

```
function X= Gauss_elimination(A,B)
% Length of various matrices
col_A= size(A,2);
Y=[A B]; % Y= [A|B] format to minimize separate operations on A and B.
row_Y= size(Y,1);
col_Y= size(Y,2);

counter=0;
for i= 1 : row_Y
    % Pivot and largest diagonal element Condition
    diagonal_max=Y(i,:);
    max_row=i;
    for k= i+1 : row_Y
        if(abs(Y(k,i)) > diagonal_max(i)) % Condition
            diagonal_max=Y(k,:);
            max_row=k;
        end
    end
    Y(max_row,:)=Y(i,:); % Updating pivot value to max in column
    Y(i,:)=diagonal_max;

    % Gauss-elimination method
    if abs(Y(i,i)) > 1e-4 % Condition to ensure no operation is done
on NULL element
        for j=i+1 : row_Y

            factor= Y(j,i)./Y(i,i); % calculating factor
            counter= counter+1;
            Y(j,:)=Y(j,:)-factor.*Y(i,:); % updating subsequent rows
            counter = counter + 2.*(col_Y - i);

        end
    end
end
% Back Substitution
X = zeros(1,col_A); % Initializing X vector to 0
col_X= size(X,2);
for i=row_Y:-1:1
    temp= sum(Y(i,i+1:col_Y-1).*X(i+1:col_X));
    counter= counter + 2.*(col_X -i) -1;
    X(i)= (Y(i,col_Y)-temp)./Y(i,i); % Backsubstitution formula
    counter= counter +2;
end
fprintf('No. of operation = %i\n', counter);
end
```

Published with MATLAB® R2021a