Lehigh University

Advanced Technology for Large Structural Systems

CEE 466 – Advanced Finite Element Methods

Project Report – Finite Element Analysis Program

By: Soheil Sadeghi

Instructor: Dr. Paolo Bocchini

October 2017

Contents

Introduction 1
Challenges 1
Verifying Examples 2
Example One – Tapered bar 2
Example Two – L-shaped beam structure 3
Example Three - three story steel frame 4
Example Four – truss beam 6
Example Five – indeterminate beam 7
Example Sixth – support settlement 8
Conclusion 10

Introduction

In this project, the target is to develop a program that using manually entered inputs, analyze a 2D model consisting of linear elements. The operator is needed to enter all required inputs in three INPUT files as explained in the program manual. As a brief description of the entire program, it is worth to mention that the program digest inputs and generate stiffness matrices and equivalent nodal load vectors for each elements. In addition, regarding the geometry of the program, rotation and Boolean matrices are generated. After these two steps, the program is prepared to produce the global stiffness matrix and total nodal load vector (containing both equivalent and point nodal loads). Finally, the governing Hook's equation of the global system is solved and displacement can be calculated.

The program has the following features:

- The program is capable to utilize four types of linear elements for analysis: 1) Truss / Bar elements, 2) Bernoulli element (with direct stiffness matrix), 3) Timoshenko element (with direct stiffness matrix), and 4) Timoshenko element (finite element with linear shape function).
- The program automatically refine elements if the FE Timoshenko elements are used in a model (automeshing feature). The program gets a maximum size of mesh as an input and break longer elements to pieces with no longer than the maximum length.
- The program is capable to consider all internal discontinuities in a model, such as internal hinges. These disconnections should be introduced in accordance to the instructions by the operator and the program automatically resolve corresponding DOFs in the global formulation.
- As the first verifying example needed, the program is capable of automatically refining a non-prismatic element with a linearly changing sectional depth. The ability is placed in the sec_prop.m file as one case of the library.
- Nodal displacements can be applied externally at each of the model nodes and the program is able to capture their effects. The operator is again needed to define imposed displacements carefully.

Challenges

The program includes different sections and there were several challenges in each of them. Here a brief explanation of some of these challenges are brought, to name but a few.

1) Automeshing: this function was the most struggling part for me, because it leads all variables to be upgraded. The way that I did that is to do meshing before computing any stiffness matrix or nodal loads. In fact, in my code, meshing is not a separated function and it is a part of the ele_nodes.m file, which is in fact one of the INPUT files. The length of elements are computed immediately after entering the elements' inputs and if meshing needed, all parameters are upgraded with respect to the number of elements. There were numerous challenges in this part that cannot be explained in this brief report. However, at the end I could debug and utilize this feature.
2) Disconnections: the second challenge that I can mention was my struggles to handle disconnections. I coded this part as the lecture notes suggested, however when Bar elements were used, singularity errors were popping up. Later I realized that the Bar element is automatically resolved connectivity wise and there is no need to condense element stiffness matrices for these elements. Now in my code, there is an 'if' condition, checking whether the element type is Bar, neglect the condensation section. The problem was solved by this correction.
3) New nodes and elements tagging: To tag new nodes and elements caused by meshing, the program is prone to a systemic error which may not be realized easily, and that is the repetitive node or

element tags. To avoid this error, I have used a more specific tagging logic, which reduced this problem considerably.

In the following sections, each of the verifying examples are described and program results are shown and assessed.

Verifying Examples

Example One – tapered bar

In this problem an axially loaded non-prismatic element is under tensile load at the right tip. The section is varying along the length with a circular cross section from radius of 55 mm at A to 23 mm at B. The element is discretized with different number of elements and the result accuracy is compared with the theoretical result from a closed form solution.

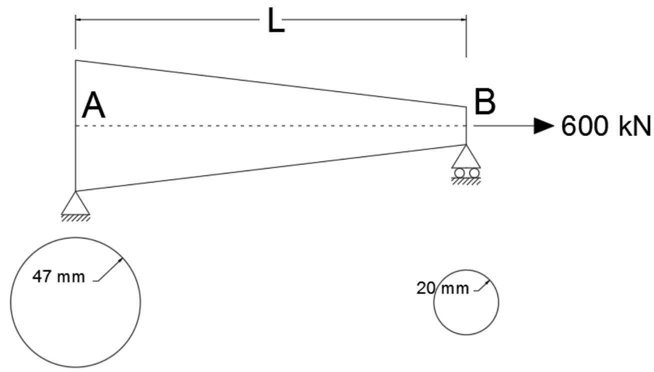


Figure 1 – Example One

In Figure 2, it is shown that how the discretization affects our approximation from the real depth-varying beam. Obviously, we expect to see better match in results of the model and closed form solution by refining the element.

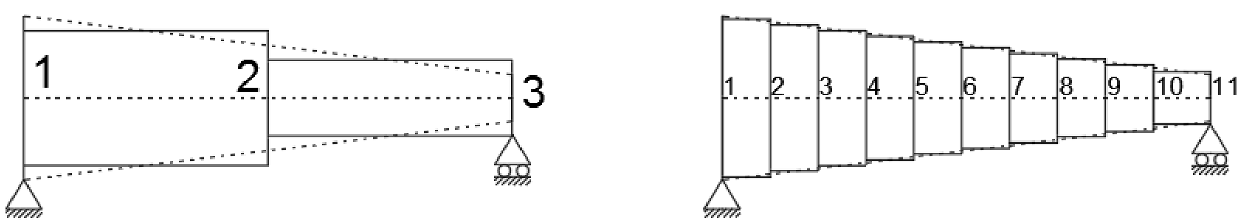


Figure 2 - Discretization of tapered bar in test problem 1

The exact value of the tip displacement can be found as follows:

$$\Delta_{h,B} = \int_0^L \frac{F}{EA(x)} dx = \frac{F}{E}\int_0^L \frac{dx}{A(x)} = \frac{F}{E}\int_0^L \frac{dx}{\pi \times (a - (a-b) \times {}^{x}/_{L})^2}$$

Substituting L = 350 cm, a = 4.7 cm, b = 2.0 cm, F = 600 kN, and E = 68 GPa, so:

$$\Delta_{h,B} = 1.0458\ cm$$

Table 1. Convergence of results for test problem 1

problem 1	
# elements	**Tip disp. (cm)**
2	0.99027
5	1.0358
10	1.0432
20	1.0451
30	1.0455
50	1.0457
exact	1.0458

As it is clear, as the number of elements increases, the tip displacement gets closer to the exact value. The results also verify the workability of the program.

Example Two – L-shaped beam structure

The second example is designed to check the ability of the code for 2D problems. It also evaluates the sensitivity of the element types on different beam lengths. The geometry of the problem is shown in Figure 3. In this example, three flexural elements, Bernoulli, Timoshenko with direct stiffness matrices and Timoshenko with linear shape function have been used and the vertical downward tip displacement of the node 3 has been compared with the exact result from the closed form solution.

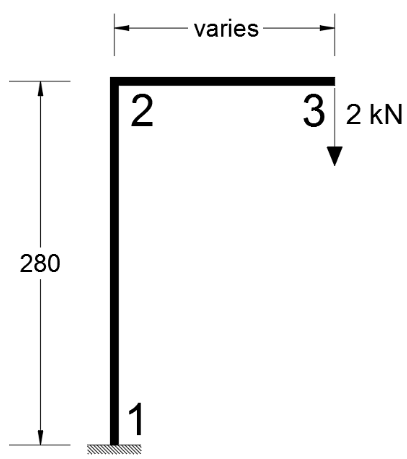


Figure 3 – Example Two

As mentioned before, if the Timoshenko element with linear shape function is used, the program automatically meshes the elements into short pieces. The length of mesh is a user defined parameter. In this example, five different lengths for pieces are considered and results are reported.

The exact answer can be calculated through the formulation below.

$$\Delta_{3,v} = \Delta_{2,axial\ in\ 12} + L \times \theta_3 + \Delta_{3,transverse\ in\ 23}$$

$$\Delta_{3,v} = \frac{FH}{EA} + L \times \frac{(FL) \times H}{EI} + \frac{FL^3}{3EI}$$

For three beam lengths, exact displacements are:

$L = 20\ cm$: $\Delta_{3,v}= 0.01639\ cm$

$L = 100\ cm$: $\Delta_{3,v}= 0.43565\ cm$

$L = 300\ cm$: $\Delta_{3,v}= 4.75047\ cm$

Table 2 shows the results for three different beam lengths and different element types.

Table 2 - Summary of results for test problem 2

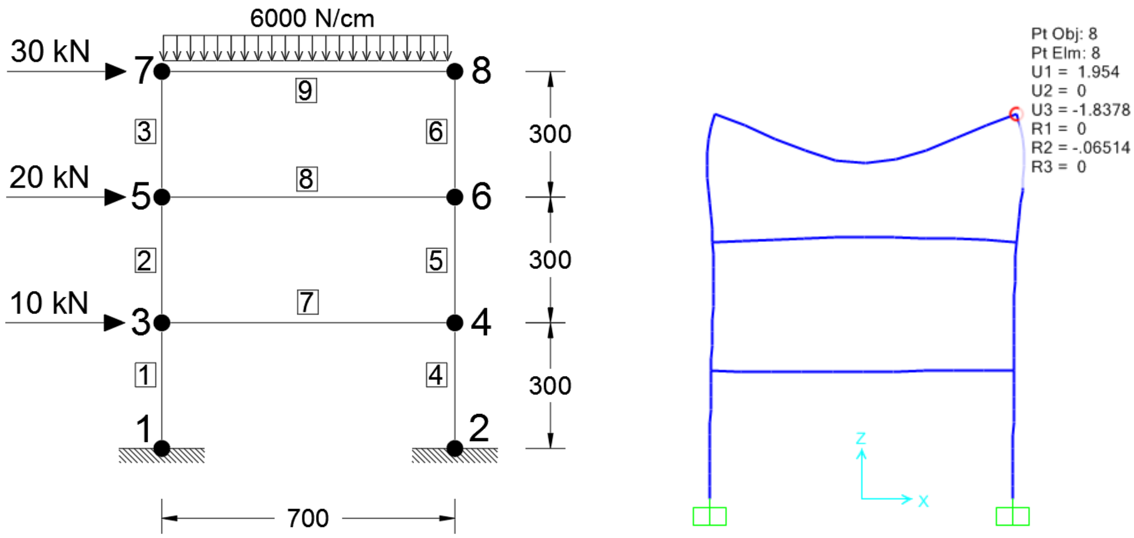
		Tip displacement (cm)		
		Beam Length		
		20 cm	100 cm	300 cm
Direct Stiffness	**Bernoulli**	0.016393	0.43565	4.7505
	Timoshenko	0.016497	0.43617	4.752
Timoshenko Linear Shape Function (mesh lengths =)	**50 cm**	0.003378	0.074106	0.78969
	20 cm	0.008947	0.23096	2.5141
	10 cm	0.013578	0.35685	3.887
	5 cm	0.015652	0.41321	4.5016
	2 cm	0.016355	0.43233	4.7101
Exact		0.01639	0.43565	4.75047

Results show that the Bernoulli element results are almost identical with the exact solution. The reason is indeed obvious. The formulation used for the closed form solution is only considering axial and flexural behavior and does not take the shear resistance into account. That implies that the formulation is in accordance with the Bernoulli element formulation. Interestingly, the Timoshenko element for all three beam lengths has given larger displacements, which is predictable, since this element is considering shear deformations in addition to the flexural and axial.

Comparing Timoshenko element with linear shape function against the exact results, indicates us that by increasing the number of elements (more refinement) or using shorter mesh lengths, results enhance. However an acceptable result can be found when the size of mesh is very small, which makes the model computationally expensive. To solve this problem, higher order shape functions can be a good idea.

Example Three - three story steel frame

As the third example, a three story steel frame with lateral point loads and vertical distributed load is investigated. In this example, like the last one, all three types of flexural members have been used and results are reported. Since this example is an indeterminate structure, reaction forces from the FE program are also controlled with the same reactions from a reliable FE commercial software. In this project, SAP2000 has been implemented to check results with.



(a) (b)

Figure 4 - (a) Example Three (b) SAP2000 Model deformed shape

For discretization, mesh sizes have been defines as 10cm, 20cm and 50cm. Note that the beam and column sections were two different types of W-sections and modeled as were assumed. Results are presented in Table 3.

Table 3 - Maximum horizontal displacement of the floors for test problem 3

		Node 8 displacements (cm)			Node 1 reaction (N,cm)		
		horizontal	**vertical**	**rotation**	**horizontal**	**vertical**	**rotation**
Direct Stiffness	**Bernoulli**	1.7994	1.8378	0.0615	6897.9	2058398.2	3089456.3
	Timoshenko	1.9540	1.8378	0.0651	6153.7	2058544.0	4574895.6
Timoshenko Linear Shape Function (mesh lengths =)	**50 cm**	1.6702	1.8378	0.0564	7792.3	2058621.1	4786696.0
	20 cm	1.9029	1.8378	0.0636	6414.1	2058557.6	4609250.2
	10 cm	1.9410	1.8378	0.0647	6218.7	2058547.5	4583500.5
Exact		1.9540	1.8378	0.0651	6153.7	2058544.0	4574896.0

Table 3 elaborates some interesting results. First, it is clear that the displacement and reaction results of the Timoshenko beam with direct stiffness matrix is identical with the SAP2000 results. This observation suggests that in SAP2000, Timoshenko stiffness matrix is the default element matrix for flexural member. The Bernoulli element results are, as expected, less than Timoshenko, but this time the difference is more distinctive. My guess is that since the model of three story building is a more complex model, its errors intensify as a result of the summation of many small errors (by error, the neglected factor of shear is meant).

The Timoshenko element with linear shape function is also working acceptably in this example. As the meshes get finer, the results are getting closer to SAP2000 results, which was our anticipation.

Example Four – truss beam

As an illustrative example for the discontinuity capturing feature of the program, a truss beam is modeled with different beam elements and results are compared. With this regard, an example from Hibbeler's Structural Analysis book has been studied. The truss beam consists of 13 elements with the area of 3.00 cm^2. In addition, the modulus of elasticity of the material is E = 200 GPa. Three nodal loads are applied on the lower chord of the beam. The problem statement with dimensions and boundary conditions is presented in Figure 5.

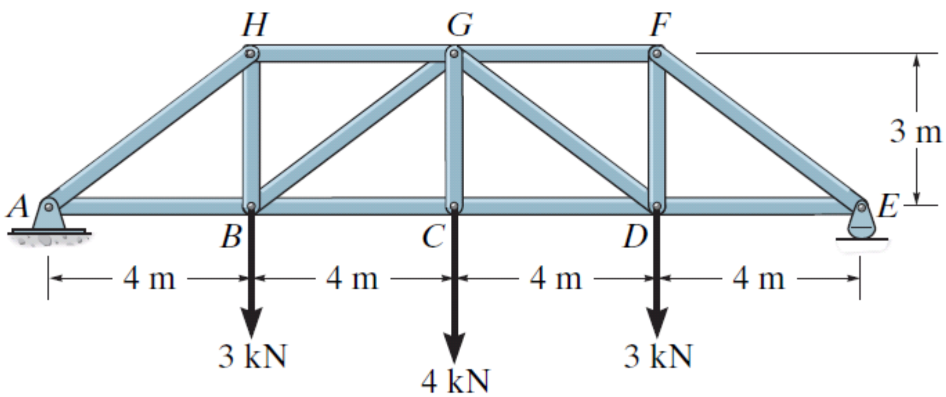


Figure 5 – Example Four

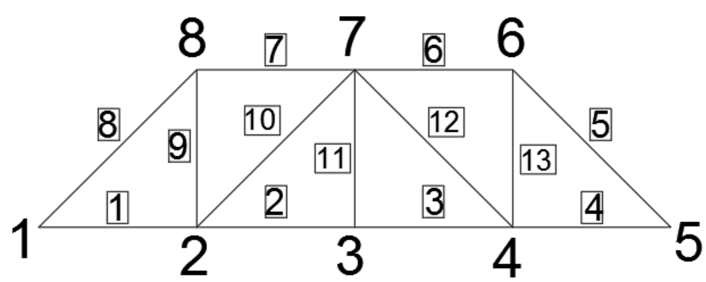


Figure 6. Nodal and element tags

If Bar element is used, the discontinuity feature does not show up itself, since the Bar element is resolved the discontinuity problems implicitly in the stiffness matrix. However, if flexural beam elements are used, the disconnection problems should be taken care of. All three types of elements have been implemented for this example and results are reported in the table below.

Table 4 – Results for different elements and SAP

	Node C displacement (cm)		Node E displacement (cm)
	disp 3 v	disp 3 h	
Bar	0.4915	0.1067	0.2133
Timoshenko	0.4916	0.1067	0.2134
Bernoulli	0.4916	0.1067	0.2134
Book	0.4910	-	-
SAP2000	0.4915	0.1067	0.2133

Since the example in the book has just been solved for the vertical displacement of the node C, for more comprehensive check of the program, a SAP model has been developed. In Figure 7, the deformed shape of the model is shown. Table 4 and Figure 7 have shown a very good compliance between the finite element program and reliable references (book and SAP2000). Note that since the chosen problem is simply supported, it is a determinate problem and so, reaction forces are not worth to show that much. However, in the next example, an indeterminate beam has been solved to check whether the program is successful to capture accurate reactions or not.

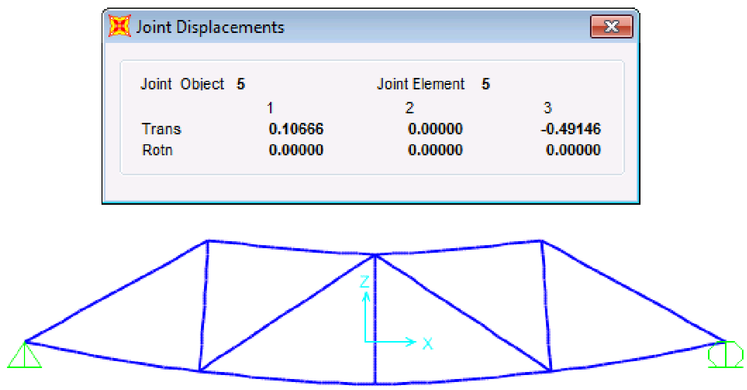


Figure 7. Deformed shape of the beam with node C displacements in cm

Example Five – indeterminate beam

As the last example, an indeterminate beam with internal disconnection and different types of loadings has modeled and results have been checked with the SAP2000 model results. The beam is assumed to be a W16x31 with E = 210 GPa and ν = 0.29. As the comparing parameters, displacements at the node 3 and 5 and also, reaction are considered. These values for different types of elements are reported in Table 5.

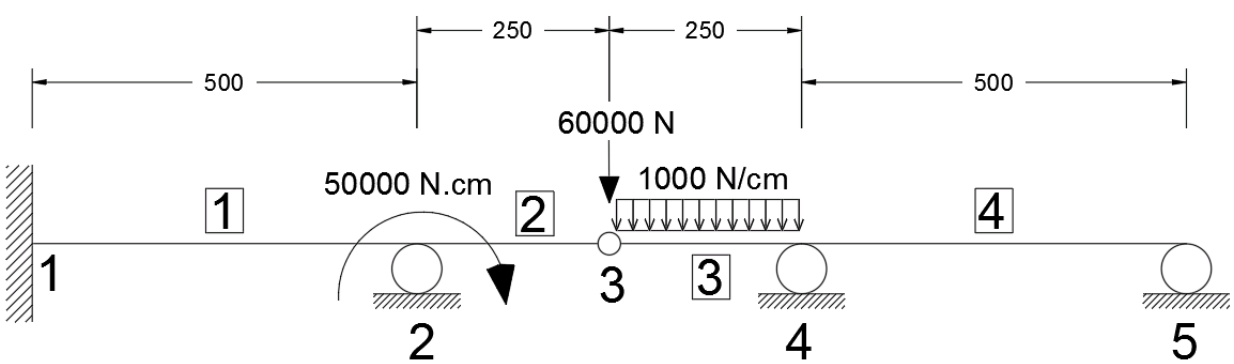


Figure 8 – Example Five

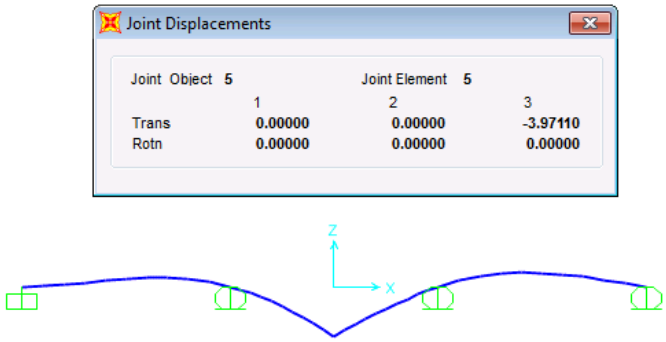


Figure 9 – deformed shape and displacements at node 3

Table 5 – Results for different elements and SAP

		vertical disp. @ 3	Reactions (N, cm)				
			Moment @ 1	**Vertical @ 1**	**Vertical @ 2**	**Vertical @ 4**	**Vertical @ 5**
Direct Stiffness	**Timoshenko**	3.977	10816547	-67205	137549	286485	-46828
	Bernoulli	3.801	11230769	-67385	128635	296227	-47477
Timoshenko Linear Shape Function (mesh lengths =)	**10 cm**	3.931	11573105	-71905	191091	234455	-43641
	5 cm	3.950	11522760	-71592	187528	237918	-43853
SAP		3.971	11225113	-69743	164129	260921	-45307

Despite good match between vertical displacements of the beam with different elements and SAP model, reaction forces are not as close as I expected. My guess is that since the nodes are not supposed to give exact forces (there are designed in a way to give exact displacements), the forces are not desirably close. Because as we know, the locations on an element if which the stresses are exact are not starting and ending nodes, but the integration points. Moreover, it is clear that as the fineness of beam increases, displacement results get closer to the exact value.

Example Sixth – support settlement

A short example is added here to demonstrate the nodal_disp matrix application, which is supposed to feature the effect of initial nodal displacements. An indeterminate beam as shown in the illustration below is not externally loaded, but the middle support settled for 1.00 cm. Reactions are compared between SAP2000 model and the program in Table 6. Note that the mechanical properties of the section and also material are the same as the fifth example.

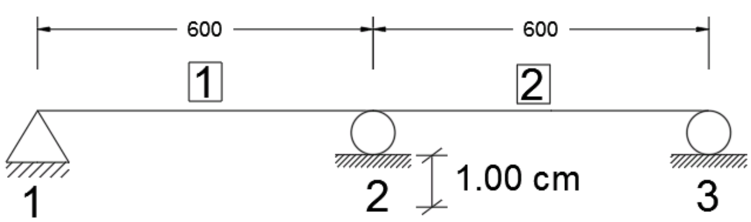


Figure 10. Example Sixth

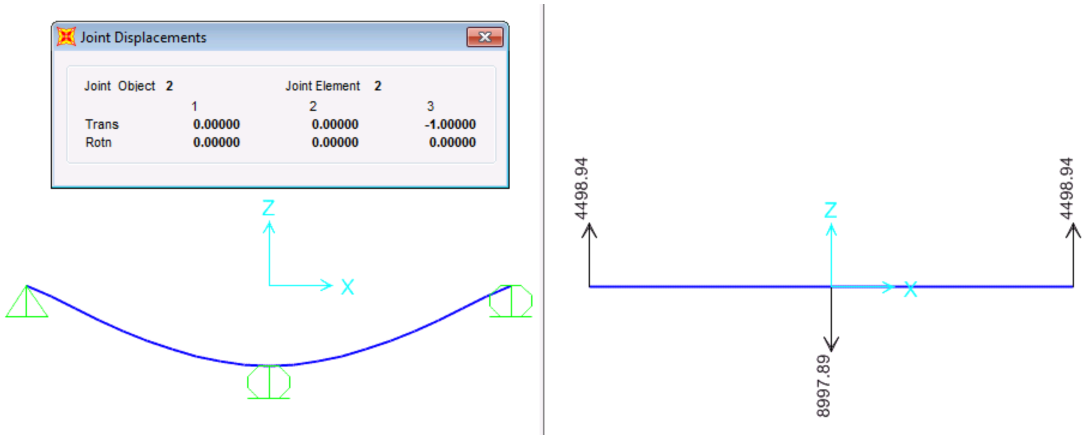


Figure 11. SAP results, displacement and reactions

As it is obvious from the Table 6, SAP2000 results are exactly the same as the program's result in the case of Timoshenko beam with direct stiffness. In case of Bernoulli beam, reactions are more than SAP2000 results. The reason for it is that since the stiffness matrix in Bernoulli beam does not have any shear terms, it gives stiffer matrix comparing to the Timoshenko beam. As a result, for a certain displacement (which is predefined by the settlement here), Bernoulli produces larger forces than the Timoshenko does.

As always, FE element with linear shape function shows more exact outputs as the mesh length decreases. It agrees with what we expect to see.

Table 6. Reaction results in different cases

		Reactions (N, cm)		
		Vertical @ 1	**Vertical @ 2**	**Vertical @ 3**
Direct Stiffness	**Timoshenko**	4498.94	8997.89	4498.94
	Bernoulli	4552.53	9105.06	4552.53
Timoshenko Linear Shape Function (mesh lengths =)	**10 cm**	4525.17	9050.35	4525.17
	5 cm	4505.50	9011.00	4505.50
SAP		4498.94	8997.89	4498.94

Conclusion

In this project, a Finite Element program was developed in the MATLAB software, based on the procedure explained in the class. The program gets inputs in different files and analyze them partially in numerous sub-functions and at the end, print out both inputs and outputs. The program still can improve considerably by adding extra features, such as higher order elements or a simplified graphical interface.

To verify sanity of the program, six examples have been done by it and compared with the benchmarks. It was tried to demonstrate extra features embedded in the program in each of these examples, e.g. automeshing, nodal displacements and internal disconnections. In general, the program was successful to analyze these illustrative examples. Some of the significant conclusions is listed hereafter:

- The more an element is refined, the better results we can expect in case of using Finite Elements with shape functions. However, the rate of enhancement in performance is not very desirably, encouraging us to use higher order shape functions.
- In SAP2000 commercial software, the default flexural element is the Timoshenko element with the direct stiffness matrix.
- Comparing two types of elements with direct stiffness matrices, it was deduced that the Timoshenko element always shows greater displacements and mostly less reaction forces. The reason is also discussed in the context.