```
%*************************************************************
%  program name:  calcs7reduc.m
%  purpose:       truss analysis
%  input:         truss geometry and loading
%  output:        member forces
%  programmer:
%  date:
%
% Note: this example covers a m x n block matrix
%*************************************************************

function calcs7reduc()
   clear

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

%  member labels
   member = ['  ab   ';'  ae   '; ...
             '  bc   ';'  be   '; ...
             '  cd   ';'  ce   '; ...
             '  de   '];

%  direction cosine calculations for A matrix (generated via points)
   abx=b(1)-a(1);  aby=b(2)-a(2);  ab = sqrt(abx^2 + aby^2);
   eabx = abx/ab;  eaby = aby/ab;  ebax = -eabx;  ebay = -eaby;
   aex=e(1)-a(1);  aey=e(2)-a(2);  ae = sqrt(aex^2 + aey^2);
   eaex = aex/ae;  eaey = aey/ae;  eeax = -eaex;  eeay = -eaey;
   bcx=c(1)-b(1);  bcy=c(2)-b(2);  bc = sqrt(bcx^2 + bcy^2);
   ebcx = bcx/bc;  ebcy = bcy/bc;  ecbx = -ebcx;  ecby = -ebcy;
   bex=e(1)-b(1);  bey=e(2)-b(2);  be = sqrt(bex^2 + bey^2);
   ebex = bex/be;  ebey = bey/be;  eebx = -ebex;  eeby = -ebey;
   cdx=d(1)-c(1);  cdy=d(2)-c(2);  cd = sqrt(cdx^2 + cdy^2);
   ecdx = cdx/cd;  ecdy = cdy/cd;  edcx = -ecdx;  edcy = -ecdy;
   cex=e(1)-c(1);  cey=e(2)-c(2);  ce = sqrt(cex^2 + cey^2);
   ecex = cex/ce;  ecey = cey/ce;  eecx = -ecex;  eecy = -ecey;
   dex=e(1)-d(1);  dey=e(2)-d(2);  de = sqrt(dex^2 + dey^2);
   edex = dex/de;  edey = dey/de;  eedx = -edex;  eedy = -edey;

%   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(80);
%   edcy =   6/sqrt(45);   edey =   8/sqrt(80);
%   eeax =  -6/sqrt(52);   eebx =   4/sqrt(97);  eecx =  7/sqrt(53); eedx =  4/sqrt(80);
%   eeay =  +4/sqrt(52);   eeby =   9/sqrt(97);  eecy = -2/sqrt(53); eedy = -8/sqrt(80);

%  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 ];

% direction cosine matrix

    A = [ eabx eaex    0    0    0    0    0    ; ...
          eaby eaey    0    0    0    0    0    ; ...
          ebax   0   ebcx ebex   0    0    0    ; ...
          ebay   0   ebcy ebey   0    0    0    ; ...
            0    0   ecbx   0  ecdx ecex   0    ; ...
            0    0   ecby   0  ecdy ecey   0    ; ...
            0    0     0    0  edcx   0  edex   ; ...
            0    0     0    0  edcy   0  edey   ; ...
            0  eeax    0  eebx   0  eecx eedx   ; ...
```

```
            0  eeay    0  eeby   0  eecy 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),b(1),2);
      aby=linspace(a(2),b(2),2);

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

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

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

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

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

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


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