

1. Structure

This program was revised based on the beam-column program. After the modification, it can solve any 2-D structure assembled by any number of rectangles (Figure 1). In this revised program, lots of effort was put on the preprocessor modification, such as structural assembly and mesh generator. The 4-node rectangular C^0 element was used in this FEA program. In this program, it is assumed that the applied loads are only applied at the nodes. In other words, this program does not consider consistent nodal loads due to body or surface loads.

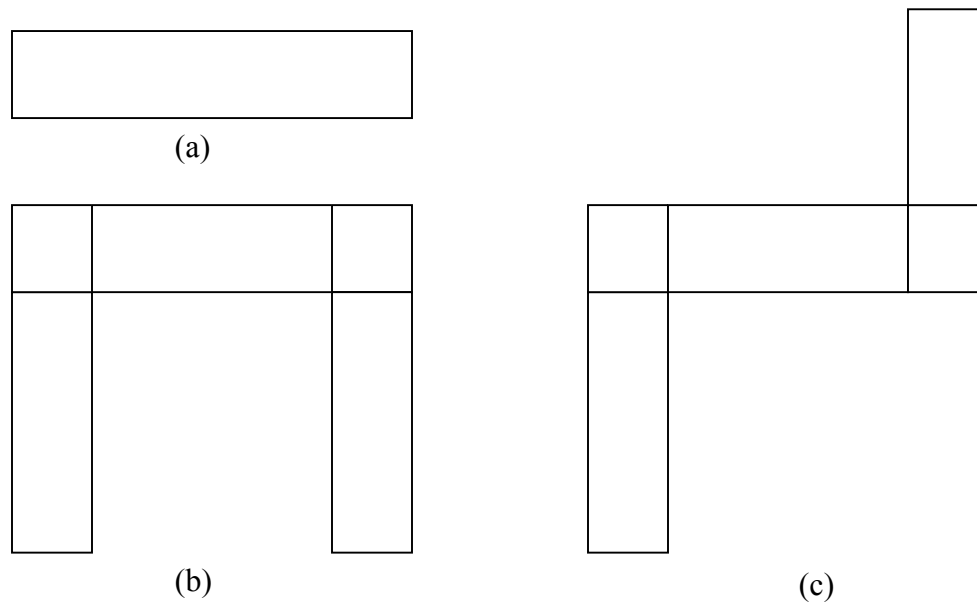


Figure 1: Simulation of a 2-D (a) beam, (b) frame, and (c) arbitrary shape assembled by rectangles

The program has the following structure:

- Clear all variables
- Begin preprocessor
- Initialize the main variables
- Loop over all elements
- get nodal coordinates
- calculate element stiffness matrix

- partition the global stiffness matrix and assemble
- create global K matrix
- End loop over elements
- Partition stiffness matrix into $K_{\text{fixed, fixed}}$ and $K_{\text{free, free}}$
- Calculate the load vector
- Initialize the displacement vector
- Calculate the global displacements
- Calculate the reactions
- Compute nodal force
- Ask to display the displacements
- Ask to display the nodal force
- Begin post-processor

2. Preprocessor

In the preprocessor, it allows user to assemble any number of rectangles to form a structure and gives the degree of freedom to assign different material properties (Young's modulus, E , and Poisson's ratio, ν), element types (plan stress and plan strain), and thickness (t) in different rectangle (however, elements within the same rectangle have the same properties). Another feature within this program is the mesh generator. This function allows user to create specific mesh in each rectangle. The preprocessor has the following structure:

- Enter the 4 corner coordinates for the first rectangle
- Assign material properties, element type, and thickness for the elements within the first rectangle
- Mesh generator for the first rectangle (Assign meshing density along horizontal and vertical direction)
- Plot the meshing result for the first rectangle
- Insert other rectangles
- Assign the boundary conditions
- Plot the result of support (red circle)

- Enter the nodal loads
- Plot the result of nodal loads (black arrow)

Remarks:

In plotting the “Meshing result for the first rectangle”, nodal number is also plotted aside each node (Figure 2). This information makes it easier to assign the boundary conditions and nodal loads in the following procedure.

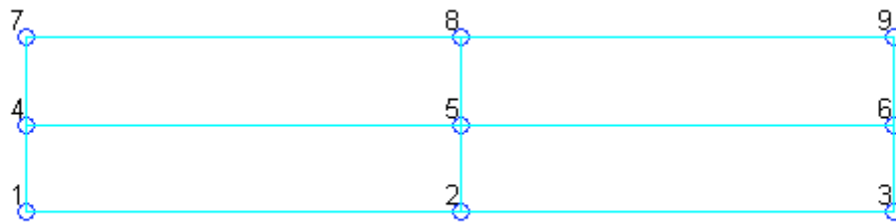


Figure 2: Plotting the meshing result for the first rectangle

The algorithm in the “Insert another rectangle” can be shown as follow,

- Loop
- assign an old rectangle which is the base for assembling the new rectangle
- assign the assembling direction (L for left, R for right, U for up, and D for down)
- assign material properties for elements within this new rectangle
- assign dimensions for the new rectangle
- mesh generator
- plot assembling result
- End loop

Figure 3 to 6 show the ability of this preprocessor by displaying the assembling process of a frame-like structure.

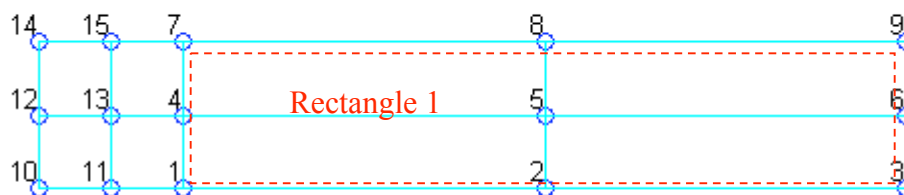


Figure 3: Plotting the assembling result I (Assembly a new rectangle to the left side of rectangle 1)

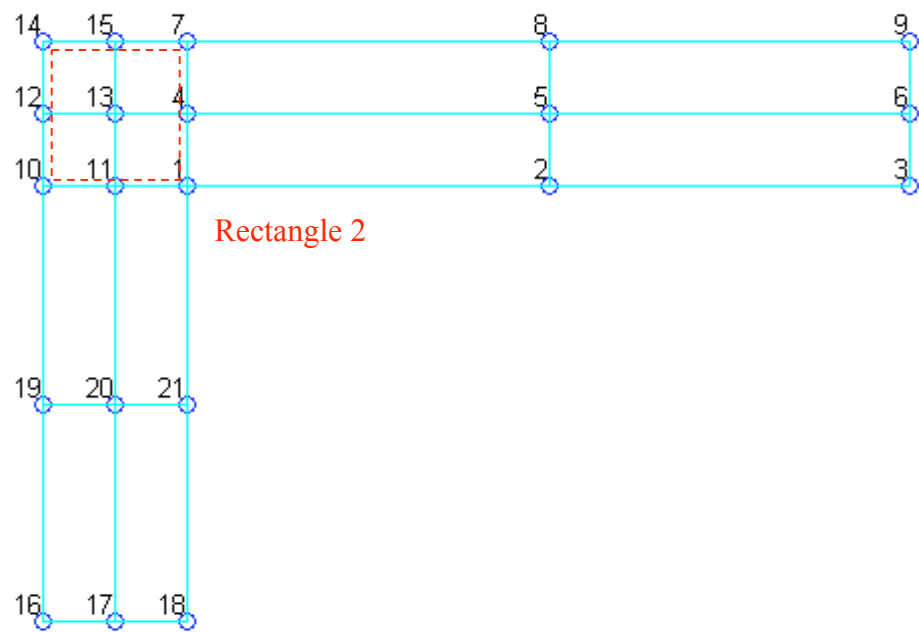


Figure 4: Plotting the assembling result II (Assembly a new rectangle to the down side of rectangle 2)

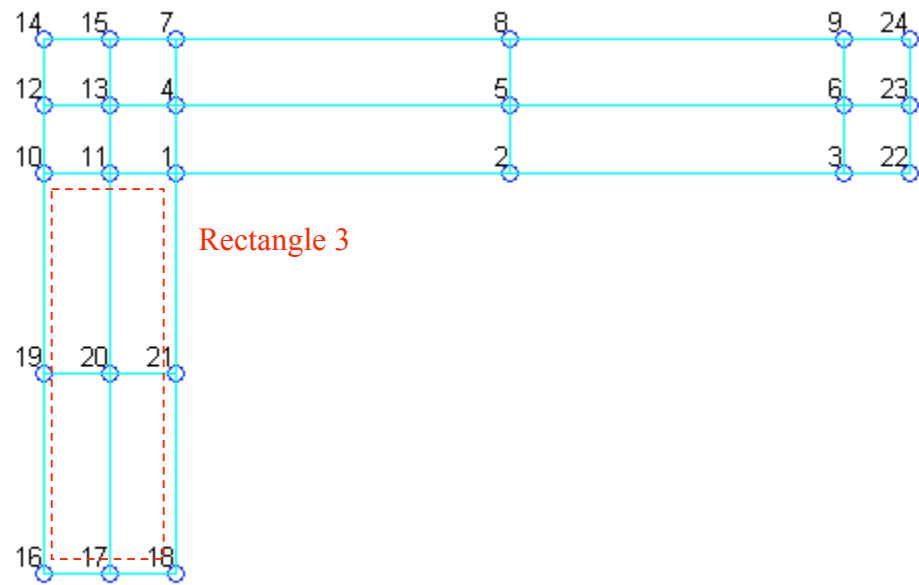


Figure 5: Plotting the assembling result III (Assembly a new rectangle to the right side of rectangle 1)

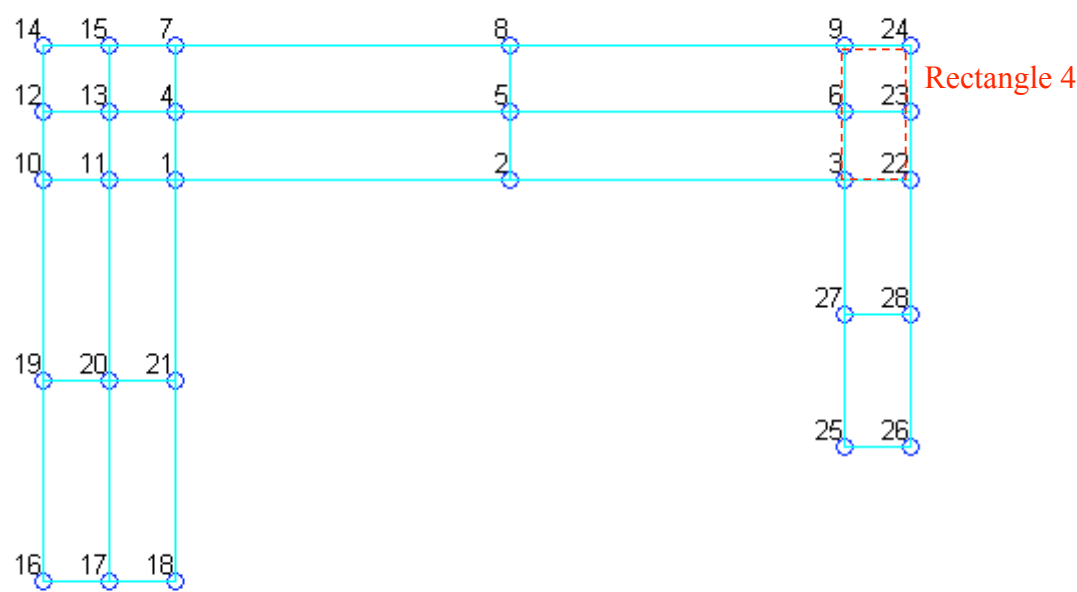


Figure 6: Plotting the assembling result IV (Assembly a new rectangle to the down side of rectangle 4)

Again, it is easier to assign the boundary conditions and nodal loads by showing the meshing result along with the nodal number aside. Figure 7 and 8 show the result after assigning the support information and loading conditions.

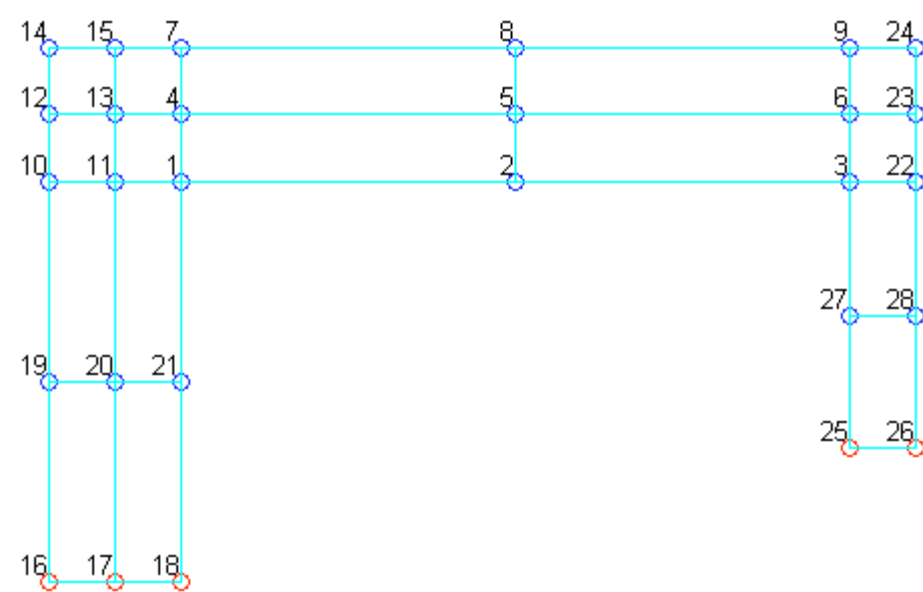


Figure 7: Plotting the support (red circle)

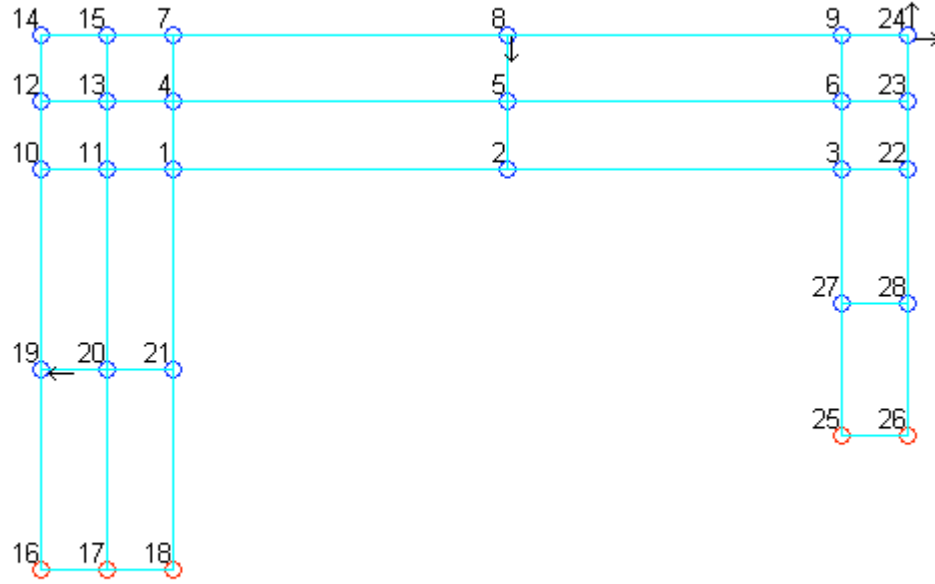


Figure 8: Plotting the nodal load (black arrows)

3. Stiffness Matrix for the 4-Node Rectangular C^0 Element

As shown in figure 9, a 4-node rectangular C^0 element has 8 degree of freedoms. The displacement field within the element can be interpolated by the shape function as follow.

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} \quad (1)$$

where

$$N_1 = \frac{(a-x)(b-y)}{4ab}$$

$$N_2 = \frac{(a+x)(b-y)}{4ab}$$

$$N_3 = \frac{(a+x)(b+y)}{4ab}$$

$$N_4 = \frac{(a-x)(b+y)}{4ab}$$

The stiffness matrix can be calculated through

$$[k]_{8 \times 8} = \int_{-b-a}^b \int_{-a}^a [B]^T [E] [B] t \, dx \, dy \quad (2)$$

where

$$[B]_{8 \times 8} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix}$$

$$[E]_{3 \times 3} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad \text{for plane stress}$$

$$[E]_{3 \times 3} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad \text{for plane strain}$$

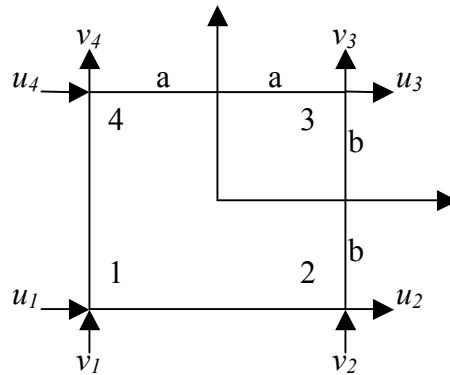


Figure 9: A 4-node rectangular C^0 element

4. Check for Correction

To check the correction of this program, a square plate (2 inch in thickness) stretched by a uniform force was analyzed (Figure 10). It is expected that the convergence problem in this simplest loading configuration is limited. Hence, this configuration was computed using FEA by only one 4-node rectangular C^0 element to check the correction of this FEA program. The simulated geometry, meshing, and boundary and loading conditions are shown in figure 11. The support conditions are fixed in x- and y-directions in node 1 and fixed in x-direction and free in y-direction in node 3.

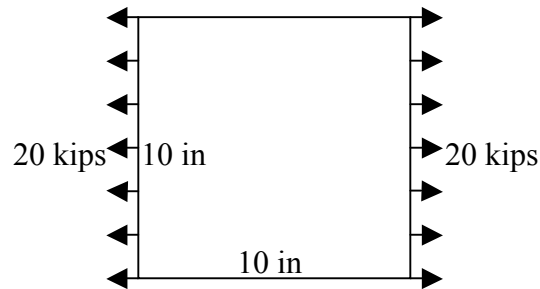


Figure 10: A square plate (2 inch in thickness) stretched by a uniform force

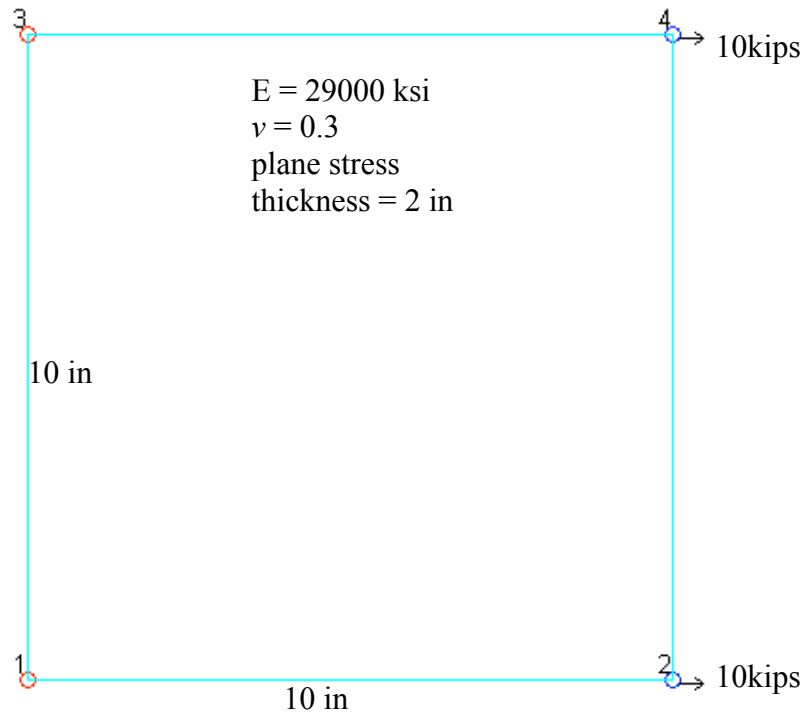


Figure 11: The simulated geometry, meshing, and boundary and loading conditions

Table 1 and 2 give the comparison between computational and theoretical results. As can be seen, the results are almost identical except some truncation error. Figure 12 shows the deformed shape (dot line, 2000 times) of this problem.

Table 1: Reactions

Node #	x-direction (kips)		y-direction (kips)	
	Computational	Theoretical	Computational	Theoretical
1	-10	-10	6.66134e-016	0
3	-10	-10	0	0

Table 2: Nodal displacements

Node #	dx (in)		dy (in)	
	Computational	Theoretical	Computational	Theoretical
1	0	0	0	0
2	0.000344828	0.000344828	1.09843e-020	0
3	0	0	-0.000103448	-0.00010448
4	0.000344828	0.000344828	-0.000103448	-0.00010448

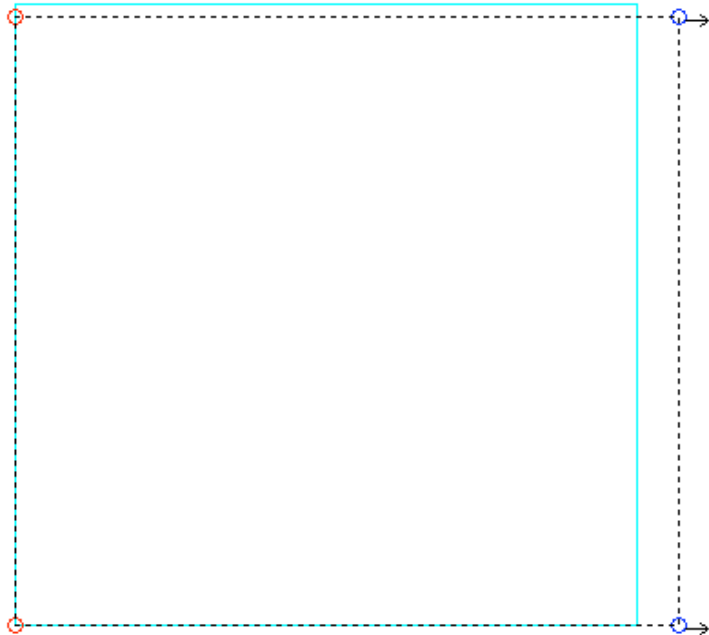


Figure 12: Comparison between the original shape, solid line, and the deformed shape, dot line, 2000 times)

(2000X) dot line

5. Convergence study

A cantilever beam problem was conducted in this section to do the convergence studies. This cantilever beam problem is identical with homework 2 and 3 (Figure 13) with 35 inches in length, 10 inches in depth, and 2 inches in thickness. The material is isotropic and linear elastic. Poisson's ratio is 0.3. Enforce plane stress condition.

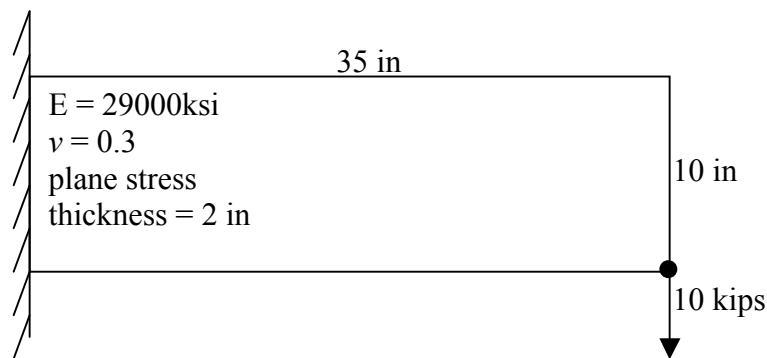


Figure 13: Cantilever beam

Table 3 shows the computational result of nodal displacement in the y-direction for the right-bottom corner (dy of the black dot in figure 13) calculated with different number of elements. This result is compared to the theoretical values and the value calculated from ANSYS (The element type used in ANSYS analysis was chosen to be the 8Node-82 quad element).

Two theoretical theories were used to calculate the theo. deformation and give comparisons with computational results. The first one is the Classical beam theory, in which the shear deformation is not considered.

$$dy = \frac{PL^3}{3EI}$$

The second theoretical calculation considered the shear deformation in the cantilever beam; however, it still assumes that plan remains plan in this calculation. This consideration is probably important for this case because the Depth to Length ratio is 3.5,

which is kind of a deep beam rather a slim one.

$$dy = \frac{PL^3}{3EI} + \frac{PL}{AG}$$

Table 3: Convergence study comparison between computation and theory

# of elements		dy (in)			
Hor. by Vert.	Total	FEA	ANSYS	Theo. (Euler)	Theo. (Shear)
1*1	1	0.00542145		0.029568966	0.031090705
2*1	2	0.0138212			
4*1	4	0.0225934			
4*2	8	0.0235768			
5*2	10		0.030946		
8*2	16	0.0284786			
8*4	32	0.029124			
10*4	40		0.031161		
16*4	64	0.0309248			
16*8	128	0.0311851			
20*8	160		0.031186		

Figure 14 plots the results in table 3. As can be seen, the computational results (both FEA program and ANSYS) converge to the theoretical value (shear consideration) as more and more elements were used. Second, the convergence speed of deformation is faster in ANSYS than in FEA program. This result can be attributed to different element type used in FEA program (4-Node rectangular C^0 element) and ANSYS (8Node-82 quad element). Third, shear consideration is necessary in this analysis (deep beam). Forth, a careful examination of the FEA result shows that the convergence speed of deformation is faster when more elements are created along the horizontal direction than along vertical direction. For example, dy increases from 0.022 to 0.023 when mesh number increases from 4*1 to 4*2. However, dy increases from 0.023 to 0.028 when mesh number increases from 4*2 to 8*2. This is because the geometry, boundary, and loading configuration make the deformation more sensitive to the meshing density along horizontal direction. Figure 15 displays the sequence of deformation with different number of elements (100 times magnitude).

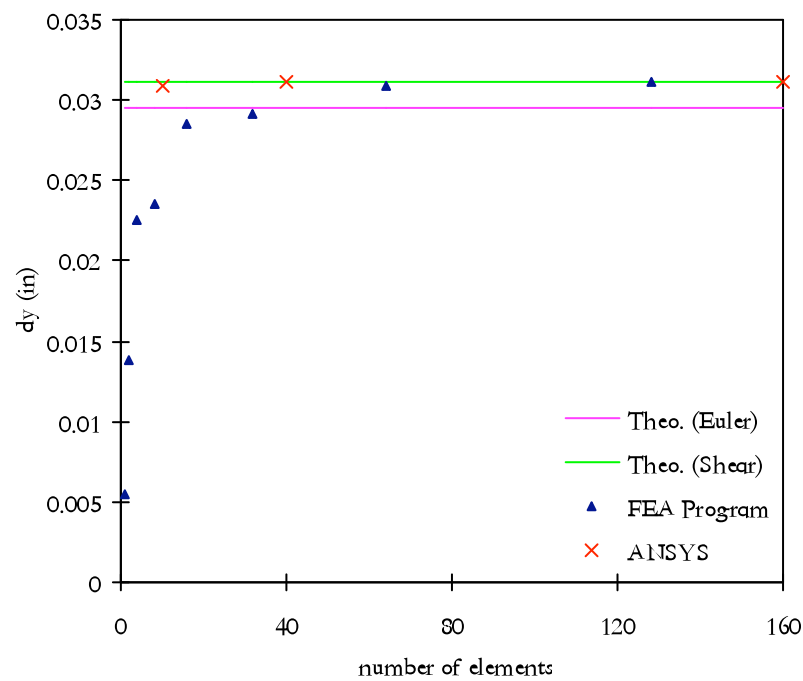
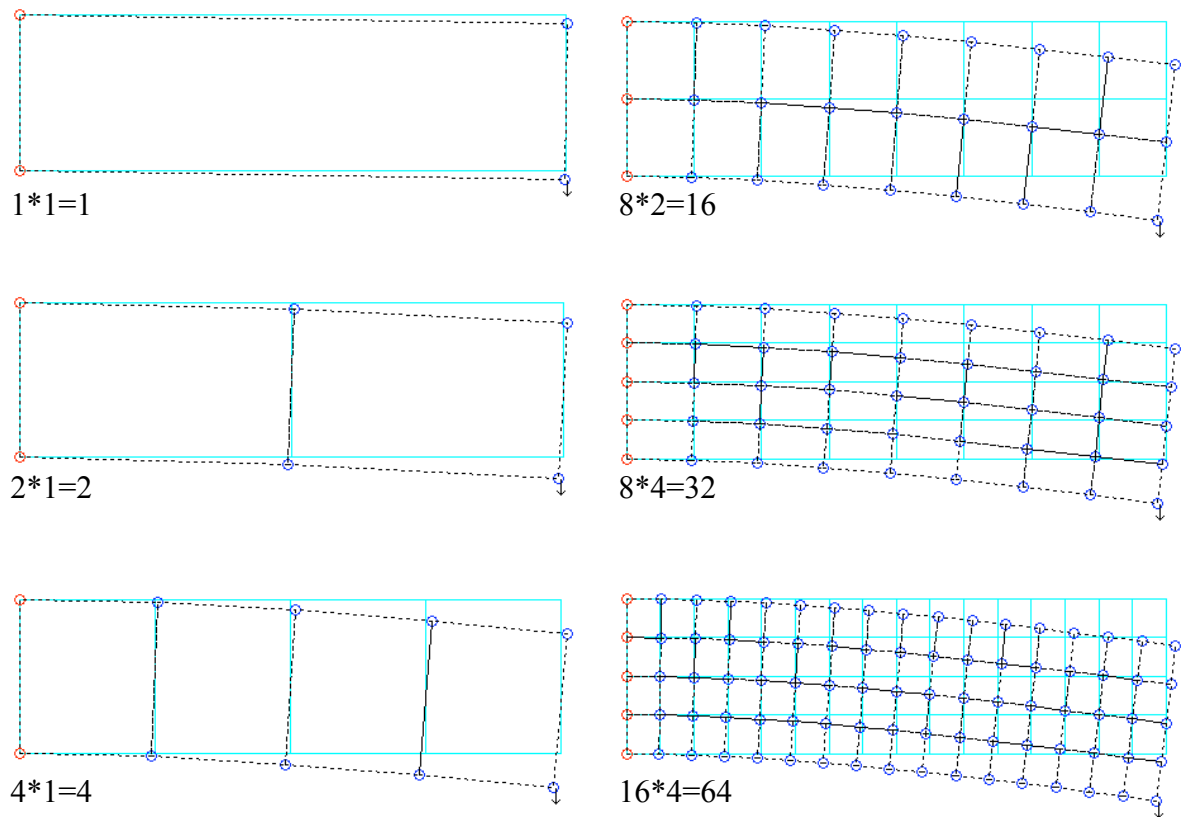


Figure 14: Convergence study and comparison between computational value and theoretical one



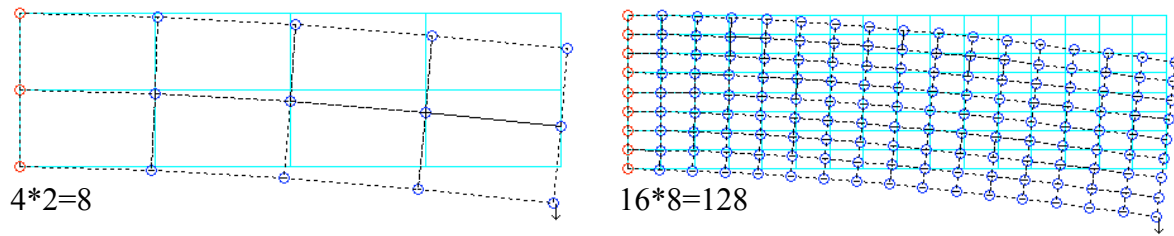
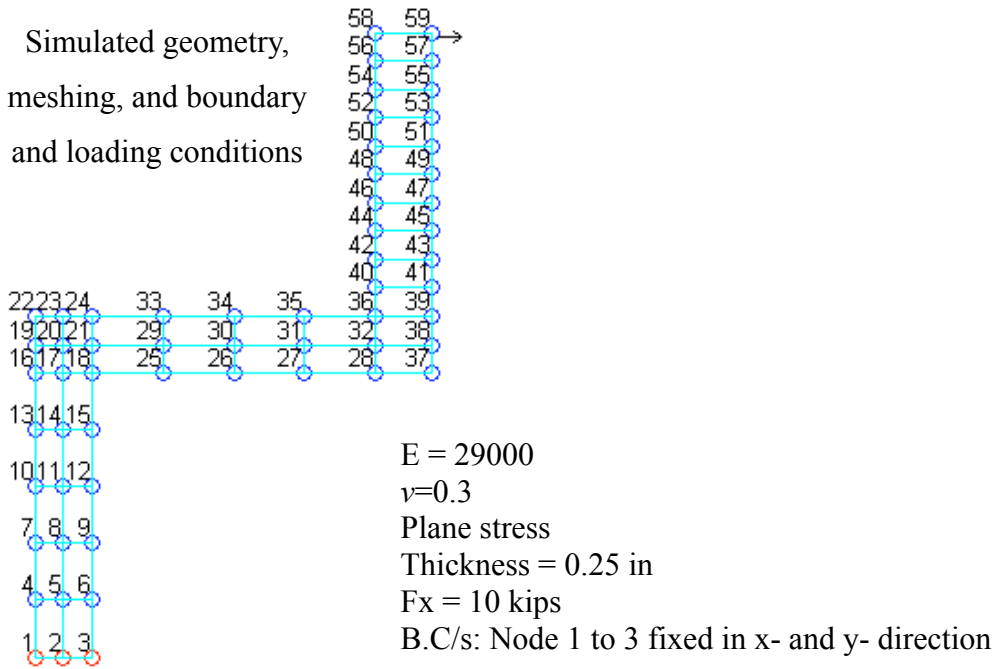
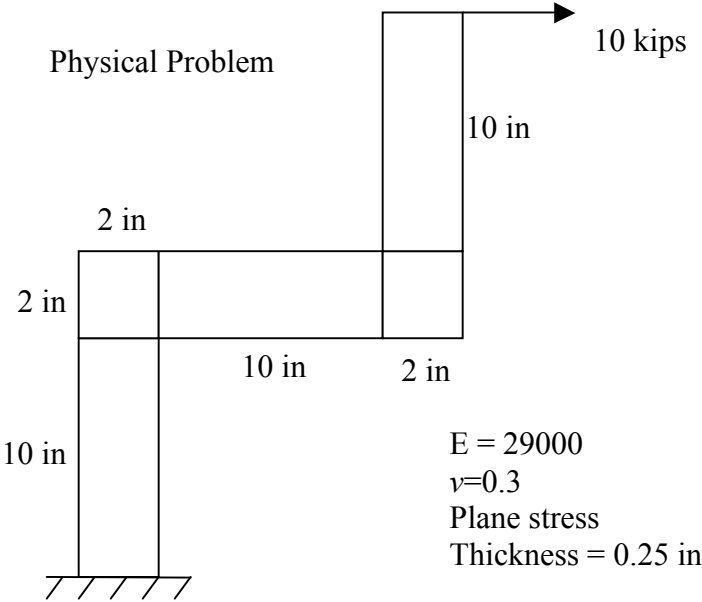


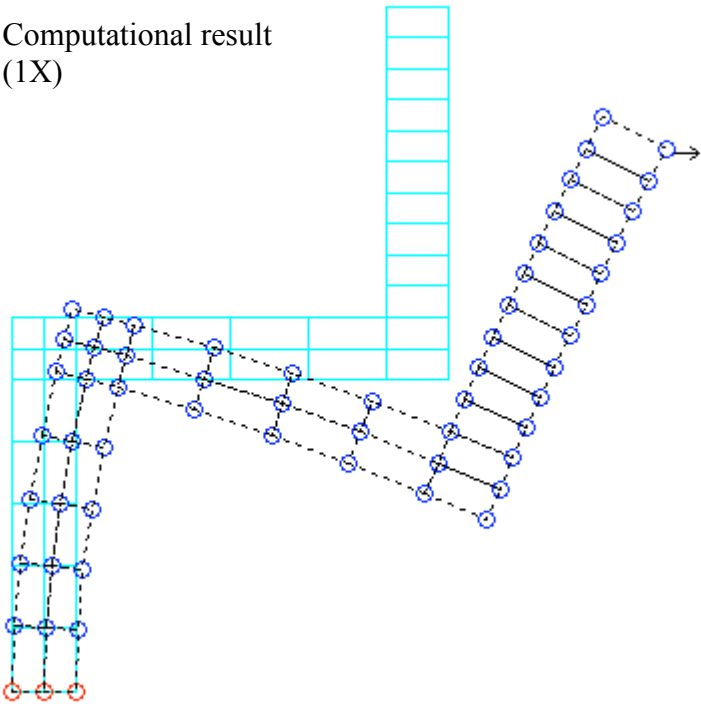
Figure 15: Sequence of deformation with different number of elements calculated by FEA program (100X).

6. Computational examples

After checking the correction (section 4) and the convergence study (section 5), we have more confidence to use this FEA program. In this section, two complicated structures were computed to demonstrate the ability of this FEA program.

Case 1: Arbitrary shape with simple loading and boundary conditions





Case 2: Frame with different material properties and boundary conditions

Physical Problem

