Implementation of 2D stress-strain Finite Element Modeling on MATLAB

Xingzhou Tu

Abstract: This report give a basic idea of how finite element modeling works and how can it be implemented on MATLAB. Firstly, the basic concept of stress tensor and strain tensor are introduced. Secondly, the principles of finite element modeling are discussed. Thirdly, I will show a finite element modeling example of cantilever beam on MATLAB. The simulation result matches theory prediction perfectly.

1. Stress Tensor and Strain Tensor

The state of stress at a point inside a material can be completely described by a 3 by 3 tensor. The tensor relates the stress vector T across a surface to the normal vector n of that surface by following equations:

$$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}$$

That tensor is so called stress tensor.

Similarly, the state of strain, or the state of deformation at a point inside a material can be also completely described by a 3 by 3 tensor. The strain tensor can be calculated as:

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial r_j} + \frac{\partial u_j}{\partial r_i}\right)$$

The vector $\vec{u}(\vec{r})$ represents the displacement of point $\vec{r} = (x, y, z)$ after deformation.

In the 2D plane stress case, there is no stress along z-direction. The stress and strain can be simplified as vectors:

$$\vec{\varepsilon}(x,y) = \left[\varepsilon_{xx}, \varepsilon_{yy}, 2\varepsilon_{xy}\right]^T = \begin{bmatrix} \partial/\partial x & 0 \\ 0 & \partial/\partial y \\ \partial/\partial y & \partial/\partial x \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}$$

$$\vec{\sigma}(x,y) = [\sigma_{xx}, \sigma_{yy}, \sigma_{xy}]^T$$

The stress vector and strain vector will obey following relationships:

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \frac{E}{1-v^2} \begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & \frac{1-v}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{bmatrix}$$

The E is the Young's modulus, v is the Poisson ratio.

To maintain static equilibrium, the force due to the stress will be equal to the external body force. Considering an infinite small element, the total force from stress tensor can be calculated by differentiating the stress field, therefore we have following equations:

$$\begin{bmatrix} \partial/\partial x & 0 & \partial/\partial y \\ 0 & \partial/\partial y & \partial/\partial x \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \end{bmatrix} = 0$$

The vector b is the body force vector.

2. Finite element modeling

To do a finite element modeling, we need to first discretize a 2D domain into many elements. In this report the turner triangle element is adopted. That element has three nodes 1, 2, 3 on its vertices, shown on Fig. 1.

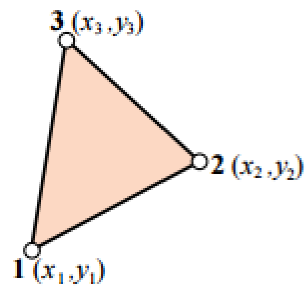

Figure 1: The schematic of the turner triangle element

The displacement and force vector of this element will be:

$$\vec{u} = [u_{1,x} \quad u_{1,y} \quad u_{2,x} \quad u_{2,y} \quad u_{3,x} \quad u_{3,y}]^T$$

$$\vec{f} = [f_{1,x} \quad f_{1,y} \quad f_{2,x} \quad f_{2,y} \quad f_{3,x} \quad f_{3,y}]^T$$

The displacement of other points inside the element can be determined by linear interpolation:

$$\begin{bmatrix} u_x(x,y) \\ u_y(x,y) \end{bmatrix} = \begin{bmatrix} N_1(x,y) & 0 & N_2(x,y) & 0 & N_2(x,y) & 0 \\ 0 & N_1(x,y) & 0 & N_2(x,y) & 0 & N_2(x,y) \end{bmatrix} \vec{u} = N\vec{u}$$

$$N_1(x,y) = \frac{2(x_2y_3 - x_3y_2) + x(y_2 - y_3) + y(x_3 - x_2)}{2A}$$

$$N_2(x,y) = \frac{2(x_3y_1 - x_1y_3) + x(y_3 - y_1) + y(x_1 - x_3)}{2A}$$

$$N_1(x,y) = \frac{2(x_1y_2 - x_2y_1) + x(y_1 - y_2) + y(x_2 - x_1)}{2A}$$

Here A is the area of the triangle.

Then we can differentiate the displacement get the strain vector and stress vector of this element:

$$\vec{\varepsilon} = B\vec{u}, B = \frac{1}{2A} \begin{bmatrix} y_2 - y_3 & 0 & y_3 - y_1 & 0 & y_1 - y_2 & 0 \\ 0 & x_3 - x_2 & 0 & x_1 - x_3 & 0 & x_2 - x_1 \\ x_3 - x_2 & y_2 - y_3 & x_1 - x_3 & y_3 - y_1 & x_2 - x_1 & y_1 - y_2 \end{bmatrix}$$

$$\vec{\sigma} = E\vec{\varepsilon} = EB\vec{u}, \qquad E = \frac{E}{1 - v^2} \begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & \frac{1-v}{2} \end{bmatrix}$$

The total potential energy will be:

$$W = \frac{1}{2}\int h\sigma^T \varepsilon \, dA - u^T f = \frac{1}{2}\int hu^T B^T EBu \, dA - u^T f = \frac{1}{2}u^T (AhB^T EB)u - u^T f$$

Here h is the thickness of the 2D domain in the z-direction.

To get static equilibrium we need $\delta W = 0$, which is the same as:

$$\vec{f} = AhB^T EB\vec{u} = K\vec{u}, \;\; K = AhB^T EB,$$

The matrix K is so called stiffness matrix of the element. It is a 6 by 6 matrix which relates the DOF of three nodes on x and y direction to the external force on the three nodes on x and y direction.

In finite element modeling, we will divide the 2D domain to many elements, calculate the stiffness matrix of each element and combine them together to get the stiffness matrix of the system. Once we have that system stiffness matrix, we can apply the boundary condition and solve $\vec{f} = K\vec{u}$.

3. Implementation on MATLAB

In this part I write FEM code on MATLAB to simulate the deflection of an aluminum cantilever beam with a load on the end of the beam. The beam is 100mm in length, 10mm*10mm cross-section. The Young's modulus is 70GPa = 70000 N/mm^2. The Poisson ratio is 0.33. The load is 10N on the –y direction at the end of the beam.

First part of the code is meshing the beam. Figure 2 shows the meshing on MATLAB. I divide the beam into 100*10 0.1mm*0.1mm squares and then divide each square into four right triangle elements. The total number of elements is 4000. The total number of nodes is 2111.

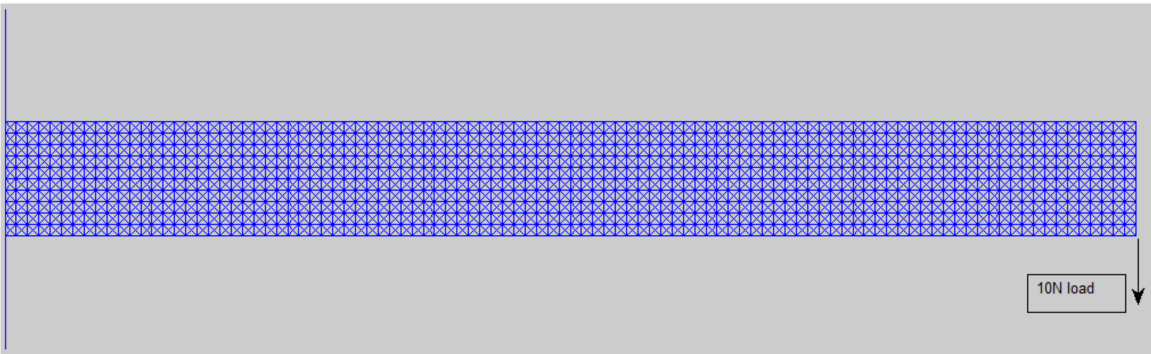

Figure 2: The meshing of the beam on MATLAB

Second part of the code is calculating the system stiffness matrix. I first build a 4222*4222 matrix for the system stiffness matrix. Then I traverse all the elements, calculate the stiffness matrix for each, and insert it into the corresponding row and column of the 4222*4222 matrix.

Third part of the code is apply the boundary condition and solve the f=Ku equation. The boundary condition consists of two parts. One is the 10N load on the end of the beam. The other is the zero displacement on the other end of the beam, corresponding to the fixed support. After applying the boundary condition and solving the equation, I get the following deflection plot:

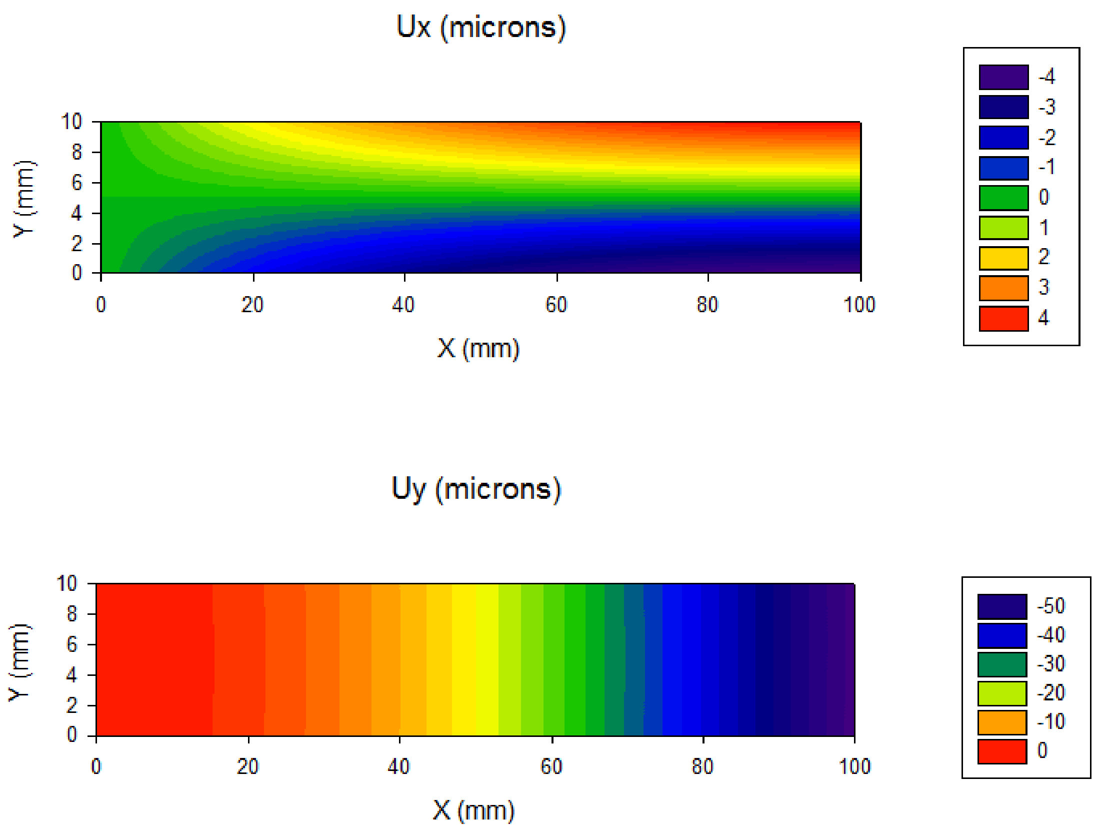

Figure 3: The displacement plot from the FEM simulation on MATLAB

The maximum y-deflection is 57.124 µm at the end of the beam.

Compared to the result calculated by equations provided on OPTI521 class:

$$\delta = \frac{FL^3}{3EI} = \frac{10N * 100mm^3}{3 * 70000Nmm^{-2} * \frac{1}{12} * 10mm * 10mm^3} = 57.143\mu m$$

The FEM simulation matches theory result perfectly.

4. Conclusion

This report introduces the basic concepts and principles of Finite Element Modeling (FEM). Once knowing these basic things, the FEM can be written into codes and used to analyze any complicate 2D stress-strain problems. The FEM can be furthermore extended into 3D using a different E matrix and a different 3D element, but the process is still the same.

Reference:

1. Website of class ASEN 5007 in UC-Boulder: http://www.colorado.edu/engineering/cas/courses.d/IFEM.d/
2. Wikipedia of Cauchy stress tensor: https://en.wikipedia.org/wiki/Cauchy_stress_tensor
3. Prof. Burge's class notes of OPTI 521 in College of Optical Science, University of Arizona