The Georgia Institute of Technology
School of Civil and Environmental Engineering
Finite Element Methods CEE6504-Spring 2020
Project Description

Designed by

Shahrokh Shahi

www.sshahi.com

Introduction

This project aims at developing a Finite Element Analysis program in MATLAB to analyze two-dimensional plane-stress and plane-strain problems. Similar to the MATLAB assignments in your homework, you are provided with some components 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 FEM 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 learn how to define an input file to input a problem data to this package. Finally, you will employ the accomplished package to analyze a few new problems. 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 and Presentation

Part III – Submission and Evaluation

Part I – MATLAB Programming

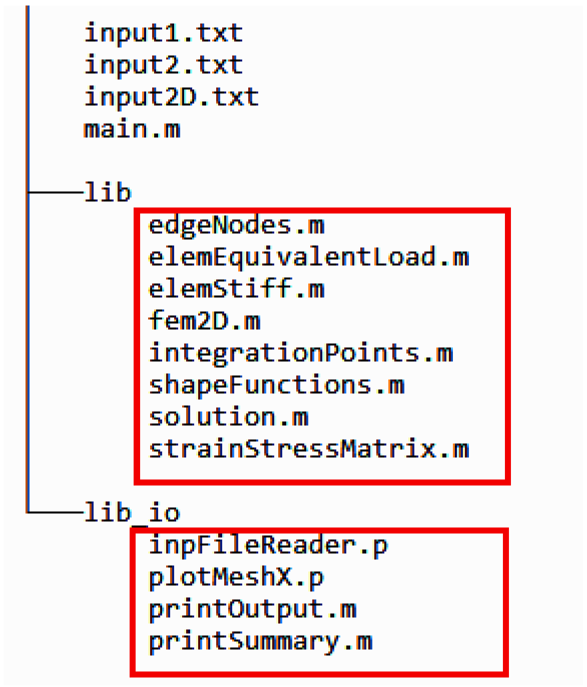
The skeleton files are available in "**project.zip**". Before explaining anything, please remember the following **important notes**:

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.

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 files structure:



```
│   input1.txt
│   input2.txt
│   input2D.txt
│   main.m
│
├───lib
│       edgeNodes.m
│       elemEquivalentLoad.m
│       elemStiff.m
│       fem2D.m
│       integrationPoints.m
│       shapeFunctions.m
│       solution.m
│       strainStressMatrix.m
│
└───lib_io
        inpFileReader.p
        plotMeshX.p
        printOutput.m
        printSummary.m
```

- 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\). Some of these functions will be called by other main functions. In this project, you need to complete these functions to obtain a working 2D FEM program. 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 fem2D.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.

The main file (main.m)
This file is the main m-file controlling the flow of the program. Once you defined 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:

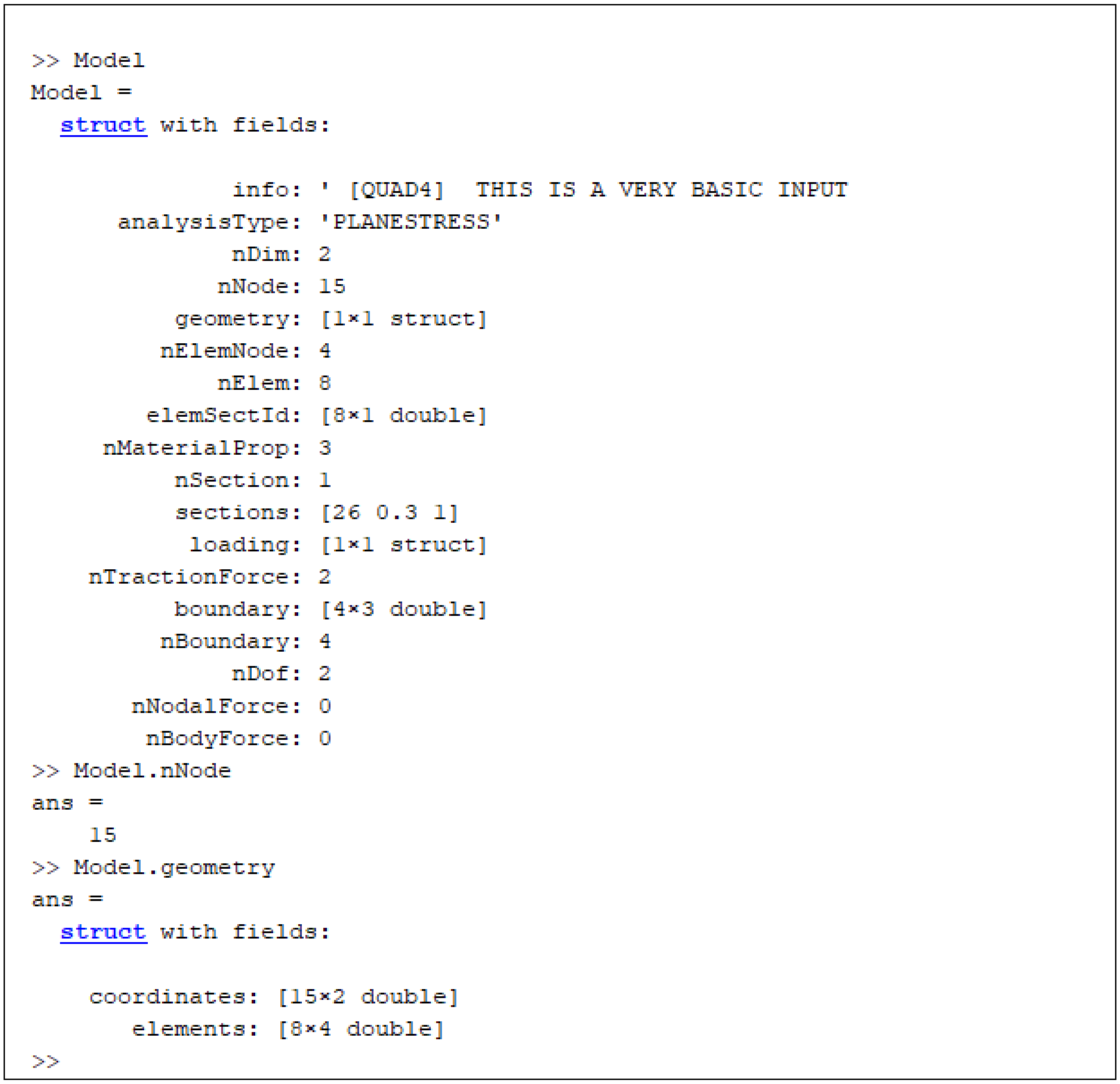
(1) Initialization: clearing the Command Window, clearing all variables, setting the output formats, and adding the required path to the MATLAB operating path. (You should NOT modify this section)

(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, these kinds of data were hard-coded at the beginning of the same m-file. However, this approach is not scalable and feasible for analyzing all problems. In this project, you must follow a specific format to define the input files. This specific format has been explained in this document (In the next section).

(3) Interpreting the input file: If you follow the instructions to define input files, the function "inpFileReader.p" (located in lib_io) can interpret the input file 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 15 and `Model.geometry` gives an inner structure named geometry including nodal coordinates and elements connectivity data.



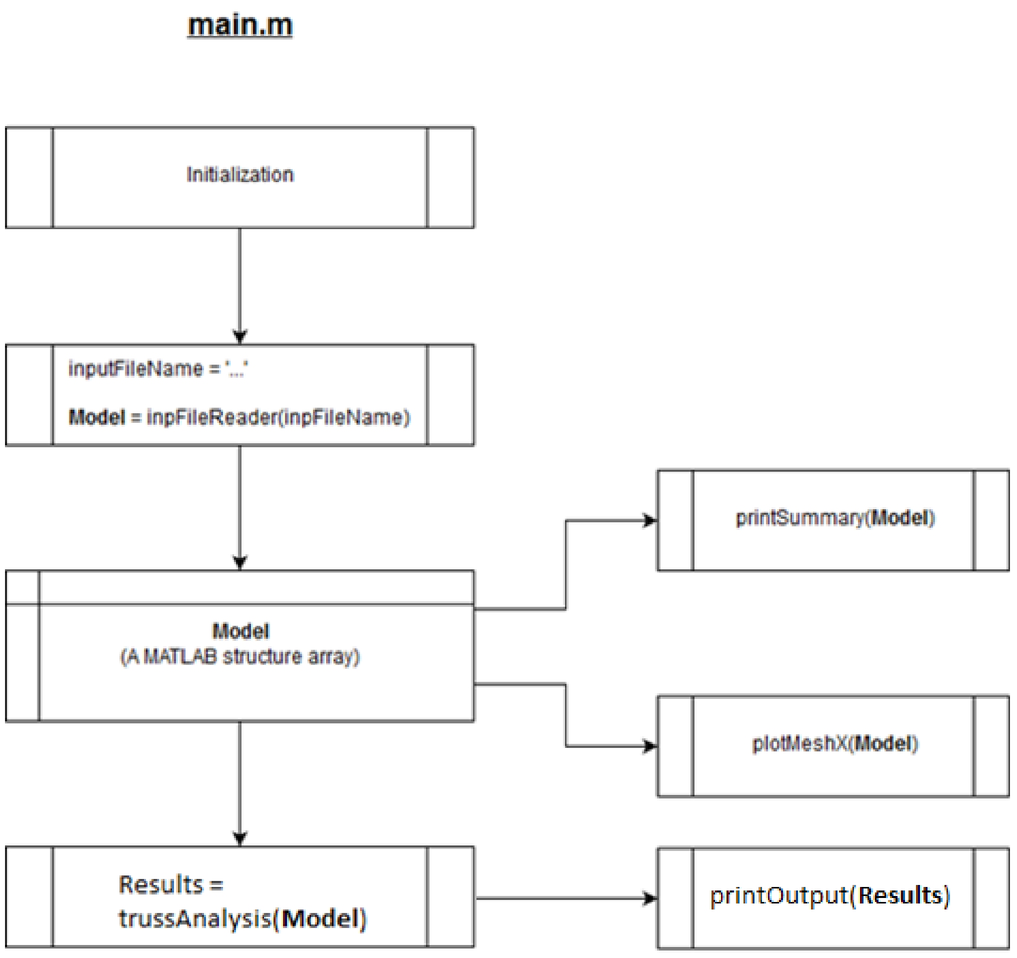
```
>> Model
Model =
  struct with fields:

              info: ' [QUAD4]  THIS IS A VERY BASIC INPUT
      analysisType: 'PLANESTRESS'
              nDim: 2
             nNode: 15
          geometry: [1×1 struct]
         nElemNode: 4
             nElem: 8
        elemSectId: [8×1 double]
     nMaterialProp: 3
          nSection: 1
          sections: [26 0.3 1]
           loading: [1×1 struct]
    nTractionForce: 2
          boundary: [4×3 double]
         nBoundary: 4
              nDof: 2
       nNodalForce: 0
        nBodyForce: 0
>> Model.nNode
ans =
    15
>> Model.geometry
ans =
  struct with fields:

    coordinates: [15×2 double]
       elements: [8×4 double]
>>
```

(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 FEM function, which is "fem2D.m". This function is located in the "lib" folder and the problem data will pass to this function by the Model struct variable. You need to complete this function and its internal functions to obtain an accomplished package. Then, you can run main.m to test your program.



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.p** 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.

Input File
Interpreter: inpFileReader.p
Model Struct

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

- **General information about input files**
    - In general, if a line starts with one of these three characters (%, #, -), it will be considered as a COMMENT line and the interpreter (inpFileReader.m) will ignore that.

      This can be very useful to add more information in the input file for future using
    - If a line starts with “*”, it means it is a COMMAND line and it follows by a space and a command. A command is a pre-defined keyword, which lets the interpreter know about the data, comes immediately after this line.

IMPORTANT REMARKS:

(1) Between the starting * and the command MUST be AT LEAST one space

(2) Between a command line and the data related to that command there must be NO empty line. In other words, the next line after a command line cannot be an empty line and it must include the data related to that command.

(3) Commands are NOT case sensitive, so for instance, “* COORDINATES”, “* coordinates”, “* Coordinates”, “*CoORdiNatEs” are all equal.

The pre-defined command keywords are presented in the following table

	Command Keyword	Description
1	INFO	General information about the problem
2	ANALYSIS	Problem type
3	COORDINATES	Nodes coordinates
4	ELEMENTS	Elements connectivity
5	SECTIONS	Sections material properties
6	NODAL	Nodal loads information
7	TRACTION	Traction forces information
8	BOUNDARY	Boundary conditions information

(1) * INFO

The line(s) comes after this line provides information about the problem. This information will be stored in the Model struct for future usages.

REMARK:

The information comes after this line is not restricted to only one line and can be multiple lines (Use enter to continue the information in a new line)

Example:

```
* INFO
This is an example of employing numerical integration to calculate the
traction forces on a plane stress/strain element edge.
[REF] Logan, D.L., 2011. A first course in the finite element method.
Cengage Learning. (Problem 10.14)
```

(2) * ANALYSIS

The line comes after this line specifies the (problem) analysis type and can be one of the following keywords:

- Plane Stress (for plane stress problems)
- Plane Strain (for plane strain problems)

REMARK:

Again, these keywords can be used in both upper case and lower case (or their combinations) and using space(s) between two words does not make any difference, e.g. Plane Stress = planestress = planeStress

Example:

```
* ANALYSIS TYPE
Plane Stress
```

(3) * COORDINATES

The lines come after this line include the nodal coordinates of the geometry, and it should be in the following format:

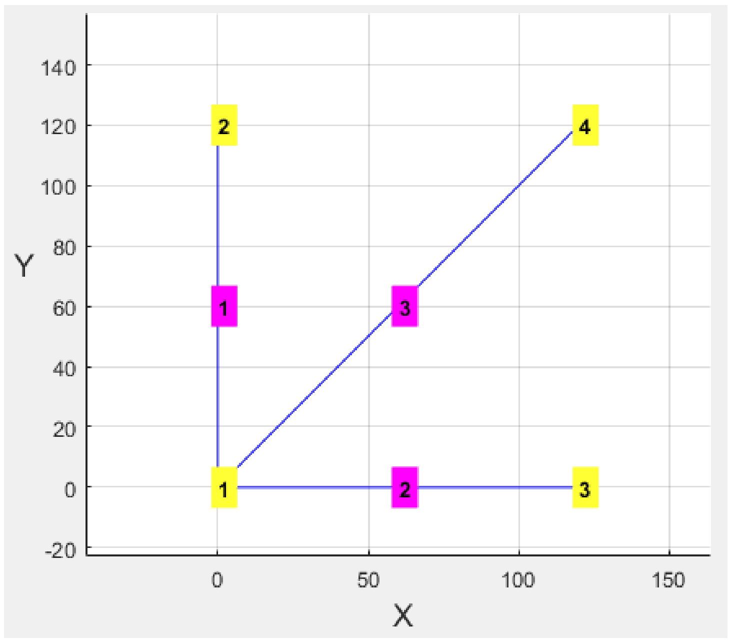
- For 2D geometries: Node ID X_i Y_i

REMARK:

The nodes can be inputted in any order. They will automatically ordered by their Node IDs.

Example:

For the following geometry,



The nodal coordinates should be inputted as follows:

```
* COORDINATES
1      0.0     0.0
2      0.0   120.0
3    120.0     0.0
4    120.0   120.0
```

(4) * ELEMENTS

The lines come after this line include elements connectivity and also section data, and it should be in the following format:

Element ID   Section ID   Node ID 1   Node ID 2   ….

where the section ID specifies the section (and the material properties of the element)

Example:

For the same geometry the element data should be inputted as the following:

```
* ELEMENTS
1  1     1  2
2  1     1  3
3  2     1  4
```

In the above example element 1 and 2 have the same material properties (Section ID 1) and element 3 has different material properties (Section ID 2). These material properties will be defined under the SECTION command.

(5) * SECTIONS

The line comes after this line include the material properties of each section and have the following format:

Section ID   Material Properties 1   Material Properties 2   …

Depending on the problem analysis type (Truss/Beam/Plane/…), the number and definition of each material property can be different. The following table describes these properties for each case:

Analysis Type	Material Properties Order
Plane Stress	Section ID, E_i, ν_i, t_i
Plane Strain	Section ID, E_i, ν_i, t_i

Example:

```
* SECTIONS
1  30e6  0.3  1
2  30e6  0.3  1
```

(6) * NODAL   or   * NODAL FORCES

The line(s) comes after this line include the nodal force(s) information

Node ID Fx_i Fy_i

Example:

```
* NODAL FORCES
1     0   -1e4
4     1e3  2e3
```

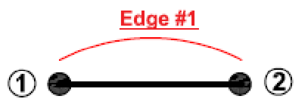
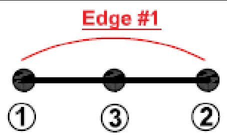
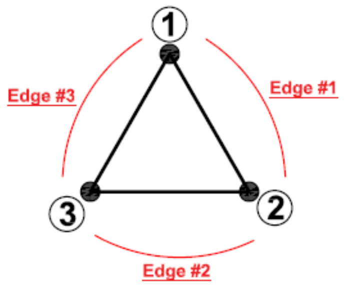
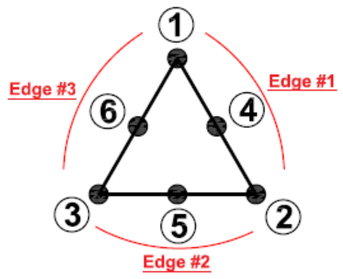
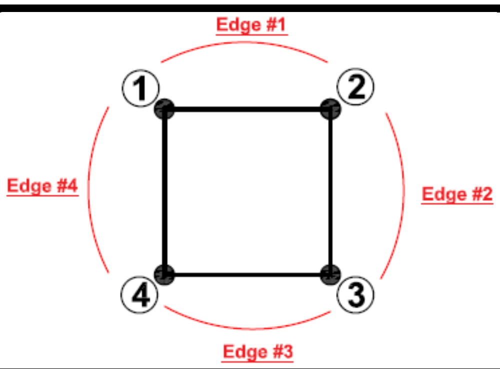
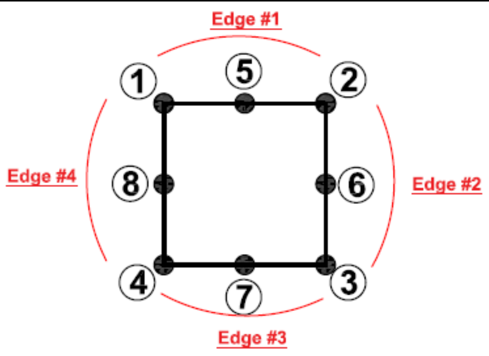
(7) * TRACTION   or   * TRACTION FORCES

The line(s) comes after this line describes the traction force(s) applied on the elements edge(s) (in 2D cases).

In general, the input should have the following format:

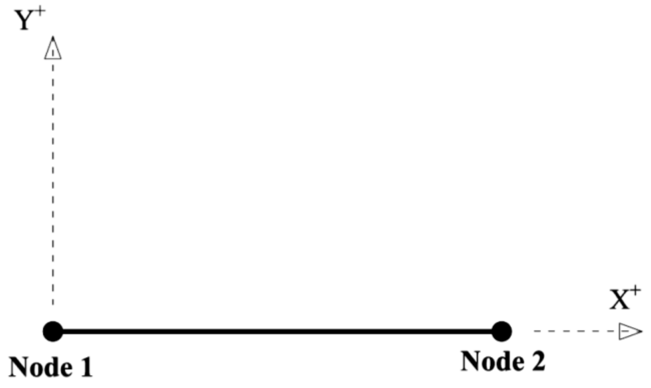
Element ID   Edge No. Tx_i Ty_i

where “Edge No.” specifies the edge to apply Tx_i and Ty_i. The convention for the edge nodes numbering is presented in the following table.

Element	Order	No of Nodes per Each Element		Edge Node Index
Bar	1	2		Edge 1 = [1, 2]
	2	3		Edge 1 = [1, 2, 3]
Triangular	1	3		Edge 1 = [1, 2] Edge 2 = [2, 3] Edge 3 = [3, 1]
	2	6		Edge 1 = [1, 2, 4] Edge 2 = [2, 3, 5] Edge 3 = [3, 1, 6]
Quadrilateral	1	4	 Edge #1 1 2 Edge #4 Edge #2 4 3 Edge #3	Edge 1 = [1, 2] Edge 2 = [2, 3] Edge 3 = [3, 4] Edge 4 = [4, 1]
	2	8	 Edge #1 1 5 2 Edge #4 8 6 Edge #2 4 7 3 Edge #3	Edge 1 = [1, 2, 5] Edge 2 = [2, 3, 6] Edge 3 = [3, 4, 7] Edge 4 = [4, 1, 8]

REMARKS:

- Forces are defined in element local coordinates
- On the edge of bar elements (Truss, Beam/Frame) the positive direction of x is from Node 1 to Node 2, and the positive direction of y-axis is upward.

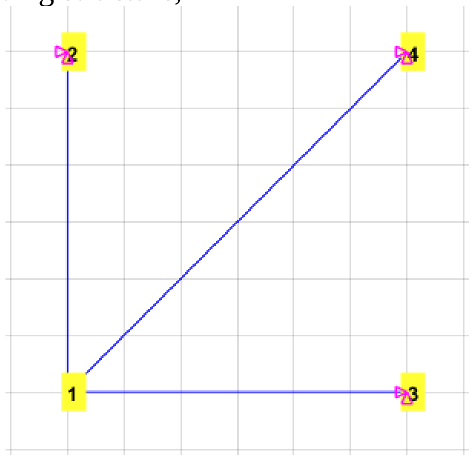


- For a bar element, edge number is always 1.

(8) * BOUNDARY or * BOUNDARY CONDITIONS

The boundary condition information will be defined in the lines after BOUNDARY command in the following format:
Node ID, DOF number, Prescribed Value

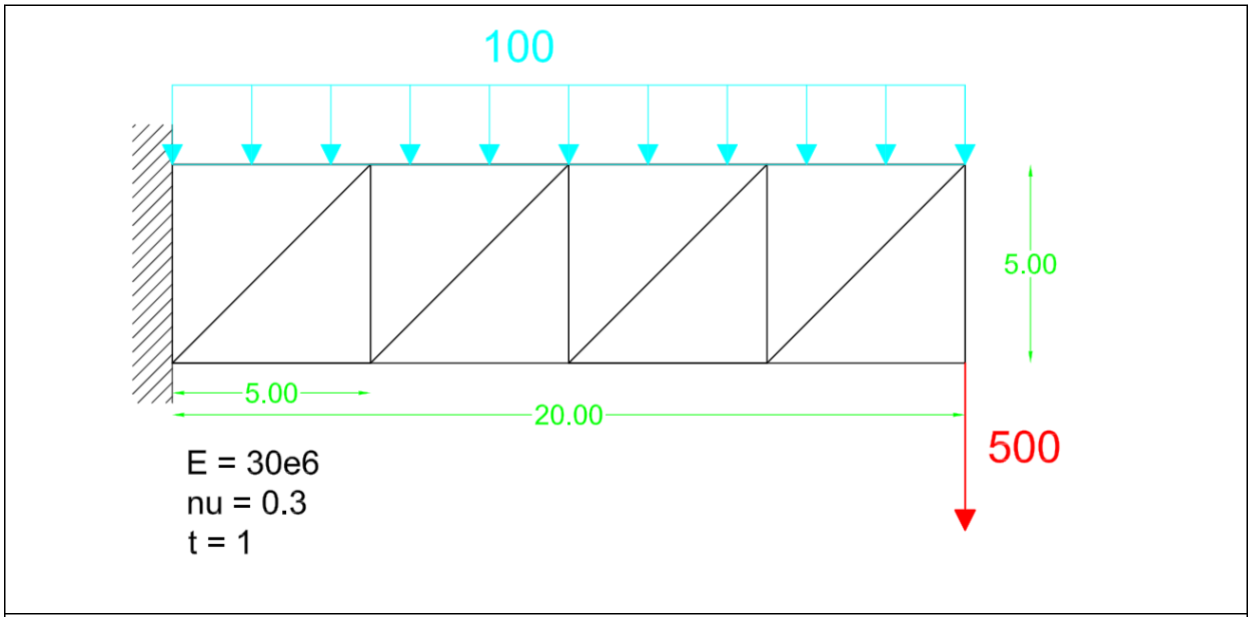
For instance, for the following structure,

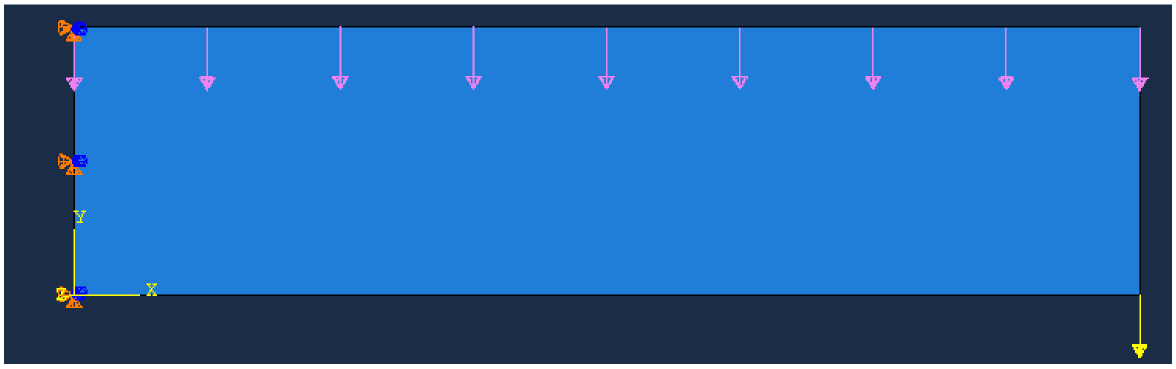


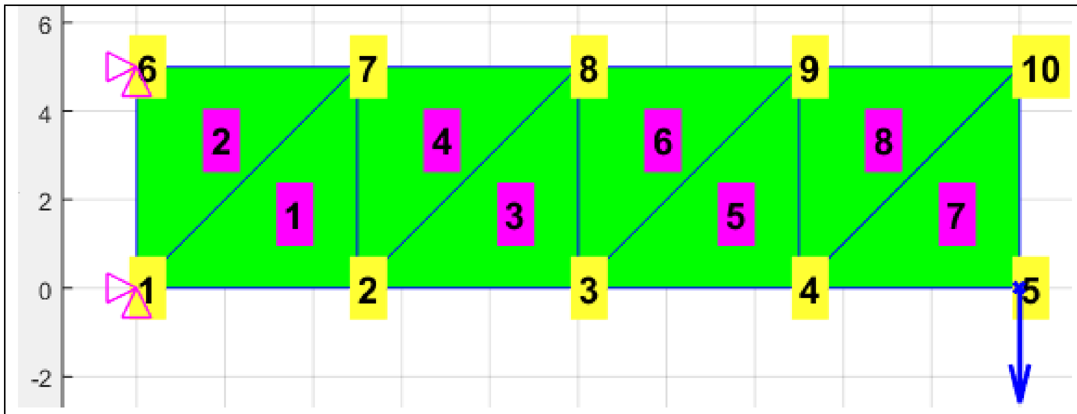
The boundary condition data is the following:

```
* BOUNDARY
2   1  0.0
2   2  0.0
3   1  0.0
3   2  0.0
4   1  0.0
4   2  0.0
```

Another Example
Consider the following problem with the material properties and loading shown on the figure:







In this case, the input file is the following:

```
* INFO
An Example of 2D Plsne Problem

* ANALYSIS TYPE
Plane Stress

* COORDINATES
1        0.0       0.0
2        5.0       0.0
3       10.0       0.0
4       15.0       0.0
5       20.0       0.0
6        0.0       5.0
7        5.0       5.0
8       10.0       5.0
9       15.0       5.0
10      20.0       5.0


* ELEMENTS
1  1    1   2   7
2  1    7   6   1
3  1    2   3   8
4  1    8   7   2
5  1    3   4   9
6  1    9   8   3
7  1    4   5  10
8  1   10   9   4


* SECTIONS
1  30e6   0.3   1


* TRACTION FORCES
2  1    0    -100
4  1    0    -100
6  1    0    -100
8  1    0    -100



* NODAL FORCES
5    0   -500



* BOUNDARY
1  1  0.0
1  2  0.0
6  1  0.0
6  2  0.0
```

GENARAL REMARKS and DIAGNOSTICS:

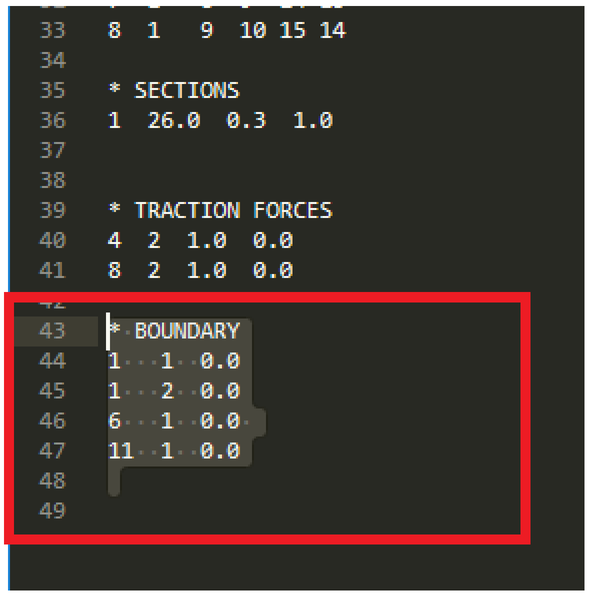
Due to the robustness of the interpreter, the input file definition is very flexible:

- The order of the sections is not restricted and they can be defined in any order. For instance, you can start the input file definition with boundary condition definition (Of course, it is always better to keep the rational order of defining sections)
- In each section (e.g. COORDINATES, ELEMENTS, SECTIONS, etc) the data line can be in any order (the data will automatically be ordered by the interpreter)
- Commands can be in both lower and upper case or a combination of them. This is also true for the Analysis Type definition
- You can add as many as comments or separator lines or even empty lines that you need to improve the clarity and readability of an input file. (Note: do NOT forget to use #, %, or – at the beginning of each comment line)

Limitations (Diagnostics):

There are some limitations in defining the input file, which may lead to failure in successfully reading of the input file:

- In a command line, there MUST be a space between the starting * and the command.
- Load equation can only be defined for TRACTION FORCES.
- NO SPACE is allowed in load equation definition
- NO SPACE at the end of each data line is allowed. For instance, the following input file is invalid because at the third line of boundary condition definition, there is an additional space at the end of line:



```
8  1   9  10 15 14

* SECTIONS
1  26.0  0.3  1.0


* TRACTION FORCES
4  2  1.0  0.0
8  2  1.0  0.0

* BOUNDARY
1   1  0.0
1   2  0.0
6   1  0.0 
11  1  0.0
```

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
  +--- analysisType:  PLANE STRESS/PLANE STRAIN/
  +--- nDim:  1=1D / 2=2D / 3=3D
  +--- nNode: total number of nodes
  +--- nElem: total number of elements
  +--- nElemNode: number of nodes per each element
  +--- nDof:  number of degrees-of-freedom per each node
  +--- nSection:  number of section
  +--- nMaterialProp: number of material properties to define section(s)
  |
  |
  +--- geometry: an inner structure including geometry data
  |  |
  |  +--- coordinates: node coordinates matrix (nNode x nDim)
  |  +--- elements: elements connectivity (nElem x nElemNode)
  |
  +--- elemSectId: the section id for each element (mElem x 1)
  +--- sections: section data/material (nSect x nMaterialProp)
  |
  +--- nNodalForce:number of nodal forces
  +--- nTractionForce: number of element traction forces
  |
  +--- loading: an inner structure including loading data
  |  |
  |  +--- nodalForces: nodal forces data (nNodalForce x nDof)
  |  +--- tractionForces: traction forces data (nTra. x nDim)
  |
  +--- nBoundary: number of boundary condition entries
  +--- boundary: boundary condition data (nBoundary x 3)
                 3: nodeID, dof, value
```

Notes:

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

- The Model structure includes two inner structures (nested structures):
    - geometry, which includes the nodal coordinates and elements connectivity data
    - loading, which includes nodal (concentrated) forces and traction forces 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.

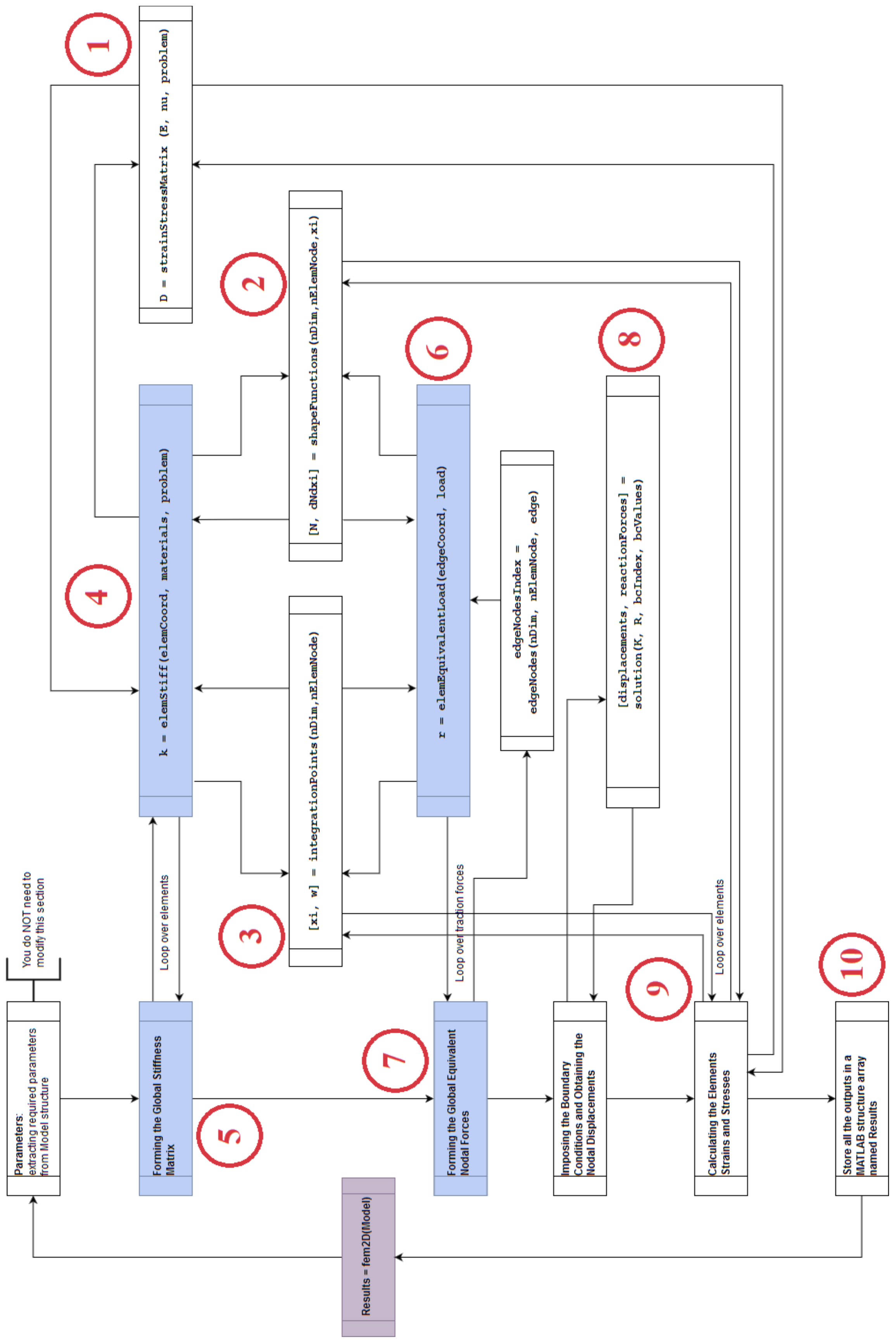
Accomplishing the Package:

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?
fem2D.m	The main FEM function to analyze a 2D plane stress/strain structure.	Yes
elemStiff.m	This function calculates the element stiffness matrix by using numerical integration.	Yes
strainStressMatrix.m	This function generates the strain-stress [D] matrix.	Yes
elemEquivalentLoad.m	This function calculates the equivalent nodal forces from traction forces by conducting numerical integration.	Yes
edgeNodes.m	This function provides a list of indices of element edge (side) nodes.	No
shapeFunctions.m	This function calculates the shape functions and their derivatives at a given integration point, xi, in the natural coordinates.	Yes

integrationPoints.m	This function provides Gauss quadrature points and weights required in the numerical integration procedure.	Yes
Solution.m	This function imposes the boundary conditions on the assembled global matrices and calculates the displacements and reaction forces	Yes
printResult.m	Displaying a summary of output results stored in the Results structure	Yes

Some of these functions will be used by multiple other functions. For instance, the function “integrationPoints.m” is an essential component for conducting numerical integration in calculating (i) element stiffness matrices, (ii) the traction equivalent nodal forces, and (iii) the elements strains and stresses at the end of the analysis. Therefore, it is a good idea to complete this kind of functions at the beginning. The following figure demonstrates a schematic diagram of all computational functions and their dependencies. The numbers in the red circles are the suggested order to accomplish the tasks.


Parameters:
extracting required parameters from Model structure
You do NOT need to modify this section
Forming the Global Stiffness Matrix
Loop over elements
k = elemStiff(elemCoord, materials, problem)
D = strainStressMatrix (E, nu, problem)
[N, dNdxi] = shapeFunctions(nDim,nElemNode,xi)
[xi, w] = integrationPoints(nDim,nElemNode)
Forming the Global Equivalent Nodal Forces
Loop over traction forces
r = elemEquivalentLoad(edgeCoord, load)
edgeNodesIndex = edgeNodes(nDim, nElemNode, edge)
Imposing the Boundary Conditions and Obtaining the Nodal Displacements
[displacements, reactionForces] = solution(K, R, bcIndex, bcValues)
Calculating the Elements Strains and Stresses
Loop over elements
Store all the outputs in a MATLAB structure array named Results
Results = fem2D(Model)
1
2
3
4
5
6
7
8
9
10

(1) **`D = strainStressMatrix (E, nu, problem)`**

This function takes material properties (E and nu), and the problem type (plain stress/ plane strain) as the inputs and generates the strain-stress [D] matrix. You only need to define this 3-by-3 matrix for each case.

(2) **`[N, dNdxi] = shapeFunctions(nDim,nElemNode,xi)`**

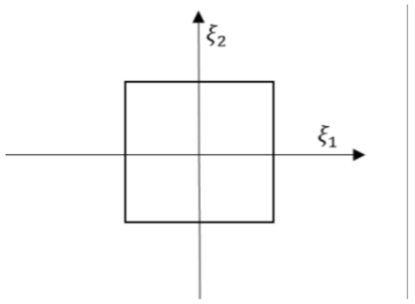
This function calculates the shape functions and their derivatives at a given integration point, xi, in the natural coordinates. The outputs are the following:

- N: A vertical vector of shape functions values at the given integration point (xi):

$$N = \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ \vdots \\ N_{nElemNode} \end{bmatrix}$$

The length of this vector is nElemNode. For example, for a linear quadrilateral element with four nodes per each element (nElemNode = 4), the vector N will be 4-by-1:

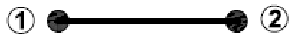
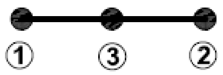
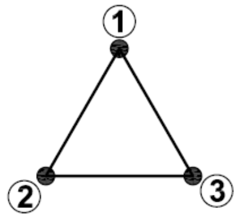
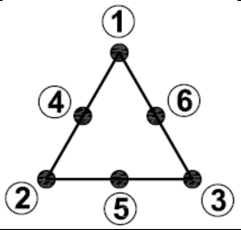
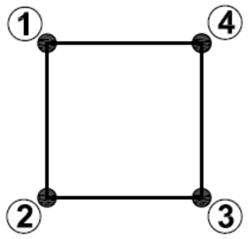
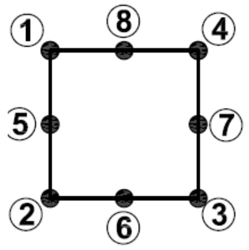
$$N = \begin{bmatrix} N_1 = \frac{1}{4}(1-\xi_1)(1-\xi_2) \\ N_2 = \frac{1}{4}(1+\xi_1)(1-\xi_2) \\ N_3 = \frac{1}{4}(1+\xi_1)(1+\xi_2) \\ N_4 = \frac{1}{4}(1-\xi_1)(1+\xi_2) \end{bmatrix}$$



- dNdxi: A matrix including shape functions derivatives (nElemNode-by-nDim). For instance, for the same quadrilateral element, the shape function derivatives matrix will have the following form:

$$dNdxi = \begin{bmatrix} \frac{\partial N_1}{\partial \xi_1} & \frac{\partial N_1}{\partial \xi_2} \\ \frac{\partial N_2}{\partial \xi_1} & \frac{\partial N_2}{\partial \xi_2} \\ \frac{\partial N_3}{\partial \xi_1} & \frac{\partial N_3}{\partial \xi_2} \\ \frac{\partial N_4}{\partial \xi_1} & \frac{\partial N_4}{\partial \xi_2} \end{bmatrix}$$

In this project, you need to define the shape functions and their derivatives for the all elements illustrated in the following table. In the skeleton file, you can find specific places to input the required expressions.

Element	Order	No of Nodes per Each Element (nElemNode)	
Bar	1	2	
	2	3	
Triangular	1	3	
	2	6	
Quadrilateral	1	4	
	2	8	

Note:
You do not have to follow the dimension of N and dNdxi matrix. However, bear in mind that if you choose another approach to form these matrices, you need to be careful about their consistency with the rest of the program.

(3) **`[xi, w] = integrationPoints(nDim,nElemNode)`**

This function takes nDim and nElemNode as inputs and returns a matrix including the Gauss quadrature points and a vector including the associated weights. These values will be used in the numerical integration procedure. The matrix, xi, is an nDim-by-nPoints matrix (nPoints is the number of integration points required in each case). The length of the weights vector, w, is also equal to nPoints.

(4) **`k = elemStiff(elemCoord, materials, problem)`**

After completing items (1) to (3), you can complete the elemStiff.m function, which calculates an element stiffness matrix with the given arguments:

- elemCoord: It is a (nElemNode-by-nDim) matrix including the nodal coordinates of the element.
- materials: It is vector including the element material properties, E, ν, t, respectively. These values will be used to obtain matrix [D] by calling the function, "strainStressMatrix.m" that you have completed in item (1).
- problem : This is a string variable specifying the problem type (Plane Stress / Strain) and employed to specify which [D] matrix needs to be evaluated and used to calculate the element stiffness matrix.

In this function, you should use the Gauss quadrature points and weights (generated in line #22 in this function) and implement the numerical integration method to calculate the element stiffness matrix.

In general, the element stiffness matrix is obtained by following equations:

$$k = \iiint_V \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\,dV$$

where B is the strain-displacement matrix. In the case of a two dimensional element, the stiffness matrix is calculated by

$$k = t\iint_A \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\,dA$$

where t is the thickness of the element. By using isoparametric formulation, in the natural coordinates, this integral transforms to

$$k = t\int_{-1}^{1}\int_{-1}^{1}\mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\ \det\mathbf{J}\ d\xi_1 d\xi_2$$

where J is the Jacobian matrix. Now, this integral can be calculated numerically by the Gauss-Legendre quadrature approach:

$$k = \sum_{i=1}^{nPoints} \mathbf{B}^{\mathrm{T}} \times \mathbf{D} \times \mathbf{B} \times \det\mathbf{J} \times t \times w_i$$

Therefore, to calculate the element stiffness matrix with numerical integration, you need to evaluate the [B] matrix and the Jacobian matrix at all the integration points within an element and then add them together.

Hints: To obtain the [B] matrix at each integration point, you need to evaluate the shape functions derivatives at that integration point.

(5) Now, you can come back to the "fem2D.m" and complete the section (B), where you need to implement the assembly procedure to form the global stiffness matrix (In the for-loop over all elements).
This task is very similar to what you have done in one of your MATLAB assignments.

(6) **`r = elemEquivalentLoad(edgeCoord, load)`**
Next section in the "fem2D.m" is the section (C) forming the vector of global equivalent nodal forces. This vector is a combination of elements equivalent nodal forces calculated from applied traction forces and the (concentrated) nodal forces.
To obtain the equivalent nodal forces, at first you need to find nodes on the side (edge) of the elements on which the traction force is applied. Then, you can find the coordinates of those nodes and conduct a numerical integration to calculate the equivalent nodal forces. In an isoparametric formulation:

$$f = \int_{-1}^{1}\int_{-1}^{1}\begin{bmatrix}T_x\\T_y\end{bmatrix}\mathbf{N}^{\mathrm{T}}\ \det\mathbf{J}\ d\xi_1 d\xi_2$$

This procedure should be done in the "elemEquivalentLoad.m" function. Again, you must implement the Gauss-Legendre quadrature approach to evaluate the above integral and calculate the equivalent nodal forces, numerically.

(7) Now, you can come back to the "fem2D.m" again, and complete section (C). In this step, at first you need to implement the assembly procedure to form the global vector of equivalent nodal forces. Then, you should add the corresponding concentrated nodal forces, if there is any, to the global vector in the loop over all the concentrated nodal forces.

(8) **`[displacements, reactionForces] = solution(K, R, bcIndex, bcValues)`**
Next task is very similar to one of your MATLAB assignments. In this step, you must complete the "solution.m" function to impose the boundary conditions, solve the system to obtain the displacements, and calculate the reaction forces.
This function will be called in the section (D) of the "fem2D.m" function, where the assembled stiffness (K) and nodal forces (F) matrices, and the boundary conditions data (bcIndex and bcValues) are passed as the input arguments to this function.

(9) Complete the section (E) in the "fem2D.m" function to calculate the strain and stress components at each integration points in each element. To complete this task, you need two nested for-loops (the outer over all elements, and the inner over all integration points). Now you have all the other functions that you need to calculate the strains and stresses:

- "integrationPoints.m" to generate the lists of integration points and weights
- "strainStressMatrix.m" to generate the stress-strain material matrix [D], which id need to calculate the stress components $\{\sigma\} = [D]\{\varepsilon\}$
- "shapeFunctions.m" to evaluate the shape functions and their derivatives at the integration points

(10) Storing all the results, including displacements, reaction forces, strains, and stresses in an output structure array named "Results". This structure is the output of this function ("fem2D.m") 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.stresses`[1]	3 × nPoints × nElem	The output element stresses at integration points
6	`Results.strains`[1]	3 × nPoints × nElem	The output element strains at integration points

(1) For more clarification, consider the stress (and strain) obtained in a first order quadrilateral elements (Q4) in which we have four Gaussian integration points. Therefore, at each integration point, we obtain three component of stress (σ_x, σ_y and τ_{xy}) and three components of strain ($\varepsilon_x, \varepsilon_y$ and γ_{xy}). In this case, the stress (and similarly strain) outputs must be formed in the following 3D array:

- Stress in element 1: Results.stresses(:,:,1) $= \begin{bmatrix} \sigma_x^1 & \sigma_x^2 & \sigma_x^3 & \sigma_x^4 \\ \sigma_y^1 & \sigma_y^2 & \sigma_y^3 & \sigma_y^4 \\ \tau_{xy}^1 & \tau_{xy}^2 & \tau_{xy}^3 & \tau_{xy}^4 \end{bmatrix}$
- Stress in element 2: Results.stresses(:,:,2) $= \begin{bmatrix} \sigma_x^1 & \sigma_x^2 & \sigma_x^3 & \sigma_x^4 \\ \sigma_y^1 & \sigma_y^2 & \sigma_y^3 & \sigma_y^4 \\ \tau_{xy}^1 & \tau_{xy}^2 & \tau_{xy}^3 & \tau_{xy}^4 \end{bmatrix}$
- …

where the superscripts demonstrates the integration point in which the components are calculated. Accordingly, the dimension of the associated array will be 3 × 4 × nElem.

(11) Completing **`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 displayed on the Command Window. Your outputs should be very similar to the following:

```
==========================================================================
                         O U T P U T     S U M M A R Y
==========================================================================
__________________________________________________________________________
                              Nodal Displacements
--------------------------------------------------------------------------
[   1]       0.00000           0.00000
[   2]       0.03846           0.00000
[   3]       0.07692           0.00000
[   4]       0.11538           0.00000
[   5]       0.15385           0.00000
[   6]       0.00000          -0.01154
[   7]       0.03846          -0.01154
[   8]       0.07692          -0.01154
[   9]       0.11538          -0.01154
[  10]       0.15385          -0.01154
[  11]       0.00000          -0.02308
[  12]       0.03846          -0.02308
[  13]       0.07692          -0.02308
[  14]       0.11538          -0.02308
[  15]       0.15385          -0.02308
__________________________________________________________________________
                                Reaction Forces
--------------------------------------------------------------------------
[   1]      -0.50000
[   2]      -0.00000
[   3]      -1.00000
[   4]      -0.50000
_____________________________________________________________________________________________________________________
                                                  Strains and Stresses
---------------------------------------------------------------------------------------------------------------------
                                                   Element Number = 1
---------------------------------------------------------------------------------------------------------------------
Integration      Location         Location                        Strains                              Stresses
   Point        (Natural)         (global)                 e11       e22       e12          s11      s22      s12
---------------------------------------------------------------------------------------------------------------------
1           ( -0.5774, -0.5774)   (  0.2113,  0.2113)      0.0385  -0.0115   0.0000      1.0000    0.0000   0.0000
2           (  0.5774, -0.5774)   (  0.7887,  0.2113)      0.0385  -0.0115   -0.0000      1.0000    0.0000   -0.0000
3           ( -0.5774,  0.5774)   (  0.2113,  0.7887)      0.0385  -0.0115   0.0000      1.0000    0.0000   0.0000
4           (  0.5774,  0.5774)   (  0.7887,  0.7887)      0.0385  -0.0115   -0.0000      1.0000    -0.0000   -0.0000
```

```
                                                 Element Number = 2
            ---------------------------------------------------------------------------------------------------------------------
            Integration     Location        Location                       Strains                        Stresses
               Point       (Natural)        (global)               e11      e22      e12          s11     s22     s12
            ---------------------------------------------------------------------------------------------------------------------
            1           ( -0.5774, -0.5774)   (  1.2113,  0.2113)     0.0385  -0.0115   0.0000     1.0000    0.0000   0.0000
            2           (  0.5774, -0.5774)   (  1.7887,  0.2113)     0.0385  -0.0115   -0.0000     1.0000    -0.0000   -0.0000
            3           ( -0.5774,  0.5774)   (  1.2113,  0.7887)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000
            4           (  0.5774,  0.5774)   (  1.7887,  0.7887)     0.0385  -0.0115   -0.0000     1.0000    -0.0000   -0.0000



                                                 Element Number = 3
            ---------------------------------------------------------------------------------------------------------------------
            Integration     Location        Location                       Strains                        Stresses
               Point       (Natural)        (global)               e11      e22      e12          s11     s22     s12
            ---------------------------------------------------------------------------------------------------------------------
            1           ( -0.5774, -0.5774)   (  2.2113,  0.2113)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000
            2           (  0.5774, -0.5774)   (  2.7887,  0.2113)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000
            3           ( -0.5774,  0.5774)   (  2.2113,  0.7887)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000
            4           (  0.5774,  0.5774)   (  2.7887,  0.7887)     0.0385  -0.0115   0.0000     1.0000    0.0000   0.0000

                                                 Element Number = 4
            ---------------------------------------------------------------------------------------------------------------------
            Integration     Location        Location                       Strains                        Stresses
               Point       (Natural)        (global)               e11      e22      e12          s11     s22     s12
            ---------------------------------------------------------------------------------------------------------------------
            1           ( -0.5774, -0.5774)   (  3.2113,  0.2113)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000
            2           (  0.5774, -0.5774)   (  3.7887,  0.2113)     0.0385  -0.0115   0.0000     1.0000    0.0000   0.0000
            3           ( -0.5774,  0.5774)   (  3.2113,  0.7887)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000
            4           (  0.5774,  0.5774)   (  3.7887,  0.7887)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000

                                                 Element Number = 5
            ---------------------------------------------------------------------------------------------------------------------
            Integration     Location        Location                       Strains                        Stresses
               Point       (Natural)        (global)               e11      e22      e12          s11     s22     s12
            ---------------------------------------------------------------------------------------------------------------------
            1           ( -0.5774, -0.5774)   (  0.2113,  1.2113)     0.0385  -0.0115   0.0000     1.0000    0.0000   0.0000
            2           (  0.5774, -0.5774)   (  0.7887,  1.2113)     0.0385  -0.0115   0.0000     1.0000    0.0000   0.0000
            3           ( -0.5774,  0.5774)   (  0.2113,  1.7887)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000
            4           (  0.5774,  0.5774)   (  0.7887,  1.7887)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000

                                                 Element Number = 6
            ---------------------------------------------------------------------------------------------------------------------
            Integration     Location        Location                       Strains                        Stresses
               Point       (Natural)        (global)               e11      e22      e12          s11     s22     s12
            ---------------------------------------------------------------------------------------------------------------------
            1           ( -0.5774, -0.5774)   (  1.2113,  1.2113)     0.0385  -0.0115   0.0000     1.0000    0.0000   0.0000
            2           (  0.5774, -0.5774)   (  1.7887,  1.2113)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000
            3           ( -0.5774,  0.5774)   (  1.2113,  1.7887)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000
            4           (  0.5774,  0.5774)   (  1.7887,  1.7887)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000

                                                 Element Number = 7
            ---------------------------------------------------------------------------------------------------------------------
            Integration     Location        Location                       Strains                        Stresses
               Point       (Natural)        (global)               e11      e22      e12          s11     s22     s12
            ---------------------------------------------------------------------------------------------------------------------
            1           ( -0.5774, -0.5774)   (  2.2113,  1.2113)     0.0385  -0.0115   0.0000     1.0000    0.0000   0.0000
            2           (  0.5774, -0.5774)   (  2.7887,  1.2113)     0.0385  -0.0115   -0.0000     1.0000    0.0000   -0.0000
            3           ( -0.5774,  0.5774)   (  2.2113,  1.7887)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000
            4           (  0.5774,  0.5774)   (  2.7887,  1.7887)     0.0385  -0.0115   -0.0000     1.0000    -0.0000   -0.0000

                                                 Element Number = 8
            ---------------------------------------------------------------------------------------------------------------------
            Integration     Location        Location                       Strains                        Stresses
               Point       (Natural)        (global)               e11      e22      e12          s11     s22     s12
            ---------------------------------------------------------------------------------------------------------------------
            1           ( -0.5774, -0.5774)   (  3.2113,  1.2113)     0.0385  -0.0115   -0.0000     1.0000    -0.0000   -0.0000
            2           (  0.5774, -0.5774)   (  3.7887,  1.2113)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000
            3           ( -0.5774,  0.5774)   (  3.2113,  1.7887)     0.0385  -0.0115   0.0000     1.0000    -0.0000   0.0000
            4           (  0.5774,  0.5774)   (  3.7887,  1.7887)     0.0385  -0.0115   0.0000     1.0000    0.0000   0.0000


            =====================================================================================================================
>>
```

Part II – Report and Presentation

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 defined elements and the numerical integration procedure
- 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 strains at integration points
- Briefly explain your code and development steps.

Presentations

Prepare a 10 minutes PowerPoint presentation explaining your development procedure and findings in this project

Part III – Submission and Evaluation

- **Submission:**

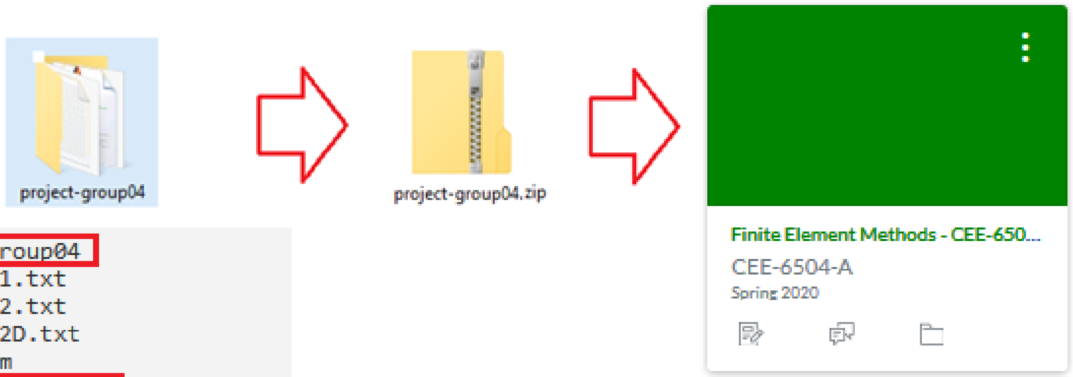
(1) The cover page of your report must include the **group number**, and the **names** and **GT IDs** of group members.

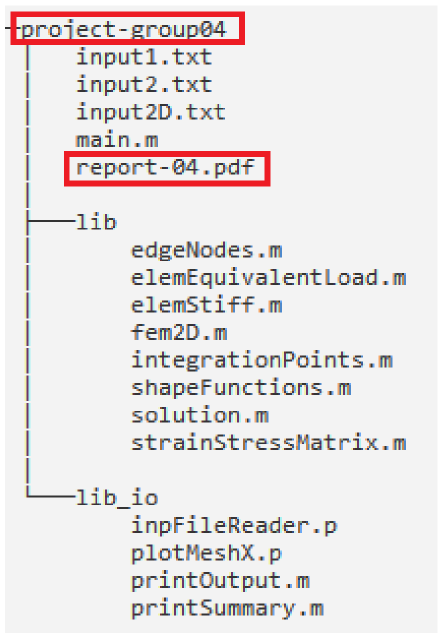
(2) Name your report **report-<Group Number>.pdf** and copy it into the main folder of your project.

(3) Rename your completed project folder to **project-group<##>** 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
│     input1.txt
│     input2.txt
│     input2D.txt
│     main.m
│     report-04.pdf
│
├───lib
│       edgeNodes.m
│       elemEquivalentLoad.m
│       elemStiff.m
│       fem2D.m
│       integrationPoints.m
│       shapeFunctions.m
│       solution.m
│       strainStressMatrix.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 ("fem2D.m") must be able to analyze the problems defined within the "Model" structure. In other words, at the time of your presentation, you should demonstrate that your program can analyze any 2D plane strain/stress problem, defined with the explained input file format, and displays the results in a proper format on the screen and write it to an output file.