MAE 404/503 Finite Elements in Engineering
Programming assignment #2

For this project you will work in groups of 3, however, all students must submit their code. It is okay if all codes submitted by a group are the same.

Write a MATLAB function that generates the **connectivity matrix** and **global stiffness matrix** for the elastic deformation of a composite bar like the one shown in Figure 1. Each section of the bar (e=1,2,3,4…) should be modeled as a single 2-node element. All sections of the bar are made from steel with properties (E = 200 GPa and ρ = 8,050 kg/m^3). The code must work for any number of sections, not just ne = 4 as shown in Figure 1. Assume all deformations and loadings are one-dimensional (i.e. along the y-axis).

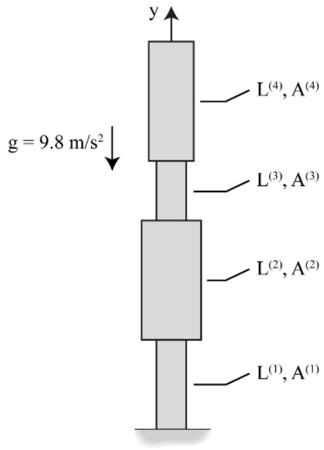

***Figure 1** Composite bar. Elements are numbered from bottom to top.*

Instructions for programming and assignment submission:

- Submit a single file (MATLAB code). The file name must be in the format “asurite_hw2.m”. Note that the separator is an underscore and you must substitute your ASU login.
- The file **must** define a function of the same name as the file name (but without the “.m”) , e.g.,

```
function [conn, K] = asurite_hw2(element_lengths, element_areas)
    % Specify default values for testing.
    if nargin == 0
        element_lengths = [10;20;30;40];  % (in mm)
        element_areas = pi*[2;4;2;4].^2;  % (in mm^2)
    end
    % Code goes here to construct connectivity and stiffness matrices.
end
```

- The input arguments, `element_lengths`, and `element_areas` are both ne×1 matrices and specify the dimensions of each element, specified from bottom to top. The values of the element lengths and areas will be specified in mm and mm^2, respectively.
- The function must return two variables. The first must be the 2×ne connectivity matrix, that gives the element nodes for each element in the mesh. The second must be the nn×nn global stiffness matrix (**with units in N/m**).
- Your code must give the correct outputs for any reasonable values of `element_lengths`, and `element_areas`. If your code only works for one specific case, you will get zero credit.
- Efficient MATLAB programming: Use array-based indexing to scatter the stiffness matrix.

Your submission will be graded electronically. Failure to comply with the above instructions will result in zero credit.

Submit your assignment to http://sparky.fulton.asu.edu/fem/

Due date: Jan 18 at 11:59 pm.

You may submit as many times as you like, however only your most recent submission will count for you grade.