This code solves the example of 2D truss structure solved in class

```
%%% 2D truss structure %%%
% Nodal coordinates
%          xi  yi
nodes = [0  0;  % node 1
          1  0;  % node 2    we assume a nominal distance a=1 (**)
          0 -1]; % node 3
n = size(nodes,1);  % number of nodes
% Elements
%         i  j  EA      we assume a nominal EAo=1 (**)
elem = [1 2  1;   % element 1
        3 2  1];  % element 2 (the elements have same EA, not same k!)
m = size(elem,1);   % number of elements
% Stiffness matrix
K = zeros(2*n,2*n);
% vector of stiffnesses, angles of inclination, and stiffness matrix;
for q = 1:m
   i = elem(q,1); xi = nodes(i,1); yi = nodes(i,2); % extracting the coordinates of the nodes of pertinence
   j = elem(q,2); xj = nodes(j,1); yj = nodes(j,2);
   L = sqrt((xj-xi)^2+(yj-yi)^2);  Lv(q)=L;
   kq = elem(q,3)/L;              % stiffness
   tht = atan((yj-yi)/(xj-xi));       % angle
   Ne = kq*[-cos(tht);-sin(tht);cos(tht);sin(tht)];
   %-- alternative path
   qv  = ([xj;yj]-[xi;yi])/norm([xj;yj]-[xi;yi]);
   Ne = (kq*[-qv;qv]);
   %--
   Nev(q,:) = Ne;
   Ke = 1/kq*Ne*Ne';
   edofs = [2*i-1,2*i,2*j-1,2*j];
   K(edofs,edofs) = K(edofs,edofs)+Ke;
end
% Boundary conditions
fix_dofs = [1 2 5 6];        % dofs constrained to be 0
free_dofs = setdiff([1:2*n],fix_dofs);  % the total number of dofs is 2*n (2 per node); this function curtails the dofs that are not
%                                          constrained
% Loads
% To respect the condition of linear elasticity we need to apply a load that is small enough to leave a strain no bigger than 1%
% (**)    If you do some calculations you will find that the nominal strain is eo = Po/EAo, where Po is the nominal load;
%         Let us assume eo = 1% = 0.01; Hence the nominal load becomes Po = 0.01 EAo
Po = 0.01;
f = [0 0 0 -Po 0 0];
% Solution
u = zeros(2*n,1);   % the function 'zeros' gives a matrix by defauls, if one of the sizes is 1, that gives a tensor
u(free_dofs) = f(free_dofs)/K(free_dofs,free_dofs);
% Plot results
figure(1)
hold on
grid on
grid minor
figure(2)
hold on
grid on
grid minor
for q =1:m
   i = elem(q,1); j = elem(q,2);
   xi = nodes(i,1); yi = nodes(i,2);    % undeformed configuration
   xj = nodes(j,1); yj = nodes(j,2);
   xid = xi+u(2*i-1); yid = yi+u(2*i);  % deformed configuration
   xjd = xj+u(2*j-1); yjd = yj+u(2*j);
   figure(1)
   %    undeformed;     deformed;
```

```
    plot([xi,xj],[yi,yj],'k-o',[xid,xjd],[yid,yjd],'b-o')
    figure(2)
    Neq = Nev(q,:);
    edofs = [2*i-1,2*i,2*j-1,2*j];
    Nq = Neq*u(edofs);
    Nv(q) = Nq;
    if Nq > 0
        plot([xi,xj],[yi,yj],'r','LineWidth',Nq*400)      % If the axial force is positive, i.e. it creates tension, then the color is red;
                                                          % The thickness of the line is proportional to the magnitude of the force
    elseif Nq < 0
        plot([xi,xj],[yi,yj],'b','LineWidth',abs(Nq)*400) % If the axial force is negative, the color of the line is blue;
    end
end
figure(1)
title('Structural displacements','fontsize',15)
legend({'Udeformed cofiguration','Deformed cofiguration'},'Location','southeast')
figure(2)
title('Diagram of axial forces','fontsize',15);
```

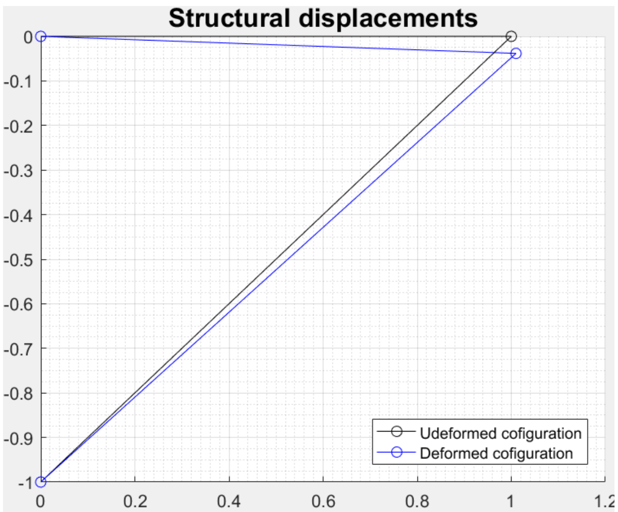

Figure 1

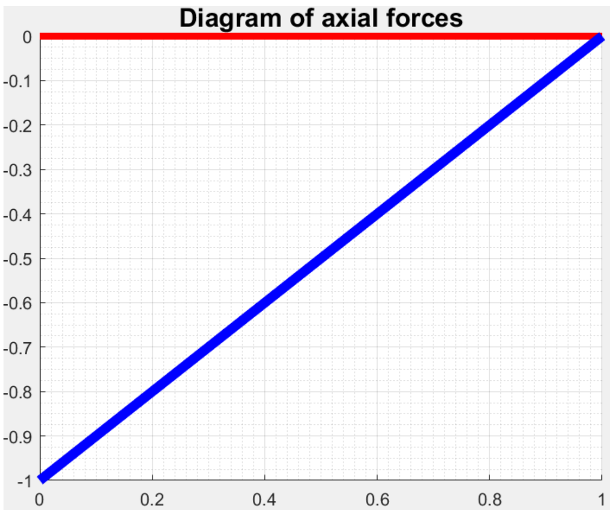

Figure 2