1 APiE Excercise: FEM for Trusses (Hint)

This is a quick overview of the steps for solving Exercise 3. As usual, there are different ways to implement the code - some more and some less efficient. For extremely large networks (1000 x 1000 or more) efficient solution techniques also matter and involve using sparse matrices, preconditioners and iterative solvers such as conjugate gradient etc. (see script/lecture for more details).

General Instructions:

- A mass-spring network is given, which is similar to a network of bars in compression/tension only i.e. a truss. Each node in the truss can move in both x,y directions (except nodes that are fixed at the bottom).
- As shown in the script (and lecture) you can write the FEM stiffness matrix for each bar/spring in the network. Given case is simple since there are only two orientations (0, 90) in the network. We STRONGLY encourage you to write a -function- which returns an element stiffness matrix (and force vector) and not compute matrix for each member individually. Thus, in Matlab you can have a function as follows,
  function [Ke, Fe]= esf_2d(E,theta)

  —compute element stiffness matrix—
  —compute element force vector—

  return Ke, Fe;

  end

  You may want to code it for arbitrary orientation (θ) and stiffness E or keep matters simple for now.
- Call the above function from the main program and assemble the element matrices in a global (big) matrix and apply boundary conditions and solve.

Algorithm

Step 1: Create a global numbering for each node in the network, store node locations (x, y) in a matrix.
Step 2: Create a connectivity array for all the elements in the network for e.g,
Elem(1,1) = 1;
Elem(1,2) = 7; %Element no. 1 is connected to global node number 1 and 7%
etc. ...
Step 3: Initialize element stiffness matrix([Ke]) and element force vector ({Fe}) to zero.
Step 4: Initialize a global stiffness matrix ([K]) and global force vector ({F}). Determine the size based on total number of nodes $\times$ 2 .
Step 5: Loop over all the elements and compute element stiffness matrix ([Ke, Fe]=esf_2d(200,90) etc.) and add [Ke] and {Fe} to [K] and {F} based on connectivity array.
Step 6: Apply boundary conditions i.e. remove rows and columns corresponding to the nodes which are fixed, this will reduce the size of your global stiffness matrix.
Step 7: Solve, {U}= [K]\{F} .

If you have any further questions please email me at s.srivastava@utwente.nl

Succes!