Assignment 10

Juan Silva

ECE 309

May 9, 2018

**Main Script**

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% Program Name: Solve\_test.m

% Author: Juan Silva Last Modified: May 9, 2018

% Description: This main function receive two inputs, a

% square matrix and a column vector, then produces a table

% comparing our calculated value versus Matlab’s built-in

% inverse function times the original column vector.

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

clear, clc, close all

format short, format compact

%Table variables

var1 = ['x' 'y' 'z']'; %Systems 1 - 3

var2 = ['x' 'y']'; %System 4

var3 = ['v' 'w' 'x' 'y' 'z']'; %System 5

% \*\*\*\*\*\*\*\* Question 1 \*\*\*\*\*\*\*\*

A = [3 4 1; 2 -2 -1; 5 4 0.5];

b = [7 -1 9];

[Ap1, bp1, X1, Xinv1] = Solve(A,b);

table(var1,X1,Xinv1,'VariableNames',{'Unknowns' 'Solve' 'Inv'})

fprintf('\n\t\t\t\tSystem 1\n')

% \*\*\*\*\*\*\*\* Question 2 \*\*\*\*\*\*\*\*

A = [1 0 1; 2 -1 -1; 7 -2 2];

b = [12 1 8];

[Ap2, bp2, X2, Xinv2] = Solve(A,b);

table(var1,X2,Xinv2,'VariableNames',{'Unknowns' 'Solve' 'Inv'})

fprintf('\n\t\tSystem 2\n')

% \*\*\*\*\*\*\*\* Question 3 \*\*\*\*\*\*\*\*

A = [1 -1 2; 0 0 4; 0 2 -1];

b = [22 44 9];

[Ap2, bp2, X2, Xinv2] = Solve(A,b);

table(var1,X2,Xinv2,'VariableNames',{'Unknowns' 'Solve' 'Inv'})

fprintf('\n\t\tSystem 3\n')

% \*\*\*\*\*\*\*\* Question 4 \*\*\*\*\*\*\*\*

A = [0.0001 1; 1 1];

b = [1 2];

[Ap4, bp4, X4, Xinv4] = Solve(A,b);

table(var2,X4,Xinv4,'VariableNames',{'Unknowns' 'Solve' 'Inv'})

fprintf('\n\t\t\t\tSystem 4\n')

% \*\*\*\*\*\*\*\* Question 5 \*\*\*\*\*\*\*\*

A = [0 -1 2 1 1; 1 1 1 -1 1; 2 1 3 2 2; 2 -3 -4 3 0; 1 5 0 0 -1];

b = [4 4 12 7 -1];

[Ap5, bp5, X5, Xinv5] = Solve(A,b);

table(var3,X5,Xinv5,'VariableNames',{'Unknowns' 'Solve' 'Inv'})

fprintf('\n\t\t\t\t\tSystem 5\n')

**Function Script**

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

% Program Name: Solve.m

% Author: Juan Silva Last Modified: May 9, 2018

% Description: This function will produce the row echelon

% form of a given matrix, applies back-substitution to

% produce the solution vector and returns four outputs.

% It will return the upper triangular matrix and its modified

% column vector, the solution column vector, and Matlab’s

% inverse \* b calculation.

% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

function [Ap, bp, X, Xinv] = Solve(A,b)

N = length(A);

S = size(A);

m = S(1);

n = S(2);

if m ~= n

disp('Error. Matrix entered must be a square matrix.')

M = A;

x = b';

return

else

% \*\*\* ROW ECHELON FORM \*\*\*

for k = 1:N %loop through all columns

for i = k:N %loop through all rows

if i == k

if abs(A(i,i)) < 0.01

A([i, N],:) = A([N, i], :);

b(:,[i, N]) = b(:,[N, i]);

end

continue

end

lambda = A(i,k) / A(k,k);

A(i,k:N) = A(i,k:N) - lambda \* A(k,k:N);

b(i) = b(i) - lambda \* b(k);

end

if A(k,k) ~= 1

b(:,k) = b(k) .\* inv(A(k,k));

A(k,:) = A(k,:) .\* inv(A(k,k));

end

end

% \*\*\* BACK SUBSTITUTION \*\*\*

M = [A b'];

A = M(:,1:N);

b = M(:,N+1);

temp = b; %save column before back sub

for i=N:-1:1

%Divide pivot to b column vector

b(i) = b(i) / A(i,i);

%Apply backwards substitution

b(1:i-1) = b(1:i-1) - A(1:i-1,i) \* b(i);

x = b; %back substituion

end

Ap = A; %REF matrix

bp = temp; %column vector after REF

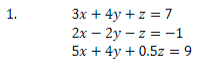
X = b; %column vector after back sub

Xinv = A\temp; %invA \* b

end

end

**Results**



**Unknowns Solve Inv**

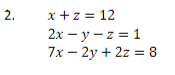
**\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ \_\_\_\_\_\_\_**

**x 0.33333 0.33333**

**y 2.1667 2.1667**

**z -2.6667 -2.6667**

**System 1**



**Unknowns Solve Inv**

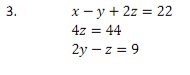
**\_\_\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_**

**x 42 42**

**y 113 113**

**z -30 -30**

**System 2**



**Unknowns Solve Inv**

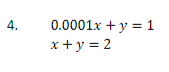
**\_\_\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_**

**x 10 10**

**y 10 10**

**z 11 11**

**System 3**



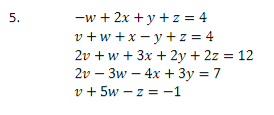
**Unknowns Solve Inv**

**\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_**

**x 1.0001 1.0001**

**y 0.9999 0.9999**

**System 4**

****

**Unknowns Solve Inv**

**\_\_\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_**

**v 2 2**

**w 0 0**

**x 0 0**

**y 1 1**

**z 3 3**

**System 5**