The Georgia Institute of Technology
School of Civil and Environmental Engineering
Structural Analysis CEE 4550 - Fall 2019
Project Description

Designed by
Shahrokh Shahi
www.sshahi.com

Introduction

This project aims at developing a program in MATLAB to analyze two- and three-dimensional trusses. Similar to the MATLAB assignments in your homework, you are provided with some components and skeleton files to facilitate your work and unify the development and grading procedure.

In this project, at first, you need to become familiar with the components of this program. Some of these components are incomplete. This document provides the instructions to help you complete these functions and obtain a working package. You also need to submit a report along with your code explaining your steps and findings during developing this program.

This instruction is structured as follows:

Part I – MATLAB Programming

Part II – Report

Part III – Submission and Evaluation

Part I – MATLAB Programming

The skeleton files are available in "**project.zip**". Before explaining anything, please remember the following **important notes**:

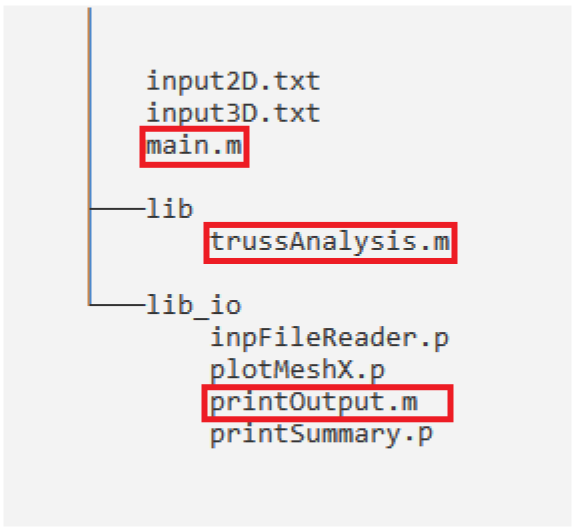
During the development of this program and working on this code:

- **Do NOT** rename the functions (do not change the name of neither functions nor files)
- **Do NOT** rename the predefined variables or the output variables.
- **Do NOT** hard-code any parameter (Your program must be able to handle any given input file)

Following these rules is very critical since your programs will be evaluated against a few benchmark problems by means of an auto-grader program and if you change the name of functions or variables it may break the auto-grader and you will lose credits.

1- The File Structure

First, download the **project.zip** file and extract it before you begin. This folder includes all the components (skeleton files) that you need to complete them to accomplish this project. The following figure illustrates the main file structure:



```
│
│   input2D.txt
│   input3D.txt
│   main.m
│
├───lib
│       trussAnalysis.m
│
└───lib_io
        inpFileReader.p
        plotMeshX.p
        printOutput.m
        printSummary.p
```

- In the root folder (project\), the script m-file "**main.m**" is the file that controls the flow of the program. Moreover, all the input files should be located in the root folder.

- All the computational components that are required to be developed are located in the "lib" folder (project\lib\). Currently, it only includes "**trussAnalysis.m**" which will be called by the main function to analyze a given truss. This file is incomplete and your main task is completing this function. If you need to develop other helper functions, they must be added to this folder, but please remember, that the main analyzer function **MUST BE** **trussAnalysis.m** and you are not allowed to rename this function. Again, if you rename this function or define any other functions to analyze truss structures, our auto-grader cannot evaluate your program and you will lose credit.

- The files located in the "lib_io" folder (project\lib_io\) are the functions provided to ease working on the input/output data, such as interpreting the input files and visualizing the geometry data. You can use these functions as black boxes and do NOT need to modify the contents of this folder, except **printOutput.m** which is needed to be completed to display the results of an analysis.

2- The main file (main.m)

This file is the main m-file controlling the flow of the program. Once you define an input file, you should enter the file name in the main file and run it. The flow of the program in main.m is as follows:

(1) Initialization: clearing the Command Window, clearing all variables, setting the output formats, and adding the required path to the MATLAB operating path. (You should NOT modify this section)

(2) Introducing an "Input File Name" that is the name of a text input file in which you input all the data required to define the problem including the geometry, material properties, loading, boundary conditions, etc. So far, in the MATLAB assignments, you have worked with input files defined with a specific format. Here, the input format is almost the same as the one used in the assignments.

(3) Interpreting the input file: In homework 1 and 2, you worked on reading and interpreting data from input files defining planar trusses. In this project, in addition to 2D (planar) trusses, you need to define and analyze 3D (space) trusses. The format of these input files is explained in the next section. If you follow the instructions for defining input files, the function "**inpFileReader.p**" (located in lib_io) can interpret them and encapsulate all the data in a MATLAB struct variable named "Model".
(A struct or structure array in MATLAB is a data type that groups related data using data containers called fields. Each field can contain any type of data. Access data in a field using dot notation of the form "structName.fieldName")
(→ More information: https://www.mathworks.com/help/matlab/ref/struct.html)
The "Model" struct has a specific structure of fields (variables), which is also explained in the next section.

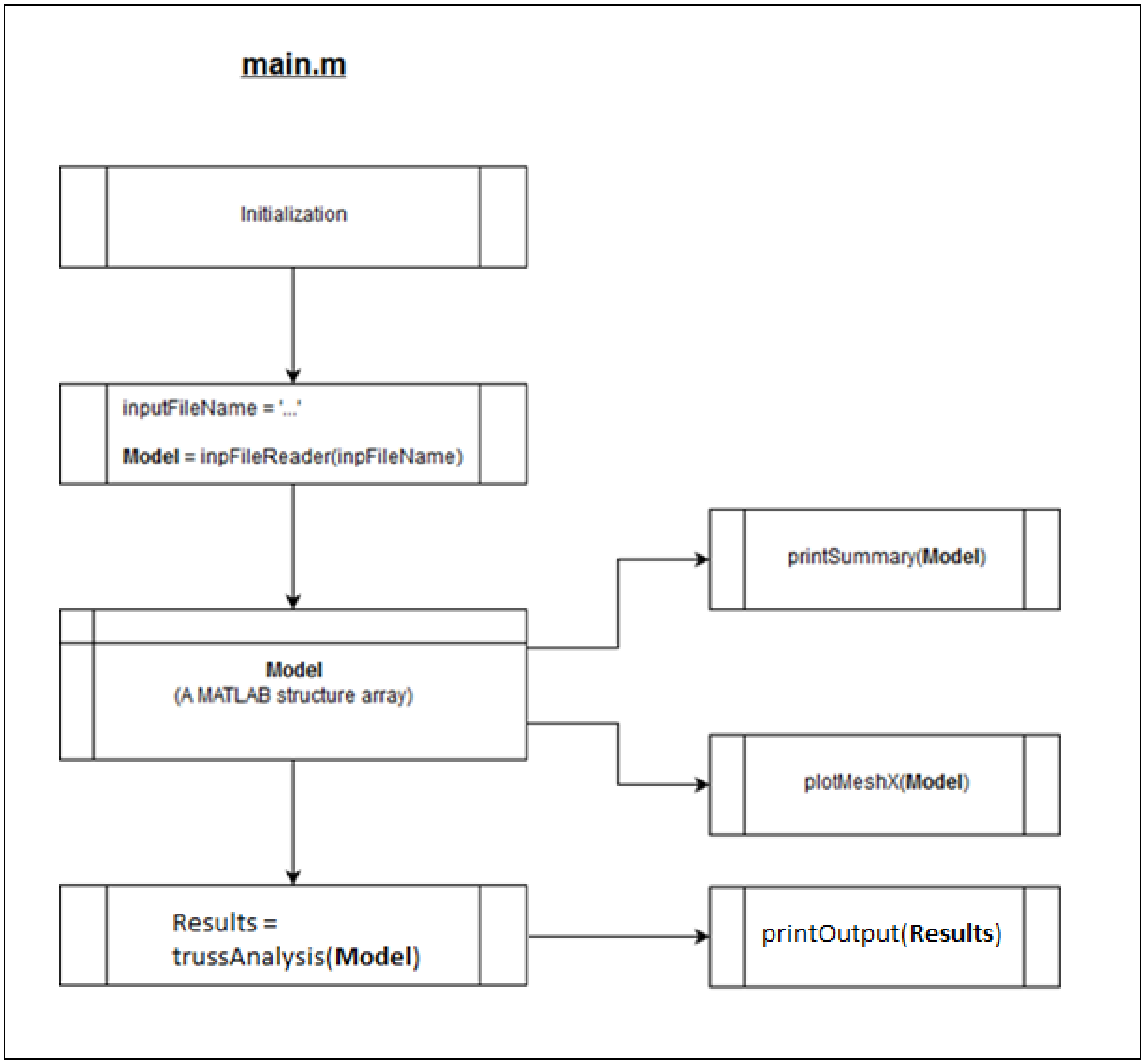
(4) Displaying a summary of the input data: After obtaining the "Model" struct variable, you can use the "**printSummary.m**" function to display a summary of the contents of this structure on the MATLAB Command Window. (This function is also located in the "lib_io" and you do NOT need to modify it)
Moreover, you can check the contents of the Model structure by just typing the name of this variable in Command Window (after running the main.m):

```
>> Model
```

Then you will see a list of fields and their values. You can use dot (.) to access the fields' values. For instance, in the following example, `Model.nNode` will return the number of nodes and `Model.geometry` gives an inner structure named geometry including nodal coordinates and elements connectivity data.

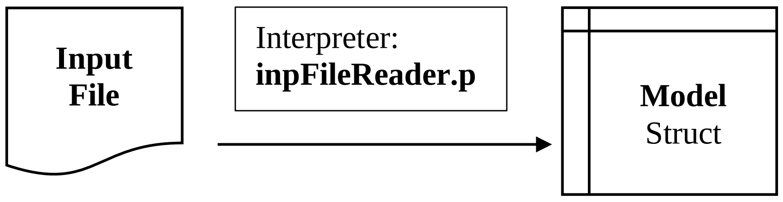
(5) Visualizing: It is very worthwhile to have a tool to visualize the geometry of the structure. In this way, you will have a better sense about the problem and be able to find any problem in your input file much easier. In the next section of "**main.m**", the function, "**plotMeshX.p**" uses the Model structure and generate a descriptive figure demonstrating nodes and elements numbers, boundary conditions and concentrated loads.

(6) The last section of this file is calling the main analyzer function, which is "**trussAnalysis.m**". This function is located in the "lib" folder and the problem data will be passed to this function by the Model struct variable. You need to complete this function to obtain an accomplished package. Then, you can run **main.m** to test your program.



2-1- Defining an Input File

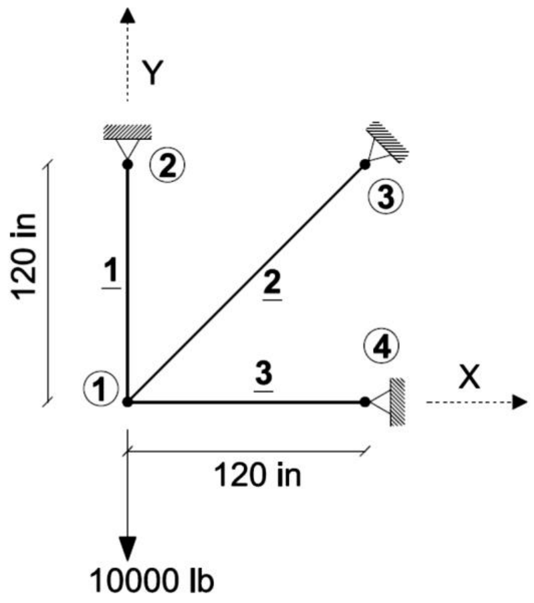
As explained earlier, all the data required to define a problem including geometry, boundary conditions, and loading must be defied in an input file, then it will be interpreted to a MATLAB struct variable, by means of **inpFileReader.m** function. You can use this function as a black box to interpret input files. Your program will eventually be evaluated against a few benchmark problems with the same input file format. Therefore, it is very important to understand and follow the instructions.

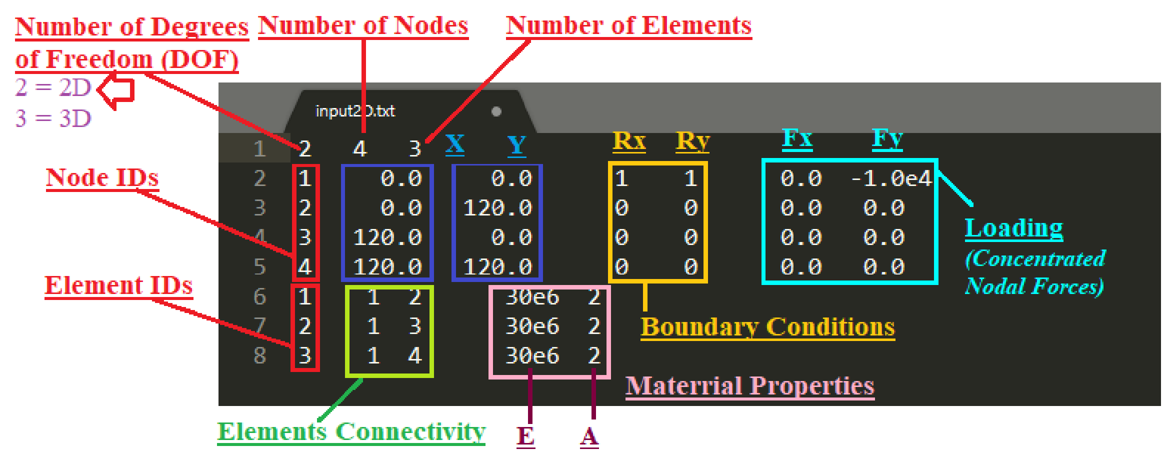


By using the function `Model=inpFileReader("inputFileName.txt")`, all the data will be stored in the Model structure variable.

The acceptable format for defining an input file is demonstrated in the following figures for some 2D and 3D truss structures:

(a) 2D Example:

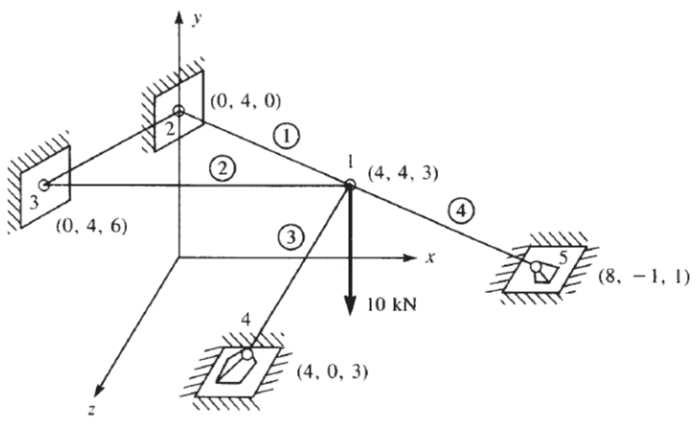


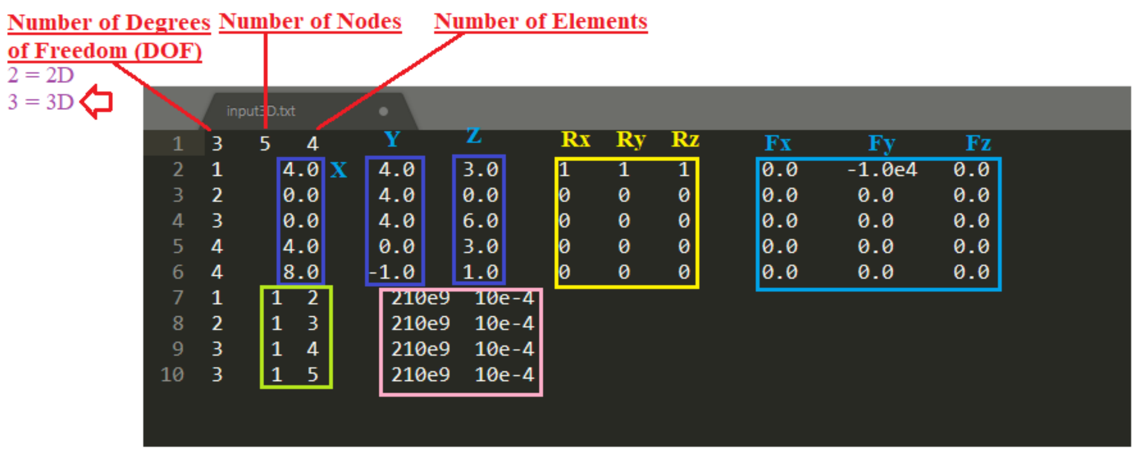


```
2    4    3
1      0.0     0.0     1     1     0.0   -1.0e4
2      0.0   120.0     0     0     0.0    0.0
3    120.0     0.0     0     0     0.0    0.0
4    120.0   120.0     0     0     0.0    0.0
1    1   2            30e6   2
2    1   3            30e6   2
3    1   4            30e6   2
```

➔ The first number nDof is 2

(b) 3D Example:





```
3   5   4
1      4.0    4.0    3.0    1   1   1    0.0   -1.0e4   0.0
2      0.0    4.0    0.0    0   0   0    0.0    0.0     0.0
3      0.0    4.0    6.0    0   0   0    0.0    0.0     0.0
4      4.0    0.0    3.0    0   0   0    0.0    0.0     0.0
4      8.0   -1.0    1.0    0   0   0    0.0    0.0     0.0
1    1  2    210e9   10e-4
2    1  3    210e9   10e-4
3    1  4    210e9   10e-4
3    1  5    210e9   10e-4
```

➔ The first number nDof is 3

2-2- A Brief Review on Variable Names and the "Model" Struct Fields

Once the function inpFilereader.m interprets an input file, the content of the input file will be stored in a MATLAB struct variable named "Model". The following is a hierarchical view of the most important fields in a Model struct:

```
Model:
  |
  +--- info:  information about the model
  +--- nNode: total number of nodes
  +--- nElem: total number of elements
  +--- nDof:  number of degrees-of-freedom per each node
  |
  |
  +--- geometry: an inner structure including geometry data
  |  |
  |  +--- coordinates: node coordinates matrix (nNode x nDof)
  |  +--- elements: elements connectivity (nElem x 2)
  |
  +--- E: elements module of elasticity (nElem x 1)
  +--- A: elements cross-sectional area (nElem x 1)
  |
  +--- F: applied nodal forces (nNode x nDof)
  +--- restrained: boundary conditions [Rx,Ry,Rz] (nNode x nDof)
```

Notes:

- As mentioned earlier, you can access a variable (field) value within a structure by using dot. Therefore, for instance, `Model.A` gives a matrix (vector) including all the cross-sectional area data, which is read by the function "inpFileReader" from the input file and stored in the Model structure.

- The Model structure includes an inner structure (nested) which is **geometry**, and includes the nodal coordinates and elements connectivity data.
  To access those values, you can use dot operator twice, for instance, `Model.geometry.coordinates`

- At this step, you can try to write new input files and then use "inpFileReader", "plotMeshX" and "printSummary" functions to become familiar with their functionality and see the contents of the Model structures.

- You do not need to be worried about Model structure array. It is just an encapsulation of multiple variables. To ease your work, in the first few lines of trussAnalysis.m skeleton file, the required variables are extracted from Model structure array and you can use them as they were defined in previous MATLAB assignments.

3- Accomplishing the Package:

Now, you are ready to start completing the computational functions. The following table illustrates a summary of these functions and their functionalities:

Function	Functionality	Needs to be completed?
main.m	The main script file controlling the flow of the analysis	Yes

trussAnalysis.m	The main analyzer function. You need to implement finite element approach to use the data stored in the Model and analysis s given truss.
	
	`Usage:`
	`>> Results = trussAnalysis(Model)`
		Yes

inpFileReader.p	Reading and interpreting an input file (for both 2D and 3D trusses)
	
	`Usage:`
	`>> Model = inpFileReader(input.txt)`
		No

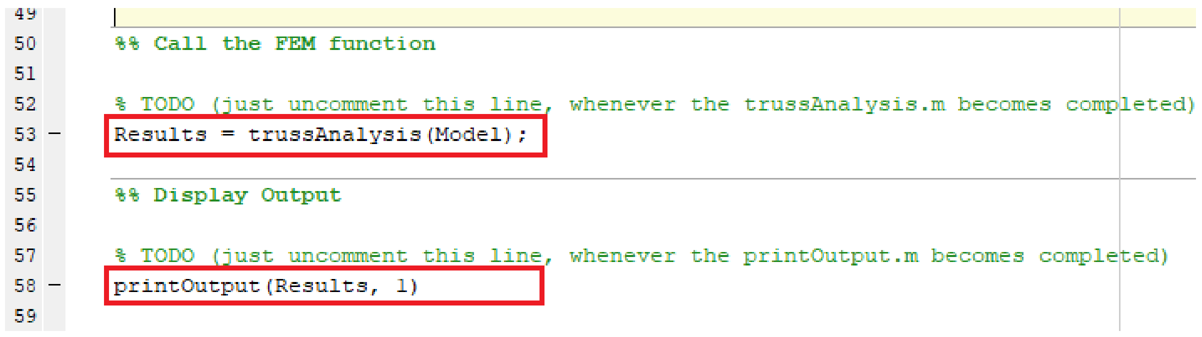
plotMeshX.p	Representing a graphical view of the truss structure stored in a Model structure
	
	`Usage:`
	`>> plotMeshX(Model)`
		No

printSummary.p	Displaying a summary of the input data stored in a Model structure
	
	`Usage:`
	`>> printSummary(Model, fid)`
	
	Where fid = 1, if we want to display data on the screen (Command Window) or fid = the id of an output file opened using **fopen** function
		No

printOutput.m	Displaying a summary of output results stored in the Results structure
	
	`Usage:`
	`>> printOutput(Results, fid)`
	
	Where fid = 1, if we want to display data on the screen (Command Window) or fid = the id of an output file opened using **fopen** function
		Yes

3-1- main.m

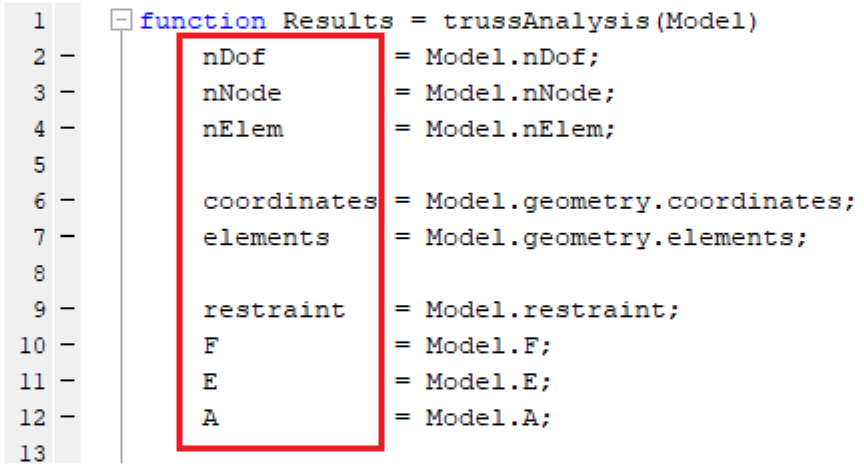
This file is almost complete. You just need to un-comment two last sections when trussAnalysis.m and printOutput.m are completed to run the analysis.



```
%% Call the FEM function

% TODO (just uncomment this line, whenever the trussAnalysis.m becomes completed)
Results = trussAnalysis(Model);

%% Display Output

% TODO (just uncomment this line, whenever the printOutput.m becomes completed)
printOutput(Results, 1)

```

3-2- trussAnalysis.m

This file is the main function that you need to develop, in which you can use the following extracted variables



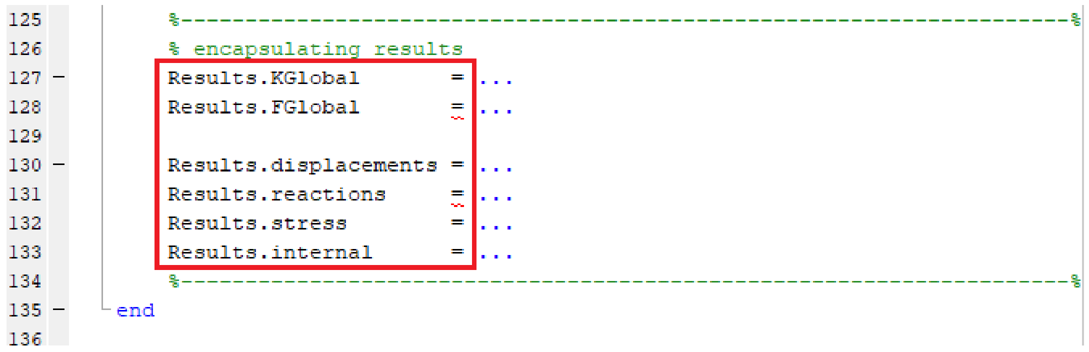
```
function Results = trussAnalysis(Model)
        nDof        = Model.nDof;
        nNode       = Model.nNode;
        nElem       = Model.nElem;

        coordinates = Model.geometry.coordinates;
        elements    = Model.geometry.elements;

        restraint   = Model.restraint;
        F           = Model.F;
        E           = Model.E;
        A           = Model.A;

```

to implement a finite element approach to analyze a given 2D or 3D truss, including

- defining elements stiffness matrices,
- assembling the global stiffness matrix,
- assembling global nodal forces vector,
- forming [K]{u}={f}, imposing boundary condition and solving the system to obtain nodal displacements and reaction forces,
- finally calculating stresses and internal forces.

Note that your program must handle both 2D and 3D trusses; therefore, you may need to use an if-statement or switch-case branching approach to handle these two different cases.

Then, you need to store all the results, including displacements, reaction forces, stresses, and internal forces, in addition to global stiffness matrix and global nodal forces vector in an output structure array named "Results".



```
    %---------------------------------------------------------------------%
    % encapsulating results
    Results.KGlobal       = ...
    Results.FGlobal       = ...

    Results.displacements = ...
    Results.reactions     = ...
    Results.stress        = ...
    Results.internal      = ...
    %---------------------------------------------------------------------%
end
```

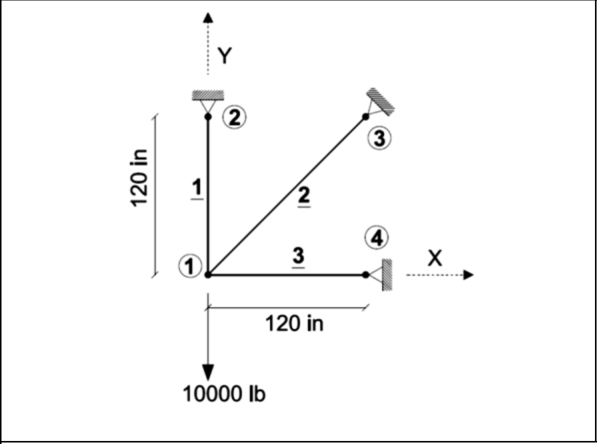
This structure is the output of this function and will be returned to the main file ("main.m"). These outputs **MUST** have the following dimensions:

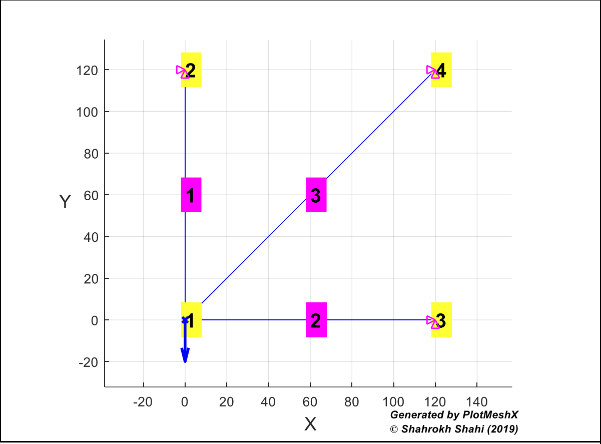
#	output	Dimension	Description
1	`Results.KGlobal`	nNode.nDof × nNode.nDof	The global assembled stiffness matrix
2	`Results.FGlobal`	nNode.nDof × 1	The global assembled nodal load vector
3	`Results.displacements`	nNode × nDof	The output nodal displacements
4	`Results.reactions`	nNode × nDof	The output reaction forces (If a dof is not restraint, then the corresponding entry of reactions matrix will be zero)
5	`Results.stress`	nElem × 1	The output element stresses
6	`Results.internal`	nElem × 1	The output element (internal) axial forces

3-3- printOutput.m

Display all the obtained results (displacements, reaction forces, stresses, and internal forces) in a proper format on the Command Window. For this purpose, you need to complete the printOutput.m function. This function takes the output structure "**Results"** and display the outputs. The following box demonstrates a sample outputs of analyzing the given 2D and 3D trusses displayed on the Command Window. Your outputs should be very similar to the following:

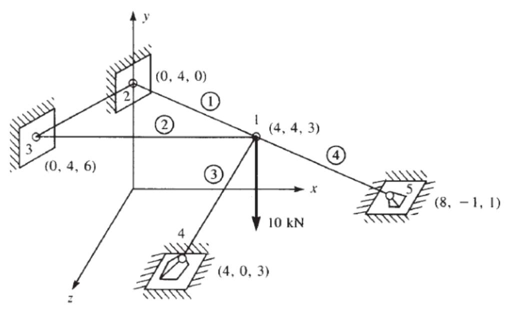
(a) 2D

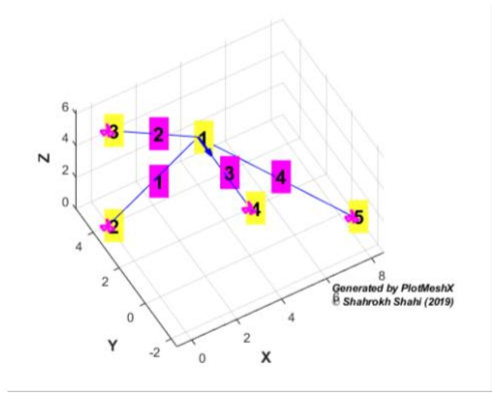




```
============================================================
                    R E S U L T S
============================================================
 Table of Contents:
   (1) Nodal Displacements
   (2) Reaction Forces
   (3) Stresses
   (4) Internal Forces
------------------------------------------------------------



============================================================
(1)          N O D A L   D I S P L A C E M E N T S
============================================================
Node ID             u               v
------------------------------------------------------------
 1            0.0041421356        -0.0158578644
 2            0.0000000000         0.0000000000
 3            0.0000000000         0.0000000000
 4            0.0000000000         0.0000000000
------------------------------------------------------------



============================================================
(2)             R E A C T I O N   F O R C E S
============================================================
Node ID             Rx               Ry
------------------------------------------------------------
 1            0.0000000000         0.0000000000
 2            0.0000000000      7928.9321881345
 3        -2071.0678118655         0.0000000000
 4         2071.0678118655      2071.0678118655
------------------------------------------------------------



============================================================
(3,4) S T R E S S E S  &  I N T E R N A L  F O R C E S
============================================================
Element         Internal (Axial)          Stresses
 ID                  Forces
------------------------------------------------------------
 1                       7928.93219           3964.46609
 2                      -2071.06781          -1035.53391
 3                       2928.93219           1464.46609
============================================================
```

(b) 3D



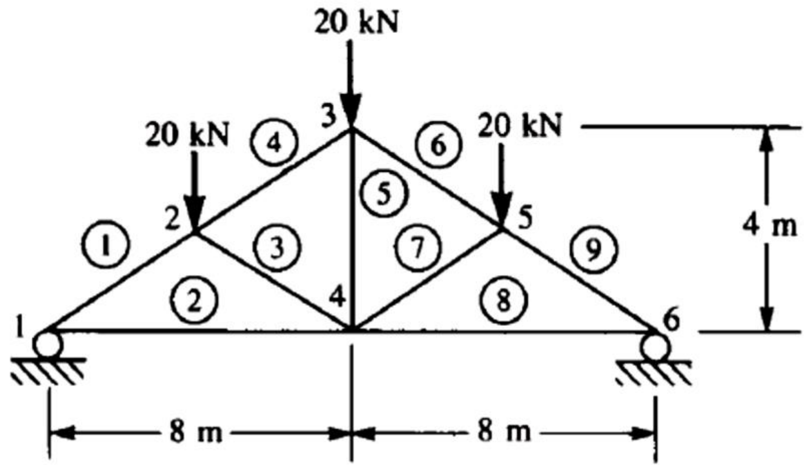


```
==============================================================
                    R E S U L T S
==============================================================
 Table of Contents:
   (1) Nodal Displacements
   (2) Reaction Forces
   (3) Stresses
   (4) Internal Forces
--------------------------------------------------------------


==============================================================
(1)          N O D A L   D I S P L A C E M E N T S
==============================================================
Node ID           u                 v                w
--------------------------------------------------------------
 1      -0.0000302368    -0.0001517731     0.0000268772
 2       0.0000000000     0.0000000000     0.0000000000
 3       0.0000000000     0.0000000000     0.0000000000
 4       0.0000000000     0.0000000000     0.0000000000
 5       0.0000000000     0.0000000000     0.0000000000
--------------------------------------------------------------


==============================================================
(2)             R E A C T I O N   F O R C E S
==============================================================
Node ID            Rx                Ry                Rz
--------------------------------------------------------------
 1         0.00000000       0.00000000        0.00000000
 2       270.92178881       0.00000000      203.19134160
 3      1354.60894403       0.00000000    -1015.95670802
 4         0.00000000    7968.08658396        0.00000000
 5     -1625.53073283    2031.91341604      812.76536642
--------------------------------------------------------------


==============================================================
(3,4) S T R E S S E S  &  I N T E R N A L   F O R C E S
==============================================================
Element         Internal (Axial)          Stresses
 ID                  Forces
--------------------------------------------------------------
 1                         -338.65224         -338652.23601
 2                        -1693.26118        -1693261.18003
 3                        -7968.08658        -7968086.58396
 4                        -2726.09791        -2726097.91360
==============================================================
```

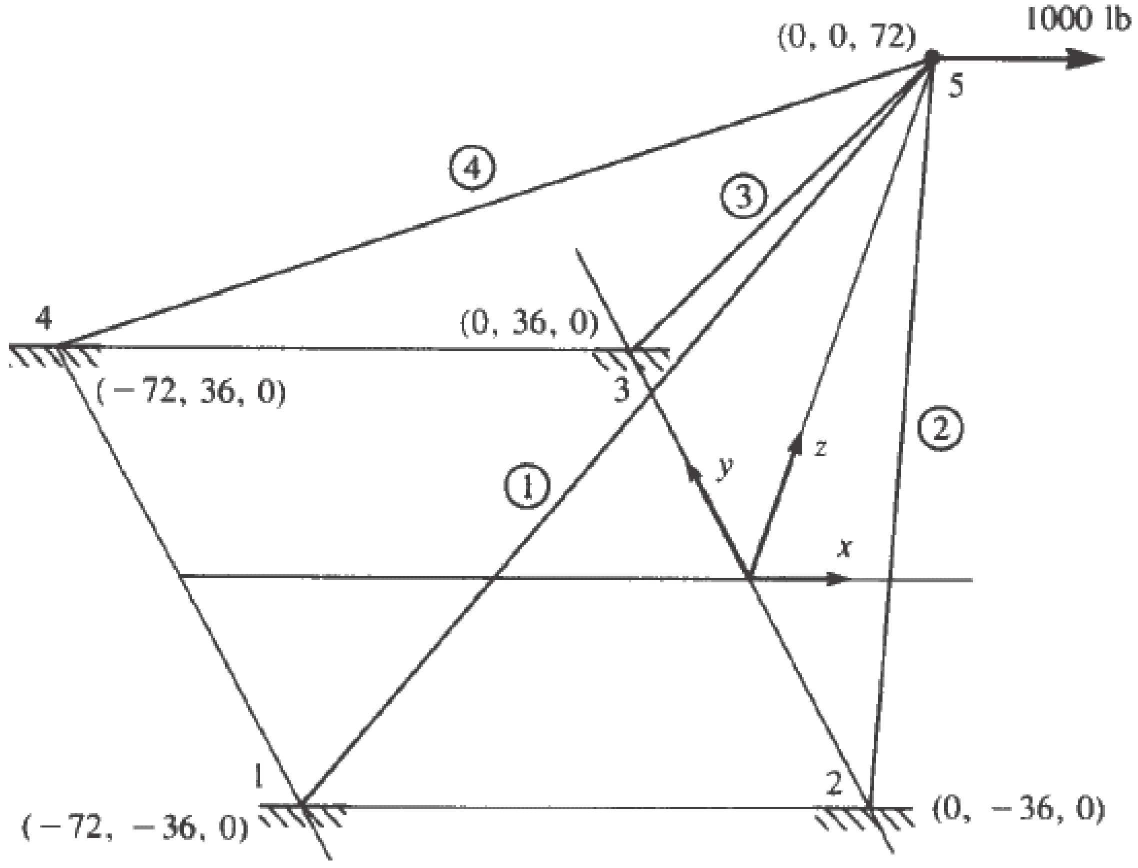
Part II – Report

Prepare a report (no more than 15 pages) including the following:

- Briefly explain the total potential energy approach to obtain the equilibrium system of equation for a structure.
- Explain what the shape functions are and how they are used in approximating the field variables.
- Definition and properties of stiffness matrix.
- Deriving the stiffness matrix for a bar element, two- and three-dimensional (planar and space) trusses
- Assembling the global stiffness matrix
- Boundary conditions and how they are imposed.
- Solving the obtained system of equations [K]{u} = {f}, and calculating nodal displacements and reaction forces
- Computation of elements stresses and internal forces in two- and three-dimensional trusses
- Briefly explain your code in “trussAnalysis.m” function.
- Use your completed program to solve the following 2D and 3D truss problems and report the results (displacements, reaction forces, stresses, and internal forces):

Problem 1 – Planar Truss
 20 kN 20 kN 20 kN 1 2 3 4 5 6 ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ 4 m 8 m 8 m
Determine the nodal displacements and the stresses in each element. All elements have E = 210 GPa and A = 10×10^{-4} m^2

Problem 2 – Space Truss



Determine the displacement of node 5, reaction forces, and the stresses in each element. Let A=4 in^2 and E=30×10^6 psi for all elements. The coordinates are in inches.

Part III – Submission and Evaluation

- **Submission:**

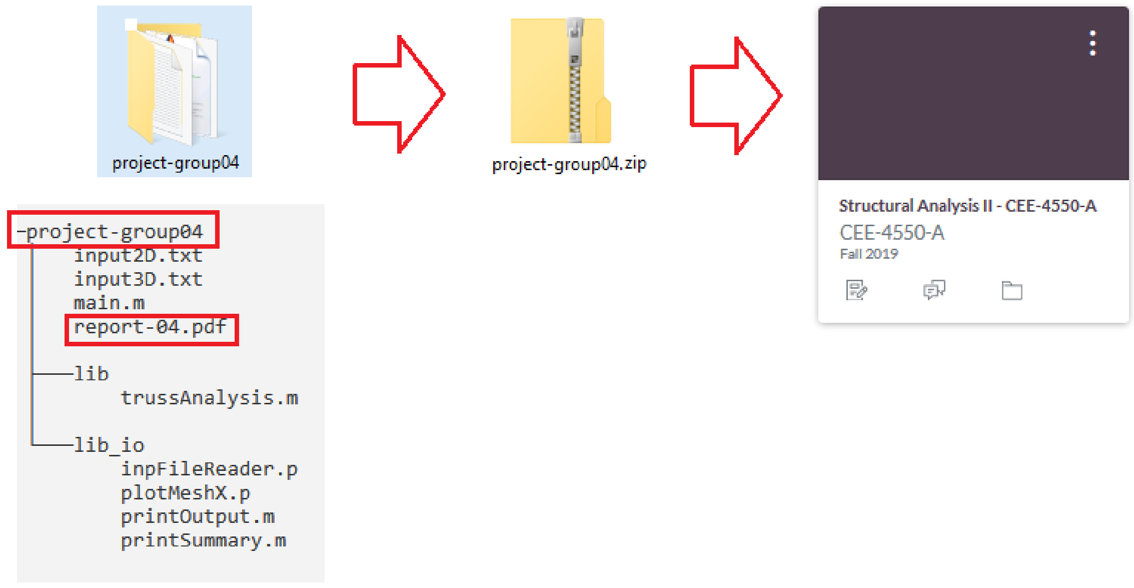
  (1) The cover page of your report must include the **group number**, and the **names** and **GT IDs** of group members.

  (2) Name your report **report-<Group Number>.pdf** and copy it into the main folder of your project.

  (3) Rename your completed project folder to **project-group<xx>** and compress it to **project-group<#>.zip** (Don't forget to rename the folder before compressing)

  (4) Submit the final zip file to canvas (It is enough to be submitted by one of the group members)

  For instance, the following is a sample submission:



```
-project-group04
    input2D.txt
    input3D.txt
    main.m
    report-04.pdf

  ├──lib
        trussAnalysis.m

  └──lib_io
        inpFileReader.p
        plotMeshX.p
        printOutput.m
        printSummary.m
```

- **Evaluation:**

  Your program will be evaluated against a few benchmark problems. In each part of this program, you are welcome to develop your own functions or use your own variable names. But, please notice that, regardless of all the development preferences, the main FEM function of your program ("trussAnalysis.m") must be able to analyze any 2D and 3D truss problems defined with an input file. In other words, at the time of your presentation, you should demonstrate that your program can analyze any planar or space truss problem, defined in the explained input file format, and display the results in a proper format on the screen and write it to an output file.