

```

*****
% program name: calcs7.m
% purpose: truss analysis
% input: truss geometry and loading
% output: member forces
% programmer:
% date:
%
% Note: this example covers a m x n block independent matrix
*****

function calcs7()
    clear

% panel points
a=[0,0]; b=[10,5]; c=[13,-6]; d=[10,-12]; e=[6,-4];

% members
% membernum = [1;2;3;4;5;6;7;8;9;10;11;12;13;14];
member = [' ab ';' ae ';' ...';
           ' ba ';' bc ';' be ';' ...';
           ' cb ';' cd ';' ce ';' ...';
           ' dc ';' de ';' ...';
           ' ea ';' eb ';' ec ';' ed '];

% geometry calculations
% abx=b(1,1)-b(1,1); aby=b(1,2)-a(1,2)
% aex=e(1,1)-a(1,1); aey=e(1,2)-a(1,2)
% bcx=c(1,1)-b(1,1); bcy=c(1,2)-b(1,2)
% bex=e(1,1)-b(1,1); bey=e(1,2)-b(1,2)
% cdx=d(1,1)-c(1,1); cdy=d(1,2)-c(1,2)
% cex=e(1,1)-c(1,1); cey=e(1,2)-c(1,2)
% dex=e(1,1)-d(1,1); dey=e(1,2)-d(1,2)

% direction cosine data for A matrix (hand entered)
eabx = 10/sqrt(125); eaex = 6/sqrt(52);
eaby = 5/sqrt(125); eaey = -4/sqrt(52);
ebax = -10/sqrt(125); ebcx = 3/sqrt(130); ebex = -4/sqrt(97);
ebay = -5/sqrt(125); ebcy = -11/sqrt(130); ebey = -9/sqrt(97);
ecbx = -3/sqrt(130); ecdx = -3/sqrt(45); ecex = -7/sqrt(53);
ecby = +11/sqrt(130); ecdy = -6/sqrt(45); ecey = 2/sqrt(53);
edcx = 3/sqrt(45); edex = -4/sqrt(45);
edcy = 6/sqrt(45); edey = 8/sqrt(45);
eeax = -6/sqrt(52); eebx = 4/sqrt(97); eeex = 7/sqrt(53); eedx = 4/sqrt(45);
eeay = +4/sqrt(52); eebx = 9/sqrt(97); eeey = -2/sqrt(53); eedy = -8/sqrt(45);

% loads and reactions vector
B = [ 0 ; 150 ; 0 ; 150 ; 0 ; 150 ; 0 ; 150-337.5 ; 0 ; 150-412.5 ];

% initialize unknown vector
X = [ 0 ; 0 ; 0 ; 0 ; 0 ; 0 ; 0 ; 0 ; 0 ; 0 ; 0 ; 0 ];

% direction cosine matrix
A = [ eabx eaex 0 0 0 0 0 0 0 0 0 0 0 0 0 ; ...
       eaby eaey 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ; ...
       0 0 ebax ebcx ebex 0 0 0 0 0 0 0 0 0 0 0 ; ...
       0 0 ebay ebcy ebey 0 0 0 0 0 0 0 0 0 0 0 0 ; ...
       0 0 0 0 ecbx ecdx ecex 0 0 0 0 0 0 0 0 0 ; ...
       0 0 0 0 ecby ecdy ecey 0 0 0 0 0 0 0 0 0 ; ...
       0 0 0 0 0 0 0 edcx edex 0 0 0 0 0 0 0 0 ; ...
       0 0 0 0 0 0 0 edcy edey 0 0 0 0 0 0 0 0 ; ...
       0 0 0 0 0 0 0 0 0 0 eeax eebx eeex eedx ; ...
       0 0 0 0 0 0 0 0 0 0 eeay eebx eeey eedy ];

% solve for forces in members
X = A\B;

```

```

% output member forces
    disp(' ')
    disp(' member force')
    disp([member(:,1:7) num2str(X)])
    disp(' ')

% plots
    trusslayout(a,b,c,d,e)

% truss layout function
    function [] = trusslayout(a,b,c,d,e)
        % connectivity
        % ab; ae; bc; be; cd; ce; de;

        abx=linspace(a(1,1),b(1,1),2);
        aby=linspace(a(1,2),b(1,2),2);

        aex=linspace(a(1,1),e(1,1),2);
        aey=linspace(a(1,2),e(1,2),2);

        bcx=linspace(b(1,1),c(1,1),2);
        bcy=linspace(b(1,2),c(1,2),2);

        bex=linspace(b(1,1),e(1,1),2);
        bey=linspace(b(1,2),e(1,2),2);

        cdx=linspace(c(1,1),d(1,1),2);
        cdy=linspace(c(1,2),d(1,2),2);

        cex=linspace(c(1,1),e(1,1),2);
        cey=linspace(c(1,2),e(1,2),2);

        dex=linspace(d(1,1),e(1,1),2);
        dey=linspace(d(1,2),e(1,2),2);

        plot(abx,aby,aex,aey,bcx,bcy,bex,bey,cdx,cdy,cex,cey,dex,dey)
        axis equal
        text(a(1,1),a(1,2),'A');text(b(1,1),b(1,2),'B');
        text(c(1,1),c(1,2),'C');text(d(1,1),d(1,2),'D');text(e(1,1),e(1,2),'E')

```