enter the right hand side vector : [12;15;14]**

**Basic solution is:**

**x =**

**6.3333**

**5.3333**

**3.3333**

**This is a Basic Feasible solution.**

**This is Non Degenerate Basic solution.**

**>>**

**c) Check your program for the following examples and count the number of basic solutions:**

**Output:** For

**Enter the coefficients matrix : [1 1 1 0;2 1 0 1]**

**Enter the right hand side vector : [40;60]**

**Basic solution is:**

**x =**

**20**

**20**

**0**

**0**

**This is a Basic Feasible solution.**

**This is Non Degenerate Basic solution.**

**Basic solution is:**

**x =**

**30**

**0**

**10**

**0**

**This is a Basic Feasible solution.**

**This is Non Degenerate Basic solution.**

**Basic solution is:**

**x =**

**40**

**0**

**0**

**-20**

**This is a Basic Infeasible solution.**

**This is Non Degenerate Basic solution.**

**Basic solution is:**

**x =**

**0**

**60**

**-20**

**0**

**This is a Basic Infeasible solution.**

**This is Non Degenerate Basic solution.**

**Basic solution is:**

**x =**

**0**

**40**

**0**

**20**

**This is a Basic Feasible solution.**

**This is Non Degenerate Basic solution.**

**Basic solution is:**

**x =**

**0**

**0**

**40**

**60**

**This is a Basic Feasible solution.**

**This is Non Degenerate Basic solution.**

**>>**

**Output: For**

**Enter the coefficients matrix : [1 1 1 0 0;1 1 0 1 0;1 0 0 0 1]**

**Enter the right hand side vector : [100;80;40]**

**Basic solution is:**

**x =**

**40**

**40**

**20**

**0**

**0**

**This is a Basic Feasible solution.**

**This is Non Degenerate Basic solution.**

**Basic solution is:**

**x =**

**40**

**60**

**0**

**-20**

**0**

**This is a Basic Infeasible solution.**

**This is Non Degenerate Basic solution.**

**Basic solution is:**

**x =**

**NaN**

**-Inf**

**0**

**0**

**-Inf**

**This is a Basic Infeasible solution.**

**This is Non Degenerate Basic solution.**

**Basic solution is:**

**x =**

**40**

**0**

**60**

**40**

**0**

**This is a Basic Feasible solution.**

**This is Non Degenerate Basic solution.**

**Basic solution is:**

**x =**

**80**

**0**

**20**

**0**

**-40**

**This is a Basic Infeasible solution.**

**This is Non Degenerate Basic solution.**

**Basic solution is:**

**x =**

**100**

**0**

**0**

**-20**

**-60**

**This is a Basic Infeasible solution.**

**This is Non Degenerate Basic solution.**

**In this case no Basic solution exist .**

**Basic solution is:**

**x =**

**0**

**80**

**20**

**0**

**40**

**This is a Basic Feasible solution.**

**This is Non Degenerate Basic solution.**

**Basic solution is:**

**x =**

**0**

**100**

**0**

**-20**

**40**

**This is a Basic Infeasible solution.**

**This is Non Degenerate Basic solution.**

**Basic solution is:**

**x =**

**0**

**0**

**100**

**80**

**40**

**This is a Basic Feasible solution.**

**This is Non Degenerate Basic solution.**

**>>**