UNIVERSITY OF CALIFORNIA AT BERKELEY
Department of Civil and Environmental Engineering
Structural Engineering, Mechanics and Materials

CE 130N, Spring 2009
Prof. S. Govindjee and Dr. T. Koyama
Lab 5

Mechanics of Structures (CE130N)
Lab 5

1 Objective

The objective of this lab is to implement the techniques introduced to analyze statically determinate and indeterminate truss and frame structures. When the size of the structure is small, i.e., the number of unknown quantities is small, one can solve the problems by hand, but as the size of the problem increases, the complexity of hand solutions grows exponentially. The systematic method introduced allows one to easily analyze large structures.

In this lab you will be asked to write some functions and procedures which will allow you to complete a program which analyzes general truss and frame structures. The steps you will follow can be roughly organized as,

1. **Truss structures**
   - Complete a function which applies boundary conditions,
   - Solve for the displacements and forces of truss structures,
   - Investigate the stability of a structure by analyzing $\mathbf{K}_{ff}$, the stiffness matrix corresponding to the free degrees-of-freedom.

2 Truss structures

2.1 Governing equation

Let us define the following quantities,

$ndim$: Number of dimensions, for 2D(2), for 3D(3)
nnp : Number of node points
nel : Number of elements (bars)

The governing equations for a truss structure are the following,

- Equilibrium

$$\mathbf{F} \quad = \quad \mathbf{A}^T\mathbf{R} \tag{1}$$

Here $\mathbf{F} \in \mathbb{R}^{ndim \cdot nnp \times 1}$ is the vector of nodal forces, $\mathbf{R} \in \mathbb{R}^{nel \times 1}$ is the vector of bar forces, and $\mathbf{A} \in \mathbb{R}^{nel \times ndim \cdot nnp}$ is the compatibility matrix.

- Kinematics

$$\begin{aligned} \boldsymbol{\varepsilon} &= [1/L]\boldsymbol{\Delta}\mathbf{L}, & (2) \\ \boldsymbol{\Delta}\mathbf{L} &= \mathbf{A}\mathbf{u}, & (3) \end{aligned}$$

combined yield,

$$\boldsymbol{\varepsilon} \quad = \quad [1/L]\mathbf{A}\mathbf{u}. \tag{4}$$

Here $\mathbf{u} \in \mathbb{R}^{ndim \cdot nnp \times 1}$ is the vector of nodal displacements, $\boldsymbol{\varepsilon} \in \mathbb{R}^{nel \times 1}$ is the vector of bar strains, $\boldsymbol{\Delta}\mathbf{L} \in \mathbb{R}^{nel \times 1}$ is the vector of bar deformations, and $[1/L] \in \mathbb{R}^{nel \times nel}$ is the diagonal matrix relating $\boldsymbol{\Delta}\mathbf{L}$ and $\boldsymbol{\varepsilon}$.

- Constitutive relation

$$\boldsymbol{\sigma} \quad = \quad [E]\boldsymbol{\varepsilon} \tag{5}$$

Here $\boldsymbol{\sigma} \in \mathbb{R}^{nel \times 1}$ is the vector of bar stresses and $[E] \in \mathbb{R}^{nel \times nel}$ is the diagonal matrix of Young's moduli.

- Resultant definition

$$\mathbf{R} \quad = \quad [A]\boldsymbol{\sigma} \tag{6}$$

Here $[A] \in \mathbb{R}^{nel \times nel}$ is the diagonal matrix of the area of the bar cross sections.

In the lecture these four relations have been combined to obtain a single equation representing equilibrium in terms of the vector of displacements $\mathbf{u}$,

$$\boxed{\mathbf{F} = \mathbf{K}\mathbf{u},} \tag{7}$$

where,

$$\begin{aligned} \mathbf{K} &:= \mathbf{A}^T[A][E][1/L]\mathbf{A}, \\ &= \mathbf{A}^T[AE/L]\mathbf{A}\ . \end{aligned}$$

and $[AE/L]$ is a diagonal matrix representing the axial stiffness of the bars,

$$\mathbf{R} \;=\; [AE/L]\boldsymbol{\Delta}\mathbf{L}\,.$$

By defining,

$$\begin{aligned}\mathbf{K}_s &:= [AE/L]\,,\\ \mathbf{V} &:= \boldsymbol{\Delta}\mathbf{L}\,,\end{aligned}$$

one can restate the governing equations in an alternate form,

- Equilibrium

$$\mathbf{F} \;=\; \mathbf{A}^T\mathbf{R}$$

- Kinematics

$$\mathbf{V} \;=\; \mathbf{A}\mathbf{u},$$

- Effective constitutive relation

$$\mathbf{R} \;=\; \mathbf{K}_s\mathbf{V}\,.$$

2.2 Boundary condition

To solve for the displacement vector $\mathbf{u}$, which has a total of $ndim \times nnp$ unknowns (degrees-of-freedom), one also requires boundary conditions. This situation is identical to the setup for the previously introduced mechanical problems governed by differential equations. Let us denote the ith entry of $\mathbf{F}$ by F_i and the ith entry of $\mathbf{u}$ by u_i. Recall that where ever a displacement (degree-of-freedom) u_i is specified or known in a certain direction, there is an unknown reaction force F_i in the corresponding direction, and where ever a force F_i is specified or known in a certain direction, the node is free to move in the corresponding direction with unknown displacement u_i. It is clear that in general, one cannot simultaneously specify the force and displacement for a degree-of-freedom.

Thus one can completely separate the set of degrees-of-freedom of the nodes (a total of $ndim \times nnp$) into two disjoint groups,

- id_f: The degrees-of-freedom which are (F)REE TO MOVE.
- id_d: The degrees-of-freedom corresponding to (D)ISPLACEMENTS KNOWN.

We will put a subscript f to specify that a degree-of-freedom is in the set id_f, and a subscript d to specify that the degree-of-freedom is in the set id_d. Using these two disjoint sets of indices, one can partition the vectors $\mathbf{F}$ and $\mathbf{u}$ into,

$$\begin{aligned}\mathbf{F} &:= \begin{bmatrix}\mathbf{F}_f\\ \mathbf{F}_d\end{bmatrix},\\ \mathbf{u} &:= \begin{bmatrix}\mathbf{u}_f\\ \mathbf{u}_d\end{bmatrix}\end{aligned}$$

where the vectors $\mathbf{F}_f$ and $\mathbf{u}_f$ contain the degrees-of-freedom corresponding to (F)REE TO MOVE degrees-of-freedom and $\mathbf{F}_d$ and $\mathbf{u}_d$ contain the degrees of freedom corresponding to (D)ISPLACEMENTS KNOWN. From this nomenclature it is clear that $\mathbf{u}_f$, the displacement at the applied forces, and $\mathbf{F}_d$, the reaction forces, are the unknowns.

Let us also partition the stiffness matrix,

$$\mathbf{K} \;:=\; \begin{bmatrix}\mathbf{K}_{ff} & \mathbf{K}_{fd}\\ \mathbf{K}_{df} & \mathbf{K}_{dd}\end{bmatrix},$$

such that the Eqn. (7), which must be solved, is,

$$\begin{bmatrix} \mathbf{F}_f \\ \mathbf{F}_d \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fd} \\ \mathbf{K}_{df} & \mathbf{K}_{dd} \end{bmatrix} \begin{bmatrix} \mathbf{u}_f \\ \mathbf{u}_d \end{bmatrix} .$$

Employing the fact that $\mathbf{u}_d$ is known, we can solve for the $\mathbf{u}_f$ and $\mathbf{F}_d$ in the following two steps,

1. Solve for the $\mathbf{u}_f$. Define,

   $$\mathbf{r}_f := \mathbf{F}_f - \mathbf{K}_{fd}\mathbf{u}_d = \mathbf{K}_{ff}\mathbf{u}_f \ ,$$

   and solve,

   $$\boxed{\mathbf{u}_f = \mathbf{K}_{ff}^{-1}\mathbf{r}_f \ .} \tag{8}$$

2. Evaluate $\mathbf{F}_d$,

   $$\mathbf{F}_d = \mathbf{K}_{df}\mathbf{u}_f + \mathbf{K}_{dd}\mathbf{u}_d \ ,$$

   or by,

   $$\mathbf{F} = \mathbf{K}\mathbf{u} \ ,$$

   since now $\mathbf{u}$ is solved and a known quantity.

3 Exercise

3.1 Download updates for the *truss* program

1. Download the file `truss_update1.zip` into your `ce130n/programs/` directory and unzip it.
2. Go to the directory `ce130n/programs/truss_update1/exercise` and run the file `init.m`.

YOU MUST RUN THE FILE init.m EVERYTIME YOU START UP MATLAB.

3.2 Construct the entire solution method

1. Complete the procedure
   `ce130n/programs/truss/core/solvemesh.m`
   up to the line of code,

   ```
   % -- COMPLETE PART OF THE FOLLOWING FUNCTION
   % [Kff,Rf]             = applybc(mesh,K,u,F);
   ```

2. Uncomment the line of the code,

   ```
   % -- COMPLETE PART OF THE FOLLOWING FUNCTION
   [Kff,Rf]             = applybc(mesh,K,u,F);
   ```

   and complete the function,
   `ce130n/programs/truss/core/applybc.m`.

   To test the function one must first load the mesh structure for the problem and then run the function,

   ```
   >> mesh_data_truss_example;
   >> solvemesh;
   ```

   Use the data structure of `mesh_data_truss_example.m` to test your function.

3. Complete the section after the line,

   ```
   % -- COMPLETE PART OF THE FOLLOWING FUNCTION
   [Kff,Rf]             = applybc(mesh,K,u,F);
   ```

   Remember to uncomment the last lines,

   ```
   % -- Insert u into mesh structure
   %mesh.u = reshape_vector2data(mesh,u);
   %mesh.f = reshape_vector2data(mesh,F);
   ```

   To test the function one must first load the mesh structure for the problem and then run the function,

   ```
   >> mesh_data_truss_example;
   >> solvemesh;
   >> plotdefo(mesh,1e1);
   ```

   The function `plotdefo` plots the deformed truss magnifying the displacement by the value 0.01. If you see that the displacements are too small to observe then increase this number.

   The correct displacement for the top node is $(0.0038, -0.0010)$.

4. Try running the Howe truss example. The data structure is defined in the code, `ce130n/programs/truss/exercise/mesh_data_howe.m`.

   The correct displacement for the middle node (node 4) is $(0.0, -0.0097)$.

   CHECKPOINT: Show the plot for the deformed structures (Example structure and Howe truss). Explain what the reaction forces, bar forces, and bar stresses are and how you can obtain them.

3.3 Check stability of truss structures

In this exercise you will be asked to check the stability of truss structures by investigating the stiffness matrix corresponding to the free degrees-of-freedom, $\mathbf{K}_{ff}$.

Using the data structures given for 3 truss structures shown in Figures 1-3, check their stability. If there are any modes in which the structure is stable, identify its shape. Once you compute the eigenvalue `d` and eigenvector `v` of a truss structure, you can use the following MATLAB function,

```
>> ploteigenvector(mesh,v,d,scale);
```

to show the shape of the eigenvector. `scale` denotes the magnifying parameter of the displacements.

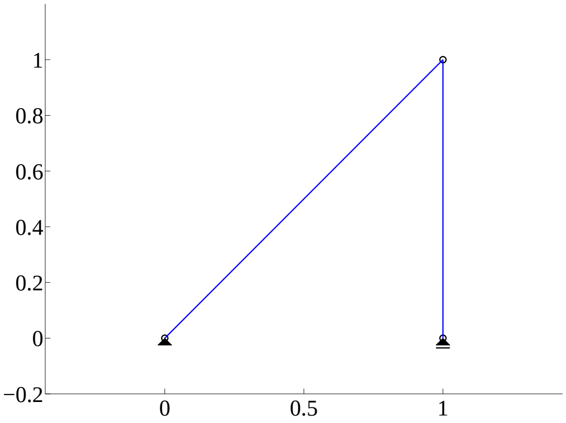

Figure 1: Filename: mesh_data_truss_st1.m

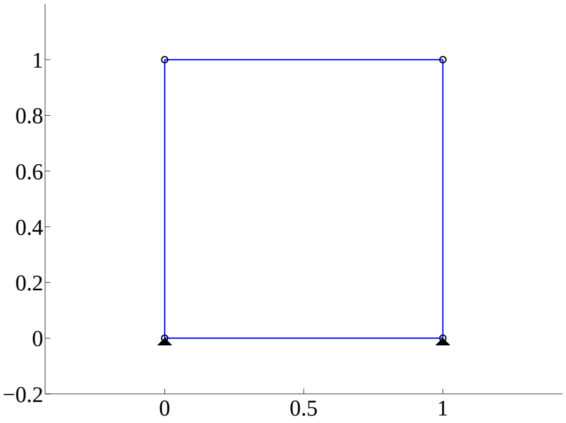

Figure 2: Filename: mesh_data_truss_st2.m

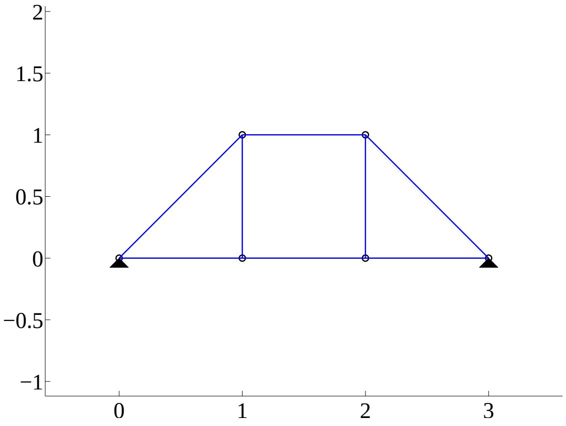

Figure 3: Filename: mesh_data_truss_st3.m

HINT: The eigenvalues and eigenvectors of $\mathbf{K}_{ff}$ can be computed in MATLAB with the command,

```
>> [V,D] = eig(full(Kff));
```

Type `help eig` for details of the function. *YOU MUST RUN THE FILE solvemesh.m BEFORE USING EIG TO SET THE IDF IN THE MESH STRUCTURE.*

CHECKPOINT: Show the plot for the modes of instability. Explain why you suspect these modes are unstable. How can you alter the structure so that they become stable?