Xingchen Liu

Goldy Kumar

EMA 605, Homework Assignment #8

Program Description:

The program has been written in *python.* The class is named CST which can solve plane stress and plane stress problems for some assembly of Constant Triangle elements.

The class variables are (can be identified inside the class as self.*variable _name)*:

1. **E** - Young's Modulus
2. **v** – Poisson Ratio
3. **iflag**- Boolean for identifying plane stress or strain
4. **xy**- list containing x,y – coordinates of the nodes in the order of indices given in input
5. **element**- matrix with nodes for each element.
6. **displacement:** Displacement vector
7. **n_nodes** – number of nodes
8. **n-element** – number of elements
9. **k_global** - global stiffness matrix. Initialized as a zero matrix with 2***n_nodes** rows and columns.
10. **E_matrix** - Stores constitutive matrix

Class member functions (explains the program steps and calculation in detail):

1. **Constitutive_matrix(self):** Generates E matrix for plane stress or plane strain problem. The formula is defined explicitly for both the cases. This uses constants **E** and **v**
2. **B(self,el):** Takes input every triangle element node and generates Strain Matrix for a Constant Strain Triangle element, assuming first node lies at origin. This is also done using explicit formula. The formula uses element nodal- coordinates.
3. **Global_Stiffness(self):** Generates Global Stiffness Matrix for the plane structure-
    a. Iterate over each element, say element ***i***
        i. Get **B** matrix and **J** for element ***i***
        ii. Compute **local- K** matrix using the relation 0.5*$|\mathbf{J}|\mathbf{B}^T \mathbf{E}\ \mathbf{B}$**.** This is a 6x6 matrix.
        iii. Form a **Temp_ global_K** matrix with 2***n_nodes** rows and columns and assign **local- K** to the right global node.
        iv. Add **Temp_ global_K** to the **k_global**

4. **Strain_stress(self,el):** Finds stress and strain in an element **el**. Strain is found by matrix multiplication of **strain matrix B** and **displacement** vector. Stress is found by matrix multiplication of **E** and **strain.**
5. **Global_load(self):** Computes global load vector by matrix multiplication of **k_global** matrix and **displacement** vector.
6. The **main** function – (***python*** supports in class testing)
    a. Reads the input from a file – nodal coordinates, displacement vector, element nodes
    b. Forms an object of the class – **testcase**.
    c. Calls **Global_Stifnness** function to compute global K matrix. Prints the non zero values in required format.
    d. Calls **Strain_stress** function for each element and prints stress and strain vector for each CST element.
    e. Calls **Global_load** function and prints it.

<u>Test cases and Comparison:</u>

Test case 1:

Let $v = -0.03, u = -\frac{v}{1-v}v = 0.01$

Input:

```
4 2

10 .25 1

2 1.0 0.0
1 0.0 0.0
4 0.0 1.0
3 1.0 1.0

1 2 3
1 3 4

1 0.0 0.0
3 0.01 -0.03
4 0.0 -0.03
2 0.01 0.0
```

Output:

```
Nonzero entries in the global stiffness matrix
( 1 , 1 , 8.0 )
( 1 , 3 , -6.0 )
( 1 , 4 , 2.0 )
( 1 , 6 , -4.0 )
( 1 , 7 , -2.0 )
( 1 , 8 , 2.0 )
( 2 , 2 , 8.0 )
( 2 , 3 , 2.0 )
( 2 , 4 , -2.0 )
( 2 , 5 , -4.0 )
( 2 , 7 , 2.0 )
( 2 , 8 , -6.0 )
( 3 , 1 , -6.0 )
( 3 , 2 , 2.0 )
( 3 , 3 , 8.0 )
( 3 , 4 , -4.0 )
( 3 , 5 , -2.0 )
( 3 , 6 , 2.0 )
( 4 , 1 , 2.0 )
( 4 , 2 , -2.0 )
( 4 , 3 , -4.0 )
( 4 , 4 , 8.0 )
( 4 , 5 , 2.0 )
( 4 , 6 , -6.0 )
( 5 , 2 , -4.0 )
( 5 , 3 , -2.0 )
( 5 , 4 , 2.0 )
( 5 , 5 , 8.0 )
( 5 , 7 , -6.0 )
( 5 , 8 , 2.0 )
( 6 , 1 , -4.0 )
( 6 , 3 , 2.0 )
( 6 , 4 , -6.0 )
( 6 , 6 , 8.0 )
( 6 , 7 , 2.0 )
( 6 , 8 , -2.0 )
( 7 , 1 , -2.0 )
( 7 , 2 , 2.0 )
( 7 , 5 , -6.0 )
( 7 , 6 , 2.0 )
( 7 , 7 , 8.0 )
( 7 , 8 , -4.0 )
( 8 , 1 , 2.0 )
( 8 , 2 , -6.0 )
( 8 , 5 , 2.0 )
```

(8 , 6 , -2.0)
(8 , 7 , -4.0)
(8 , 8 , 8.0)

Stress and Strain in each element
element 1 : strain [0.01 -0.03 0.] ; stress [0. -0.32 0.]
element 2 : strain [0.01 -0.03 0.] ; stress [0. -0.32 0.]

Global Load Vector
[0. 0.16 0. 0.16 0. -0.16 0. -0.16]

Global Stiffness Matrix from notes

$$\underline{\underline{K}} = \frac{c}{2}\begin{pmatrix} \frac{3}{2}-2\nu & \nu-\frac{1}{2} & \nu & 0 & 0 \\ \nu-\frac{1}{2} & \frac{3}{2}-2\nu & 0 & \nu-1 & \nu \\ \nu & 0 & \frac{3}{2}-2\nu & \frac{1}{2}-\nu & \nu-\frac{1}{2} \\ 0 & \nu-1 & \frac{1}{2}-\nu & \frac{3}{2}-2\nu & -\frac{1}{2} \\ 0 & \nu & \nu-\frac{1}{2} & -\frac{1}{2} & \frac{3}{2}-2\nu \end{pmatrix}$$

Global Stiffness Matrix

8	0	-6	2	0	-4	-2	2
0	8	2	-2	-4	0	2	-6
-6	2	8	-4	-2	2	0	0
2	-2	-4	8	2	-6	0	0
0	-4	-2	2	8	0	-6	2
-4	0	2	-6	0	8	2	-2
-2	2	0	0	-6	2	8	-4
2	-6	0	0	2	-2	-4	8

Load vector from notes

$$\underline{R}^T = \begin{pmatrix} 0 & 0 & \sigma_y/2 & 0 & \sigma_y/2 \end{pmatrix}$$

Load vector

$$[\,0. \quad 0.16 \quad 0. \quad 0.16 \quad 0. \quad -0.16 \quad 0. \quad -0.16]^T$$

In this test case, we reach the same relationship of stiffness matrix and load vector compare to notes.

Test case 2:

$$u_1 = u(0,0) = 0, v_1 = v(0,0) = 0$$

$$u_2 = u(1,0) = 0, v_2 = v(1,0) = 0$$

$$u_3 = u(1,1) = -\frac{\pi}{2}, v_3 = v(1,1) = \frac{\pi}{2}$$

$$u_4 = u(0,1) = 0, v_4 = v(0,1) = 0$$

Input:

```
4 2

10 .25 1

2 1.0 0.0
1 0.0 0.0
4 0.0 1.0
3 1.0 1.0

1 2 3
1 3 4

1 0.0 0.0
3 -1.57079633 1.57079633
4 0.0 0.0
2 0.0 0.0
```

Output:

```
Nonzero entries in the global stiffness matrix
( 1 , 1 , 8.0 )
( 1 , 3 , -6.0 )
( 1 , 4 , 2.0 )
( 1 , 6 , -4.0 )
( 1 , 7 , -2.0 )
( 1 , 8 , 2.0 )
( 2 , 2 , 8.0 )
( 2 , 3 , 2.0 )
( 2 , 4 , -2.0 )
( 2 , 5 , -4.0 )
( 2 , 7 , 2.0 )
( 2 , 8 , -6.0 )
( 3 , 1 , -6.0 )
( 3 , 2 , 2.0 )
( 3 , 3 , 8.0 )
```

```
( 3 , 4 , -4.0 )
( 3 , 5 , -2.0 )
( 3 , 6 , 2.0 )
( 4 , 1 , 2.0 )
( 4 , 2 , -2.0 )
( 4 , 3 , -4.0 )
( 4 , 4 , 8.0 )
( 4 , 5 , 2.0 )
( 4 , 6 , -6.0 )
( 5 , 2 , -4.0 )
( 5 , 3 , -2.0 )
( 5 , 4 , 2.0 )
( 5 , 5 , 8.0 )
( 5 , 7 , -6.0 )
( 5 , 8 , 2.0 )
( 6 , 1 , -4.0 )
( 6 , 3 , 2.0 )
( 6 , 4 , -6.0 )
( 6 , 6 , 8.0 )
( 6 , 7 , 2.0 )
( 6 , 8 , -2.0 )
( 7 , 1 , -2.0 )
( 7 , 2 , 2.0 )
( 7 , 5 , -6.0 )
( 7 , 6 , 2.0 )
( 7 , 7 , 8.0 )
( 7 , 8 , -4.0 )
( 8 , 1 , 2.0 )
( 8 , 2 , -6.0 )
( 8 , 5 , 2.0 )
( 8 , 6 , -2.0 )
( 8 , 7 , -4.0 )
( 8 , 8 , 8.0 )

Stress and Strain in each element
element  1 : strain [ 0.          1.57079633 -1.57079633] ; stress  [  6.28318532  18.84955596  -
6.28318532]
element  2 : strain [-1.57079633  0.          1.57079633] ; stress  [-18.84955596  -6.28318532
6.28318532]

Global Load Vector
[ -6.28318532   6.28318532   6.28318532 -12.56637064 -12.56637064
  12.56637064  12.56637064  -6.28318532]
```

Global Stiffness Matrix from notes

$$\frac{c}{2}\begin{pmatrix} \frac{3}{2}-2\nu & 0 \\ 0 & \frac{3}{2}-2\nu \end{pmatrix}$$

Global Stiffness Matrix

8	0	-6	2	0	-4	-2	2
0	8	2	-2	-4	0	2	-6
-6	2	8	-4	-2	2	0	0
2	-2	-4	8	2	-6	0	0
0	-4	-2	2	8	0	-6	2
-4	0	2	-6	0	8	2	-2
-2	2	0	0	-6	2	8	-4
2	-6	0	0	2	-2	-4	8

In this test case, we reach the same result of stiffness matrix compare to notes.