#1

```
A=[2 -4 0 2;0 0 2 6];b=[4;2];linearsystem_solverF(A,b)
disp('-------------------------')
disp(A)
disp(b)
soln=A\b;disp(soln)
disp('---------------------------------')
type linearsystem_solverF
disp('---------------------------------')

% similar solution will be obtained with linsolve(A,b)
disp(linsolve(A,b))
% pseudoinverse will also give the solution
pinv(sym(A))*sym(b)
disp('-------------------------')
```

```
    1   -2    0    1    2
    0    0    1    3    1

system consistent: No PIVOT in the last column of RREF
free variables exist
x_1 BASIC variable
x_3 BASIC variable
   x2 FREE variable
   x4 FREE variable
p-vector
     2
     0
     1
     0

v-vector associated with the free variable x2
     2
     1
     0
     0

v-vector associated with the free variable x4
    -1
     0
    -3
     1

Null space v
     2   -1
     1    0
```

0 -3
0 1

ans =

2 0 1 0

----

2 -4 0 2
0 0 2 6

4
2

0
-0.8333
0
0.3333

```matlab
function linearsystem_solverF(A,b )
% p m shankar, September 2016
% provides solution to a set of linear equations Ax=b. Identifies the
% system as consistent or inconsistent (providing explanations). If the
% system is consistent, the free variables and basic variables are
% identified and if free variables exisit, NULL space is used to obtain the
% solution vector. The null space, vector solution clearly identifying the
% free variables are displayed. The verification of the solution is also
% provided.
% If the system is inconsistent, method of least squares is invoked to
% obtain the solution. Augemneted matrix {Trans(A)*A  Trans(A)*b] is
% created and the RREF is examined to see if a unique solution exists. If a
% unique solution exists, the LS solution, LS Error vector and LS error are
% provided.
% If a unique solution does not exist, free variables are identified and
% the solution is obtianed using the NULL space. Once again, the solution
% vector is displayed clearly identifying the free variable.
%
% display is generated only for coefficient matrices of size 4 x 4 or less
% otherwise, results are displayed on the command window
%
% all results displayed on a single page
close all
Ab=[A,b];
[mx,nx]=size(A);
rA=rank(A);rAb=rank(Ab);
if rA~=rAb;
    disp('The Linear System is inconsistent')
    tex1='INCOnSISTENT system: Last column of RRF has a PIVOT';
    rr1=rref(Ab);
    rr1=(round(rr1*100))/100;
```

```
    figure, xlim([0,5]),ylim([0,3])
    title('Solution of a System of Equations Ax = b','color','b')
    text(-.75,2.75,{'Input Augmented Matrix';'[A b]'},'color','b')
     text(-0.75,2.42, ['No. of Columns of A: n = ',num2str(nx)],'color','r')
     text(1.6,2.42,['rank(A) = ',num2str(rA),'\neq rank([A b]) = ',...
         num2str(rAb)] ,'color','r')
    text(1,2.75,num2str(Ab))
    text(2.5,2.75,'RREF ([A b])','color','b')
    text(3.4,2.75,num2str(rr1))
    axis off
    disp(tex1)
    text(-.75,2.3,'INCOnSISTENT (Last column of RRF has a PIVOT): Least
Squares Solutions',...
        'color','b','fontweight','bold')
    disp('USE LEAST SQUARES SOLUTIONS')
    C=A'*A; bb=A'*b; CC=[C,bb];
    BT=1;% to complete the processing for inconsistent systems
    [p,Kfree]= system_solutionf(CC,BT) ;
    if isempty(Kfree)==1 % give the LSE error if the solution is unique
        text(0,0.1,{'Error vector';'b-Ax'},'color','b')
        text(1.6,.05,num2str(double(b-A*p')))
        text(3,.1,{'Least Squares Error';'||b-Ax||'},'color','b')
        text(4.5,.1,num2str(norm(double(b-A*p'))))
        xr=0:.5:4;
         plot(xr,0.36*ones(length(xr),1),'--k','linewidth',1.2)
    else
    end;
else
    BT=0; %to complete the processing for consistent systems
    system_solutionf(Ab,BT)
end;

if mx>4||nx>4
   close all
else
    clc
end;
end

function [p,Kfree] = system_solutionf(AA,BQ)
% p m shankar September 2016
% obtains the solutions to the matrix equation [A]x=b
% AA is the augmented matrix
A=AA(:,1:end-1);b=AA(:,end);[m,n]=size(A);
[ref1,P1]=rref(A);
[ref2,P2]=rref([A,b]);% P2 are the numbers of the PIVOT columns
disp(ref2)
if BQ==1
    text(-.75,1.95,{'Augmented Matrix';'[A^T.A  A^T.b]'},'color','b')
    text(2.5,1.95,{'RREF of';'[A^T.A  A^T.b]'},'color','b')
    if length(P1)==n
    tex1={'system NOW consistent';'Identify free/basic var'};
      text(-.75,1.3,tex1,'color','r')
      text(4,1.33,{'All variables BASIC';'Solution is UNIQUE';...
          'Empty Null space'},'color','b')
    else
```

```
      tex1={'system consistent';'Identify free/basic var';...
           'Null space [V]'};
         text(-.75,1.35,tex1,'color','r')
    end;
    text(0.8,1.95,num2str(AA))
    ref3=(round(ref2*100))/100;%
    text(3.5,1.95,num2str(ref3))
else
    figure,xlim([0,5]),ylim([0,3]),axis off
    title('Solution of a System of Equations','color','b')
    text(-0.75,2.75,{'Augmented Matrix'; '[A b]'},'color','b')
    text(-0.75,2.42, ['No. of Columns of A: n = ',num2str(n)],'color','b')
     text(2,2.42,['rank(A)= rank([A b])= ',num2str(length(P1))],...
         'color','b')
    text(0.75,2.75,num2str(AA))
    text(2.5,2.75,{'RREF'; '  ([A b])'},'color','b')
    ref3=(round(ref2*100))/100;%
    text(3.3,2.75,num2str(ref3))
    tex1='system consistent: No PIVOT in the last column of RREF';
    if  length(P1)== n
         text(-0.75,2,'Identify free/basic var','color','r')
          text(3.2,1.95,{'All variables BASIC: UNIQUE Solution';...
          'Empty Null space';'rank(A) = rank([A,b]) = n'},'color','b')
    else
    text(-0.75,1.9,{'Identify free/basic var';...
        'Null space [V]';' ';'rank(A)= rank([A,b]) < n'},'color','r')
    end;
    text(-.75,2.3,tex1,'color','r','fontweight','bold')
end;
disp(tex1)
Kbasic=P1;KB=length(P1);
if KB<n
    disp('free variables exist')
else % number of pivot columns=number of columns of A
    disp('all variables basic')
end;
PP=ref2(:,end);% last column of the rref([A b])
% deteremine basic & free variables & rearrange PP to create p-vector
if BQ==0
    yd1=2;
    yd2=2.05;
    xd1=1;
    xd2=2.1;
else
    yd1=1.45;
    yd2=1.35;
      xd1=1;
      xd2=2;
end;
for kb=1:KB
    disp(['x_',num2str(Kbasic(kb)),' BASIC variable'])
    text(xd1,yd1-(kb-1)*.13,['x_',num2str(Kbasic(kb)),...
        ' basic variable'],'fontsize',8,'color','b')
    p(Kbasic(kb))=PP(kb);
end;
Kfree=[];
kbp=1;
```

```
for kff=1:n
    if kff~=Kbasic
        Kfree=[Kfree,kff];
        disp(['    x',num2str(kff),' FREE variable'])
        text(xd2,yd2-(kbp-1)*.13,['x_',num2str(kff),'  free variable'],...
            'fontsize',8,'color','b')
        kbp=kbp+1;
        p(kff)=0;
    else
    end;
end;
disp('p-vector'),  disp(p')
pp=p;% needed for verification before round off
p=(round(p*100))/100;
if isempty(Kfree)~=1
    if BQ==0
        yd0=0.85;
    else
        yd0=0.65;
    end;
    text(-0.75,yd0,...
  {'Solution vector: [x]=[p]+\Sigma_{[k]} x_{[k]}* v_{[k]  \Rightarrow}';...
   ['     with k= [',num2str(Kfree),']']},'color','b','fontweight','bold')
    BP=0;
else
    text(-0.75,.85,{'Solution vector written as [x]=[p]   \Rightarrow';...
        '         UNIQUE Solution'},'color','b')
    BP=1;% for verification purposes
end;
if isempty(Kfree)==1
    for k=1:n
        text(2.5,0.95-(k-1)*.2,['x_',num2str(k),'=
',num2str(p(k))],'color','b')
    end;
else  % free variables exist
    for k=1:n
        if BQ==1
            yd=0.73;
        else
            yd=1.2;
        end;
        text(2.85,yd-(k-1)*.2,['x_',num2str(k),'=  ',num2str(p(k))],...
            'color','b')
    end;
    v=null(sym(A,'r'));
    v=double(v);
    v=(round(v*100))/100;
    for kk=1:length(Kfree)
        xk=3+.6+(kk-1)*.5;
        if BQ==1;
        text(xk, .65,['+ x_',num2str(Kfree(kk)),' '],'color','r')
        else
            text(xk, 1,['+ x_',num2str(Kfree(kk)),' '],'color','r')
        end;
        disp(['v-vector associated with the free variable
x',num2str(Kfree(kk))])
        disp(v(:,kk))
```

```
        for k=1:n
            if BQ==1
            text(xk+.35,.75-(k-1)*.2,[num2str(double(v(k,kk)))],'color','b')
            else
              text(xk+.35,1.25-(k-
1)*.2,[num2str(double(v(k,kk)))],'color','b')
            end;
        end;
    end;

end;
hold on
xr=0:.5:4;
plot(xr,2.2*ones(length(xr),1),'--k','linewidth',1.2)
plot(xr,1.55*ones(length(xr),1),'--k','linewidth',1.2)
if BP==1 && BQ==0
 %   WW=rref(sym([A,b]));
  AP=sym(A)*pp'-sym(b);
    text(-0.75,0.1,'Verification: A*p-b must be NULL:  A*p-b
\Rightarrow','color','b')
    text(2.5,0.0,num2str(double(AP)),'color','r','fontsize',8)
    plot(xr,0.25*ones(length(xr),1),'--k','linewidth',1.2)
elseif BP==0 && BQ==0
    AP=sym(A)*sym(p')-sym(b);
      text(-0.75,0.05,{'Verification:    A*p-b \Rightarrow';'A*p-b must be
NULL'},...
          'color','b')
    text(1.5,0.0,num2str(double(AP)),'color','r','fontsize',8)
    CV=sym(A)*sym(v);
    text(2.5,0.05,{'Verification:    A*v \Rightarrow';'A*v must be NULL'},...
          'color','b')
       text(4.5,0.0,num2str(double(CV)),'color','r','fontsize',8)
          plot(xr,0.25*ones(length(xr),1),'--k','linewidth',1.2)
else
      plot(xr,1.08*ones(length(xr),1),'--k','linewidth',1.2)
%
end;

if BP==0 && BQ==0
  text(3.6,1.85,{'Null Space'; '    [v]'},'color','b')
  text(4.7,1.85,num2str(double(v)),'color','r')
  disp('Null space v')
  disp(v)
elseif BP==0 && BQ==1
     text(3.5,1.3,{'Null Space'; '  [v]'},'color','b')
  text(4.6,1.3,num2str(double(v)),'color','r')
  disp('Null space v')
  disp(v)

end;

end
```

0

```
   -0.8333
         0
    0.3333


ans =

  1/3
 -2/3
    0
  1/3

------------------------
```

Solution of a System of Equations

Augmented Matrix [A b]

2	-4	0	2	4
0	0	2	6	2

RREF ([A b])

1	-2	0	1	2
0	0	1	3	1

No. of Columns of A: n = 4

rank(A)= rank([A b])= 2

system consistent: No PIVOT in the last column of RREF

Identify free/basic var

Null space [V]

x_1 basic variable

x_3 basic variable

x_2 free variable

x_4 free variable

Null Space [v]

2	-1
1	0
0	-3
0	1

rank(A)= rank([A,b]) < n

Solution vector: $[x]=[p]+\Sigma_{[k]} x_{[k]}{}^* v_{[k]} \Rightarrow$

with k= [2 4]

$x_1= 2$ 2   -1

$x_2= 0$ $+ x_2$ 1 $+ x_4$ 0

$x_3= 1$ 0   -3

$x_4= 0$ 0   1

Verification: A*p-b ⇒

A*p-b must be NULL

0

0

Verification: A*v ⇒

A*v must be NULL

0 0

0 0

Published with MATLAB® R2016a