

## Gaussian Elimination

```
clear all;
%Change matrix and result vector
%Must be invertible and non-singular
A = [1 1 1 1;5 1 2 1;0 2 3 1;0 1 3 1];
b = [-8; 3; -3; 10];
cols = size(A,1); %getting dimensions
rows = size(A,2); %getting dimensions
if (cols ~= rows)
    %only square matrix control is made
    return;
end
dim = cols;
disp('MatLab inv(A)*b Result:');
inv(A)*b
A = [A,b]; %augmentation
%Elimination
for i = 1:dim-1
    for j = i+1:dim
        m = A(j,i)/A(i,i);
        for k = 1:dim+1
            A(j,k) = A(j,k) - m*A(i,k);
        end
    end
end
disp('Lower Triangle:');
A
%Back Substitution
res (dim) = A(dim,dim+1) / A(dim,dim);
k = dim-1;
while (k>=1)
    l = dim;tot = 0;
    while (l>=k)
        tot = tot + A(k, l) * res(l);
        l = l-1;
    end
    res(k) = (A (k,dim+1) - tot) / A(k,k);
    k = k -1;
end
disp('Result Vector:');
res
```

MatLab inv(A)\*b Result:

ans =

```
0.4444
-13.0000
9.2222
-4.6667
```

Lower Triangle:

A =

```
1.0000 1.0000 1.0000 1.0000 -8.0000
0 -4.0000 -3.0000 -4.0000 43.0000
0 0 1.5000 -1.0000 18.5000
0 0 0 1.5000 -7.0000
```

Result Vector:

res =

```
0.4444 -13.0000 9.2222 -4.6667
```