FEA Simulation Manual

Yangjun Liu

marcoliu@nuaa.edu.cn

Table of Contents

Abstract

MATLAB is a multi-paradigm numerical computing environment which allows matrix manipulations, plotting of data and symbolic computing. It's programming -friendly to achieve Finite Element Analysis. This is a simulation report of three types of finite elements using MATLAB language. Each chapter is divided into three sections: Basic Equations and Steps, Simulation Results and Additional Example. The examples from the textbook ***Variational principle and Finite Element Method*** will be applied at every step expounded in the Basic Equations and Steps section. The second section gives the simulation results and diagrams of these examples. Additional examples from the reference books is simulated to check the wide applicability of programs. The Example 1 and Example 2 are used to distinguish examples from the textbook and examples from the reference books.

1 The Plane Frame Element

The Plane Frame Element (a.k.a. The Beam Element with axial forces) is a 2D element with two nodes and six degrees of freedom. Compared with the normal plane beam element, axial forces must be considered besides shear forces and bending moments. The plane frame element has modulus of elasticity E, moment of inertia I, cross-sectional area A. The E, I, A were assigned to 2×10^7 N/cm^2, 25cm^4, 10cm^2 respectively in Example 1. There are two coordinate systems named the local coordinate system and the global coordinate system. Example 1 shown in Fig. 1.1 is from the textbook ***Variational principle and Finite Element Method***(page 79).

求图 5－8 所示刚架的节点位移和内力。各杆材料和几何尺寸均相同。$E = 2\times10^7\ \mathrm{N/cm^2}$，$a = 100\mathrm{cm}$，$A = 10\mathrm{cm}^2$，$J_z = 25\mathrm{cm}^4$，$P = 10000\mathrm{N}$。

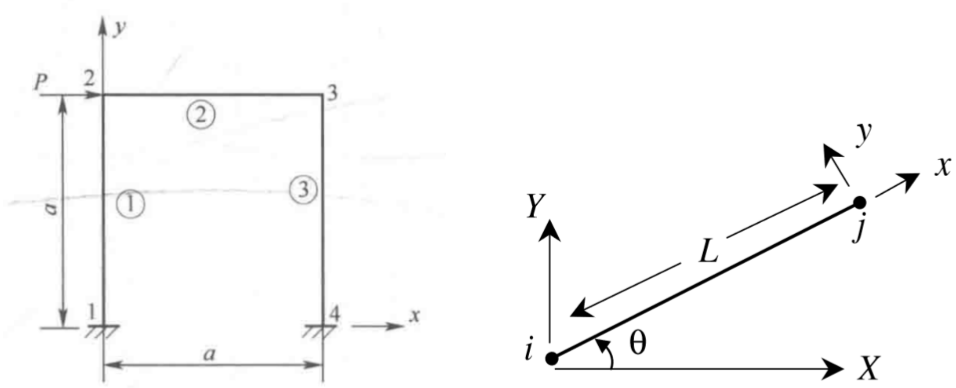


Fig 1.1 Example 1 of The Plane Frame Element

1.1 Basic Equations and Steps

To solve the plane frame element problem, the PlaneFrameMain function runs by the following steps.

Step 1 - Discretizing the rigid Frame (Inputting data)

Users must discretize the structure, number the elements and number the nodes manually, then input those data including the nodal global coordinates matrix, the element connectivity list, the nodal displacement vector and the equivalent nodal force vector into the program with the help of the human-computer interaction section. Table 1.1 shows the discretizing results of Example 1.

Table 1.1 Discretizing results of Example 1

单元号		节点坐标				l/cm	A/cm^2	J/cm^4	E/(N/cm^2)	方向余弦			
编号	i-j	x_i/cm	y_i/cm	x_j/cm	y_j/cm					$\lambda_{\bar{x}x}$	$\lambda_{\bar{x}y}$	$\lambda_{\bar{y}x}$	$\lambda_{\bar{y}y}$
①	1-2	0	0	0	100	100	10	25	2×10^7	0	1	-1	0
②	2-3	0	100	100	100	100	10	25	2×10^7	1	0	0	1
③	3-4	100	100	100	0	100	10	25	2×10^7	0	-1	1	0

Step 2 - Calculating the element Stiffness Matrix

The length of each beam are calculated from the x, y coordinates automatically using the BeamLength function. The PlaneFrameElementStiffness function calculates the coordinate transformation matrix R, the local stiffness matrix k' of each element and element stiffness matrix k step by step. The matrices k' and R are given by the following expression.

$$[k'] = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \tag{1.1}$$

$$[R] = \begin{bmatrix} C & S & 0 & 0 & 0 & 0 \\ -S & C & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & C & S & 0 \\ 0 & 0 & 0 & -S & C & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.2}$$

The element stiffness matrix k is calculated by using the following equation:

$[k] = [R]^T [k'][R]$ (1.3)

The coordinate transformation matrix and the stiffness matrix of element 1 in Example 1 are shown below:

```
lamda(:, :, 1) =

     0     1     0     0     0     0
    -1     0     0     0     0     0
     0     0     1     0     0     0
     0     0     0     0     1     0
     0     0     0    -1     0     0
     0     0     0     0     0     1

k(:, :, 1) =

        6000           0     -300000       -6000           0     -300000
           0     2000000           0           0    -2000000           0
     -300000           0    20000000      300000           0    10000000
       -6000           0      300000        6000           0      300000
           0    -2000000           0           0     2000000           0
     -300000           0    10000000      300000           0    20000000
```

Step 3 - Assembling element stiffness matrixes and get the frame stiffness matrix

The PlaneFrameAssemble function adds zeros to expand the element stiffness matrix from 6×6 to 3n×3n in order to obtain the global stiffness matrix by matrix addition, while n is the total number of nodes. The global stiffness matrix of Example 1 is shown below:

```
K =

        6000           0     -300000       -6000           0     -300000           0           0           0           0           0           0
           0     2000000           0           0    -2000000           0           0           0           0           0           0           0
     -300000           0    20000000      300000           0    10000000           0           0           0           0           0           0
       -6000           0      300000     2006000           0      300000    -2000000           0           0           0           0           0
           0    -2000000           0           0     2006000      300000           0       -6000      300000           0           0           0
     -300000           0    10000000      300000      300000    40000000           0     -300000    10000000           0           0           0
           0           0           0    -2000000           0           0     2006000           0      300000       -6000           0      300000
           0           0           0           0       -6000     -300000           0     2006000     -300000           0    -2000000           0
           0           0           0           0      300000    10000000      300000     -300000    40000000     -300000           0    10000000
           0           0           0           0           0           0       -6000           0     -300000        6000           0     -300000
           0           0           0           0           0           0           0    -2000000           0           0     2000000           0
           0           0           0           0           0           0      300000           0    10000000     -300000           0    20000000
```

Step 4 - Constraint handling - applying the boundary conditions

The constraint handling section of PlaneFrameMain function applies the displacement boundary conditions and equivalent nodal loads to calculate unknown nodal dis placements and unknown nodal loads. Users should note that 0.0011 represents unknown displacements, while 1.111 represents unknown nodal loads. The nodal displacement vector U and the equivalent nodal loads vector P of Example 1 are shown below. The horizontal displacement u(cm), vertical displacement v(cm) and rotation φ(rad) at node i can be queried by U(3i-2), U(3i-1), U(3i) respectively, and the horizontal force Px(N), vertical force Py(N) and moment M(N·cm) at node i can be queried by P(3i-2), P(3i-1), P(3i) respectively. The u, v, and φ should be assigned to zeros at the fixed ends, while the u, v, and φ should be assigned to 0.0011 at the free nodes.

U=[0;0;0; 0.0011;0.0011;0.0011;0.0011;0.0011;0.0011; 0;0;0]

P=[1.111;1.111;1.111; 10000;0;0;0;0;0; 1.111;1.111;1.111]

Step 5 - Solving the system of linear equations (in global coordinate system)

Once the global stiffness matrix K is obtained we have the following structure equation:

$$[K]U = P \tag{1.3}$$

The Constraint Handling section calculates unknown nodal displacements and unknown reactions by matrix partition and Gaussian elimination. The calculated vectors U and P are shown below:

```
U =                 P =

        0           1.0e+05 *
        0
        0           -0.0500
   1.1936           -0.0428
   0.0021            2.8615
  -0.0072            0.1000
   1.1911                 0
  -0.0021                 0
  -0.0072                 0
        0                 0
        0                 0
        0           -0.0500
                     0.0428
                     2.8565
```

Step 6 - Calculating nodal internal forces of each beam (in local coordinate system)

PlaneFrameElementNodalInternalForces function calculate nodal internal forces of each element on the basis of equation (1.4). The nodal internal forces of Example 1 are shown as F matrix.

$$f=[k'][R]u \qquad (1.4)$$

```
F =

   1.0e+05 *

  -0.0428    0.0500    0.0428
   0.0500   -0.0428    0.0500
   2.8615   -2.1423    2.1398
   0.0428   -0.0500   -0.0428
  -0.0500    0.0428   -0.0500
   2.1423   -2.1398    2.8565
```

1.2 Simulation Results

The nodal displacements were obtained at step 5.

```
U =

          0
          0
          0
     1.1936
     0.0021
    -0.0072
     1.1911
    -0.0021
    -0.0072
          0
          0
          0
```

The internal force diagrams of Example 1 were plotted by PlaneFrameElementInternalForceDiagram2 function.

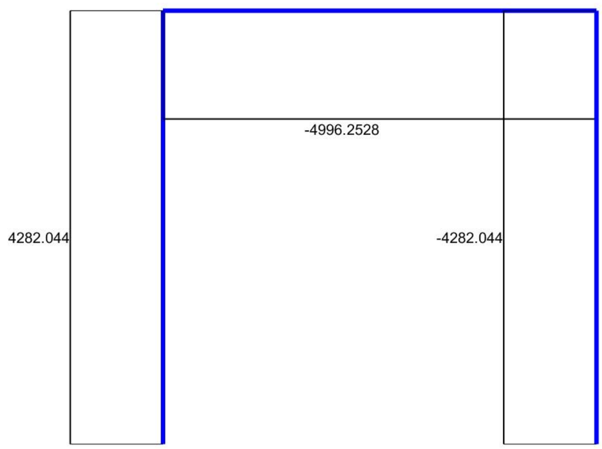


Fig 1.2 Axial Force Diagram of Example 1

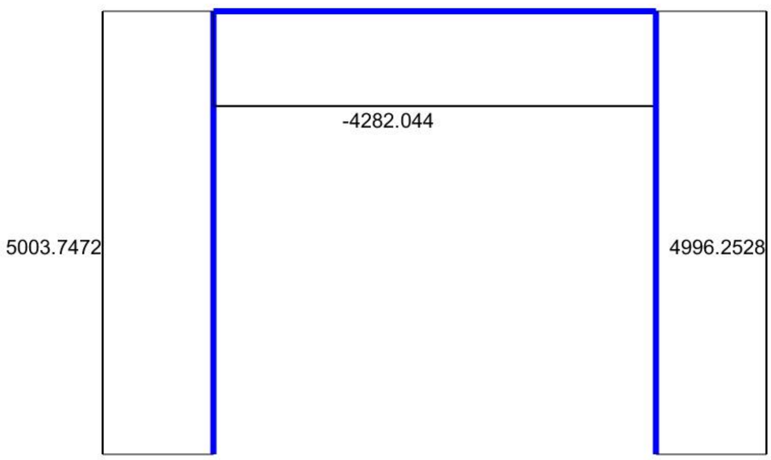


Fig 1.3 Shear Force Diagram of Example 1

Fig 1.4　Bending Moment Diagram of Example 1

Compared to the results given in textbook Variational principle and Finite Element Method, the computational error is within a reasonable range.

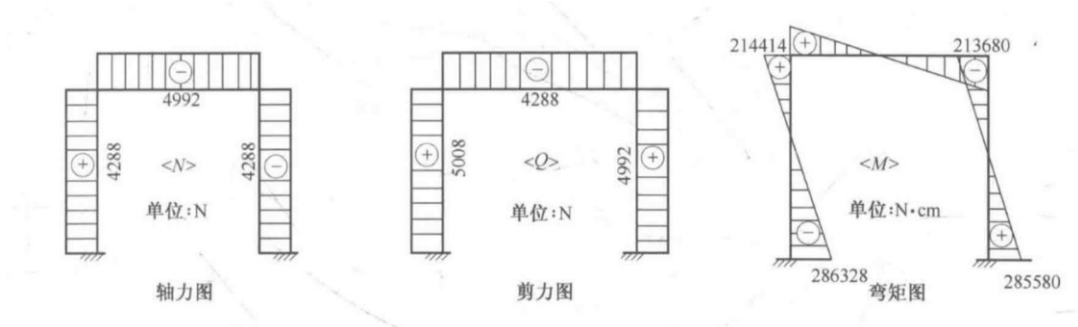


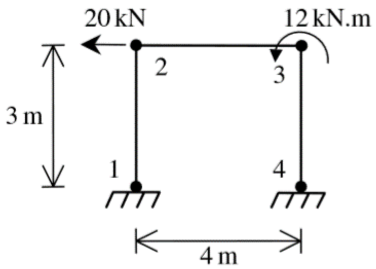
Fig 1.5 Internal force diagrams given in textbook

1.3 Additional Example

Example 2 is from the page 143 of ***MATLAB Guide to Finite Elements***

Consider the plane frame shown in Fig. 8.3. Given $E = 210\,\text{GPa}$, $A = 2 \times 10^{-2}\,\text{m}^2$, and $I = 5 \times 10^{-5}\,\text{m}^4$, determine:

1. the global stiffness matrix for the structure.
2. the displacements and rotations at nodes 2 and 3.
3. the reactions at nodes 1 and 4.
4. the axial force, shear force, and bending moment in each element.
5. the axial force diagram for each element.
6. the shear force diagram for each element.
7. the bending moment diagram for each element.



Internal forces calculated by PlaneFrameMain function:

F =

1.0e+06 *

0.008586518257709	-0.007810292633703	-0.008586518257709
-0.012189707366297	0.008586518257709	-0.007810292633704
-2.102534895380519	1.554377314508556	-0.680229988575155
-0.008586518257709	0.007810292633703	0.008586518257709
0.012189707366297	-0.008586518257709	0.007810292633704
-1.554377314508555	1.880229988575155	-1.662857801535917

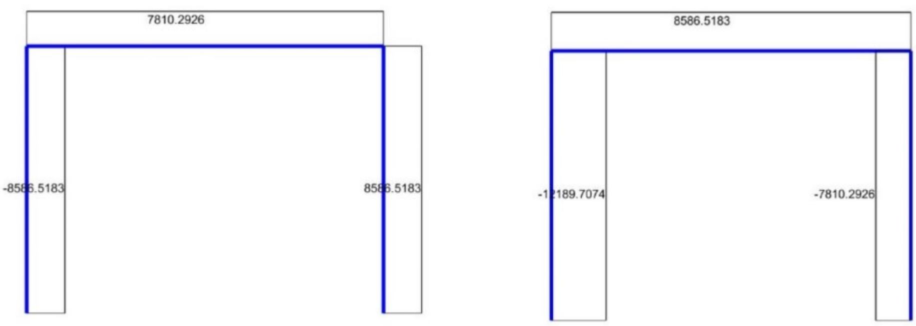


Fig 1.6 Axial Force Diagram and Shear Force Diagram of Example 2

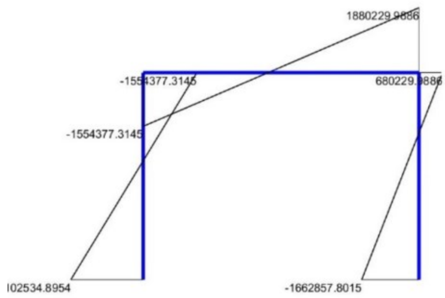


Fig 1.7 Bending Moment Diagram of Example 2

2 The Constant strain triangular element

The Constant Strain Triangular Element (abbreviated as CST Element in this report), also known as The Linear Triangular Element, in Fig. 2.1, is a 2D element with three nodes and six degrees of freedom. It has modulus of elasticity E (kN/m^2), Poisson's ratio ν, thickness t (m). The E, ν and t were assigned to 2×10^6 kN/m^2, 0.167, 0.4m respectively in Example 1. There is just one coordinate system named the global coordinate system. Example 1 shown in Fig. 2.2 is from the the textbook ***Variational principle and Finite Element Method***(page 109).

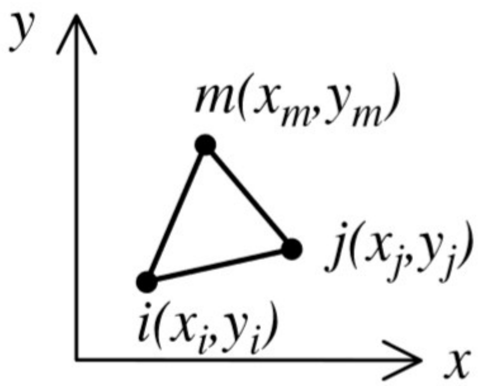


Fig 2.1 The Constant strain triangular element

矩形截面简支梁如图 6－20 所示，跨度 18m，高 3m，厚度 0. 4m，弹性模量 $E=2\times10^6$t/m^2，泊松比 μ=0. 167。在梁的上顶面上承受均匀载荷 $q=10$t/m^2。

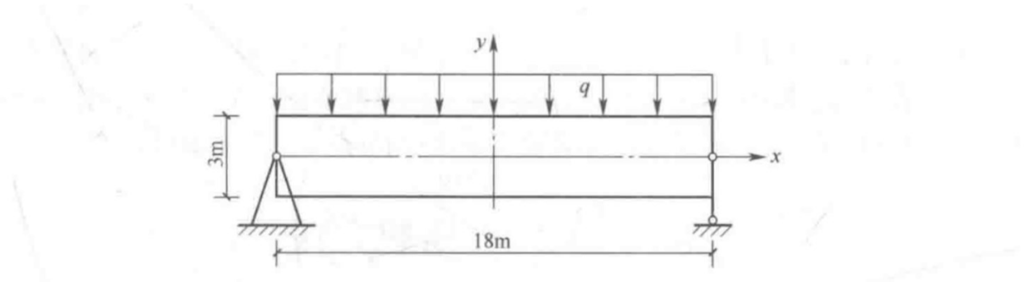


Fig 2.2 Example 1

2.1 Basic Equations and Steps

To solve the plane frame element problem, the ConstantStrianTriangularElementMain function runs by the following steps.

Step 1: Discretizing the Structure (Inputting data)

Users must discretize the structure, number the elements and number the nodes manually, then input the data into the program with the help of the human-computer interaction section.

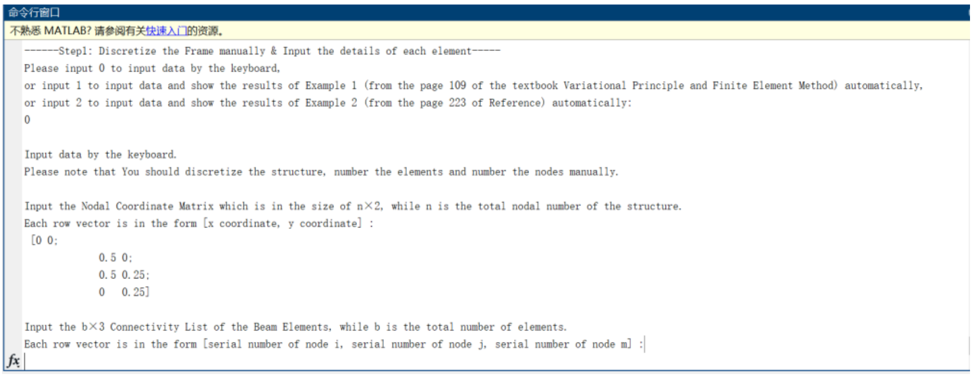


Fig 2.3 human-computer interaction (inputting by the keyboard)

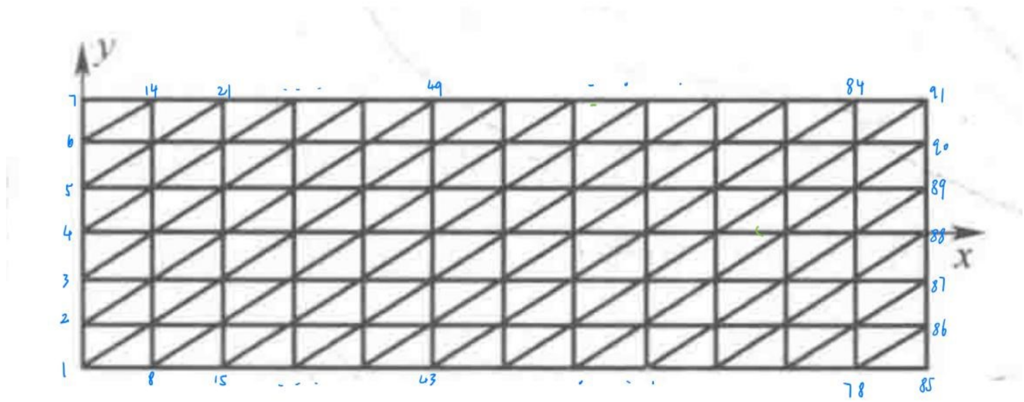


Fig 2.4 Discretizing result of Example 1

Step 2: Calculating the element Stiffness Matrix

The triangular element area are calculated from the x, y coordinates automatically using the CSTElementArea function. The CSTElementStiffness function calculates the Element Geometric Matrix (a.k.a. The Element Strain Matrix) B and the element stiffness matrix k based on the equations below.

$$\begin{aligned}\beta_i &= y_j - y_m\\ \beta_j &= y_m - y_i\\ \beta_m &= y_i - y_j\\ \gamma_i &= x_m - x_j\\ \gamma_j &= x_i - x_m\\ \gamma_m &= x_j - x_i\end{aligned} \tag{2.1}$$

$$[B] = \frac{1}{2A}\begin{bmatrix}\beta_i & 0 & \beta_j & 0 & \beta_m & 0\\ 0 & \gamma_i & 0 & \gamma_j & 0 & \gamma_m\\ \gamma_i & \beta_i & \gamma_j & \beta_j & \gamma_m & \beta_m\end{bmatrix} \tag{2.2}$$

$$[k] = tA[B]^T[D][B] \tag{2.3}$$

The Elasticity Matrix D is chosen according to the question type ('stress' or 'strain'). For cases of plane stress the matrix D is given by

$$[D] = \frac{E}{1-\nu^2}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & \dfrac{1-\nu}{2}\end{bmatrix} \tag{2.4}$$

For cases of plane strain the matrix D is given by

$$[D] = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix}1-\nu & \nu & 0\\ \nu & 1-\nu & 0\\ 0 & 0 & \dfrac{1-2\nu}{2}\end{bmatrix} \tag{2.5}$$

The Element Geometric Matrix and the element stiffness matrix of element 1 in Example 1are shown below:

```
>> B(:,:,1)

ans =

   -1.3333         0    1.3333         0         0         0
         0         0         0   -2.0000         0    2.0000
         0   -1.3333   -2.0000    1.3333    2.0000         0

>> k(:,:,1)

ans =

   1.0e+05 *

    2.7432         0   -2.7432    0.6872         0   -0.6872
         0    1.1425    1.7138   -1.1425   -1.7138         0
   -2.7432    1.7138    5.3139   -2.4010   -2.5707    0.6872
    0.6872   -1.1425   -2.4010    7.3147    1.7138   -6.1721
         0   -1.7138   -2.5707    1.7138    2.5707         0
   -0.6872         0    0.6872   -6.1721         0    6.1721
```

Step 3: Assembling element stiffness matrices

The CSTElementStiffnessMatrixAssemble function adds zeros to expand the element stiffness matrix from 6×6 to 2n×2n in order to obtain the global stiffness matrix K by matrix addition, while n is the total number of nodes. The global stiffness matrix of Example 1 is too large to be shown here.

Step 4: Applying the boundary conditions and calculating the unknown displacements and reactions

The Constraint Handling section of ConstantStrianTriangularElementMain function applies the displacement boundary conditions and equivalent nodal loads to calculate unknown nodal displacements and unknown nodal loads. Users should note that 0.0011 represents unknown displacements, while 1.111 represents unknown nodal loads. The horizontal displacement u(m) and vertical displacement v(m) at node i can be queried by U(2i-1), U(2i) respectively, while the horizontal force Px(kN) and vertical force Py(kN) at node i can be queried by P(2i-1), P(2i) respectively. The v should be assigned to zero at simply support, while the u, v should be assigned to 0.0011 at the free nodes. Once the global stiffness matrix K is obtained we have the following equation:

$$[K]U = P \qquad (2.6)$$

The Constraint Handling section calculates unknown nodal displacements and unknown reactions by matrix partition and Gaussian elimination. The vertical displacements and the horizontal forces at line x=0 are shown below:

```
>> U(2*7:-2:2)

ans =

   -0.0028
   -0.0028
   -0.0029
   -0.0029
   -0.0029
   -0.0029
   -0.0028
```

```
>> P(2*7-1:-2:2-1)

ans =

   26.2126
   32.1934
   16.3034
    0.6067
  -15.0821
  -31.1120
  -29.1220
```

Step 5: Calculating the element stress vectors

The CSTElementStresses function calculates element stresses using the Elasticity Matrix, the Geometric Matrix and the element displacement vector. It returns the 3x1 element stress vector in the form of $\{\sigma\}=[\sigma_x \ \ \sigma_y \ \ \tau_{xy}]^T$

$$\{\sigma\} = [D][B]\{u\} \qquad (2.7)$$

2.2 Simulation Results

The σx at straight line x=0:

```
-262.4025  -183.9321  -105.4617   -26.9914    51.4790   129.9493   208.4197
```

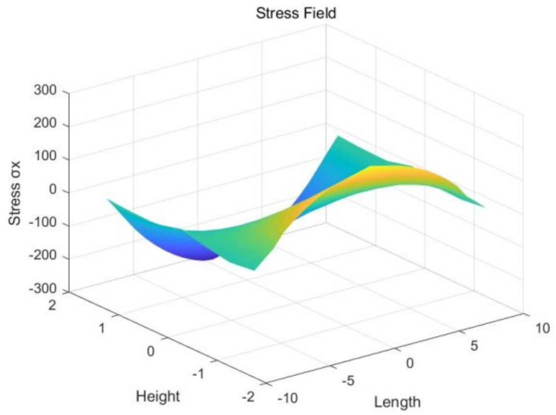


Fig 2.5 Global stress field of σx

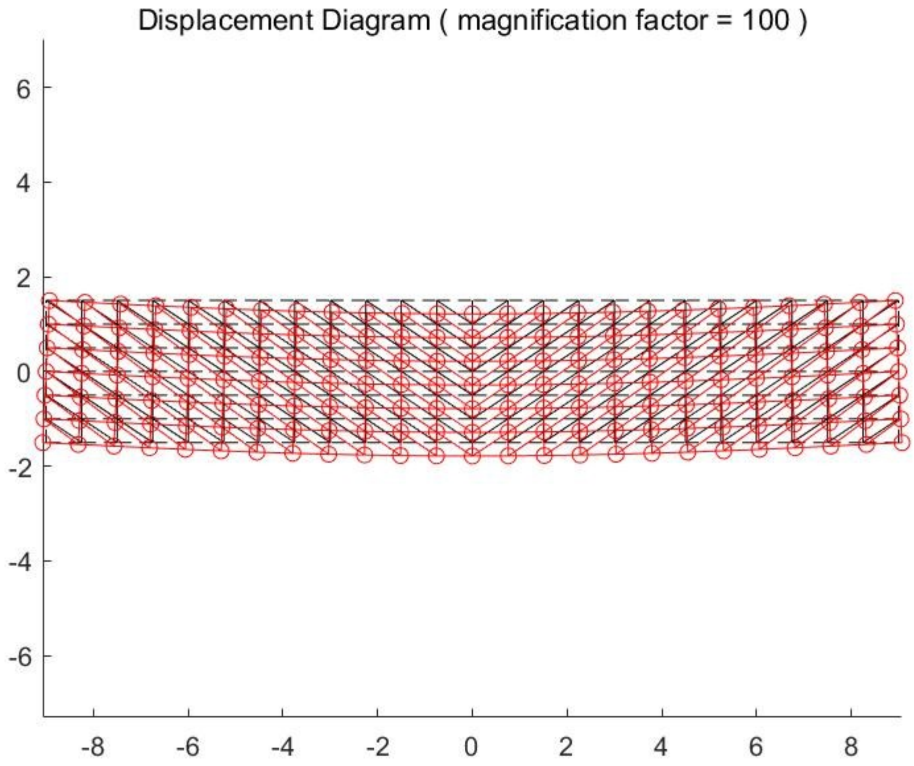


Fig 2.6 Global Displacement Diagram

Compared to the results given in textbook ***Variational principle and Finite Element Method***, the computational error is within a reasonable range.

Table 2.1 results given in the textbook

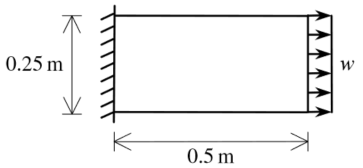
截面高度 / 单元类型	1.5	1.0	0.5	0	−0.5	−1.0	−1.5
常应变三角形单元	−262	−184	−106	−28	50	131	212

2.3 Additional Example

Example 2 is from the page 222 of ***MATLAB Guide to Finite Elements.***

Consider the thin plate subjected to a uniformly distributed load as shown in Fig. 11.2. The plate is discretized using two linear triangular elements as shown in Fig. 11.3. Given $E = 210\,\mathrm{GPa}$, $\nu = 0.3$, $t = 0.025\,\mathrm{m}$, and $w = 3000\,\mathrm{kN/m^2}$, determine:

1. the global stiffness matrix for the structure.
2. the horizontal and vertical displacements at nodes 2 and 3.
3. the reactions at nodes 1 and 4.
4. the stresses in each element.
5. the principal stresses and principal angle for each element.



The displacement diagram of Example 2 was plotted by the ConstantStrianTriangularElementMain function:

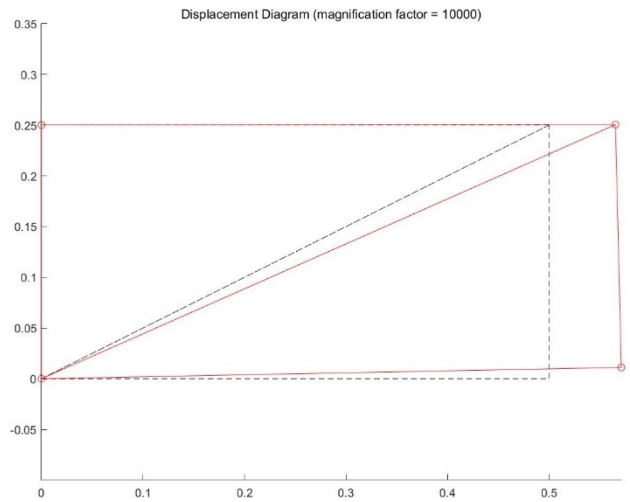


Fig 2.7 Displacement Diagram of Example 2

3 The Eight Nodes Quadrilateral Element

The Eight Nodes Quadrilateral Element (a.k.a. The Quadratic Quadrilateral Element) shown in Fig. 3.1 is a 2D isoperimetric element with eight nodes and 16 degrees of freedom. It has modulus of elasticity E ($\mathrm{kN/m^2}$), Poisson's ratio ν, thickness t (m). The

E, ν and t were assigned to 2×10^6 kN/m^2, 0.167, 0.4m respectively in Example 1. There are two coordinate systems named the global coordinate system and the natural coordinate system respectively. Example 1 is from the textbook ***Variational principle and Finite Element Method***(page 109)**.**

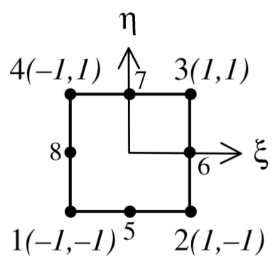


Fig 3.1 The Eight Points Element with Natural Coordinates

3.1 Basic Equations and Steps

To solve the plane frame element problem, the EightPointsQuadrilateralElementMain function runs by the following steps.

Step 1: Discretizing the Structure (Inputting data)

Users must discretize the structure, number the elements and number the nodes manually, then input the data into the program with the help of the human-computer interaction section.

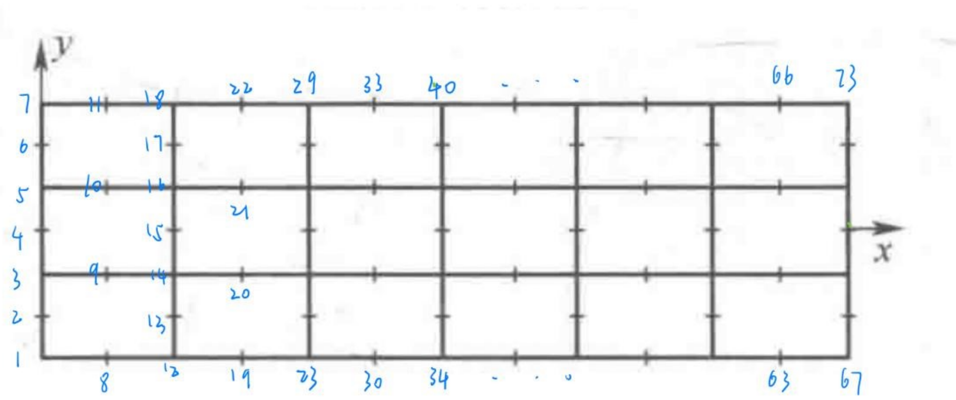


Fig 2.4 Discretizing result of Example 1

Step 2: Calculating the element Stiffness Matrix

The triangular element area is calculated from the x, y coordinates automatically. The QuadraticQuadElementStiffness function calculates the Element Geometric Matrix B and the element stiffness matrix k based on the equations below.

The shape functions is given by

$$\begin{aligned}
N_1 &= \frac{1}{4}(1-\xi)(1-\eta)(-\xi-\eta-1)\\
N_2 &= \frac{1}{4}(1+\xi)(1-\eta)(\xi-\eta-1)\\
N_3 &= \frac{1}{4}(1+\xi)(1+\eta)(\xi+\eta-1)\\
N_4 &= \frac{1}{4}(1-\xi)(1+\eta)(-\xi+\eta-1)\\
N_5 &= \frac{1}{2}(1-\eta)(1+\xi)(1-\xi)\\
N_6 &= \frac{1}{2}(1+\xi)(1+\eta)(1-\eta)\\
N_7 &= \frac{1}{2}(1+\eta)(1+\xi)(1-\xi)\\
N_8 &= \frac{1}{2}(1-\xi)(1+\eta)(1-\eta)
\end{aligned} \tag{3.1}$$

$$\begin{aligned}
x &= N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4 + N_5x_5 + N_6x_6 + N_7x_7 + N_8x_8\\
y &= N_1y_1 + N_2y_2 + N_3y_3 + N_4y_4 + N_5y_5 + N_6y_6 + N_7y_7 + N_8y_8
\end{aligned} \tag{3.2}$$

$$[N] = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 & N_5 & 0 & N_6 & 0 & N_7 & 0 & N_8 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 & N_5 & 0 & N_6 & 0 & N_7 & 0 & N_8 \end{bmatrix} \tag{3.3}$$

The Jacobi matrix for this element is given by

$$[J] = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} \tag{3.4}$$

B is given by

$$[D^{/}] = \frac{1}{|J|}\begin{bmatrix} \frac{\partial y}{\partial \eta}\frac{\partial (\ )}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial (\ )}{\partial \eta} & 0 \\ 0 & \frac{\partial x}{\partial \xi}\frac{\partial (\ )}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial (\ )}{\partial \xi} \\ \frac{\partial x}{\partial \xi}\frac{\partial (\ )}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial (\ )}{\partial \xi} & \frac{\partial y}{\partial \eta}\frac{\partial (\ )}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial (\ )}{\partial \eta} \end{bmatrix} \tag{3.5}$$

$$[B] = [D^{/}][N] \tag{3.6}$$

The element stiffness matrix for the Eight Points quadrilateral element is written in terms of a double integral as follows:

$$[k] = t\int_{-1}^{1}\int_{-1}^{1} [B]^T[D][B]\,|J|\,d\xi d\eta \tag{3.7}$$

The Elasticity Matrix D is chosen according to the question type ('stress' or 'strain').

Step 3: Assembling element stiffness matrices

The QuadraticQuadAssemble function adds zeros to expand the element stiffness matrix from 16×16 to 2n×2n in order to obtain the global stiffness matrix K by matrix addition, while n is the total number of nodes. The global stiffness matrix of Example 1 is too large to be shown here.

Step 4: Applying the boundary conditions and calculating the unknown displacements and reactions

The Constraint Handling section of EightPointsQuadrilateralElementMain function applies the displacement boundary conditions and equivalent nodal loads to calculate unknown nodal displacements and unknown nodal loads. Users should note that 0.0011 represents unknown displacements, while 1.111 represents unknown nodal loads. The horizontal displacement u(m) and vertical displacement v(m) at node i can be queried by U(2i-1), U(2i) respectively, while the horizontal force Px(kN) and vertical force Py(kN) at node i can be queried by P(2i-1), P(2i) respectively. The v should be assigned to zero at simply support, while the u, v should be assigned to 0.0011 at the free nodes.

Once the global stiffness matrix K is obtained we have the following equation:

$$[K]U = P \tag{3.8}$$

The Constraint Handling section calculates unknown nodal displacements and unknown reactions by matrix partition and Gaussian elimination. The vertical displacements and the horizontal forces at line x=0 are shown below:

```
>> U(2*7:-2:2)          >> P(2*7-1:-2:1)

ans =                   ans =

   -0.0032                 18.1016
   -0.0032                 47.9187
   -0.0032                 11.8577
   -0.0032                  0.0000
   -0.0032                -11.8577
   -0.0032                -47.9187
   -0.0032                -18.1016
```

Step 5: Calculating the element stress vectors

The QuadraticQuadElementStresses function calculates element stresses using the Elasticity Matrix, the Geometric Matrix and the element displacement vector. It returns the 3x1 element stress vector in the form of $\{\sigma\}=[\sigma_x \quad \sigma_y \quad \tau_{xy}]^T$

$$\{\sigma\} = [D][B]\{u\} \tag{3.9}$$

3.2 Simulation Results

The stresses at line x=0 in the form of $[\sigma_x \quad \sigma_y \quad \tau_{xy}]^T$

-273.2214	-180.2195	-89.6062	-0.0000	89.6062	180.2195	273.2214
-10.5785	-9.0868	-7.4496	-5.0000	-2.5504	-0.9132	0.5785
0.1795	0.2569	0.2714	0.2928	0.2714	0.2569	0.1795

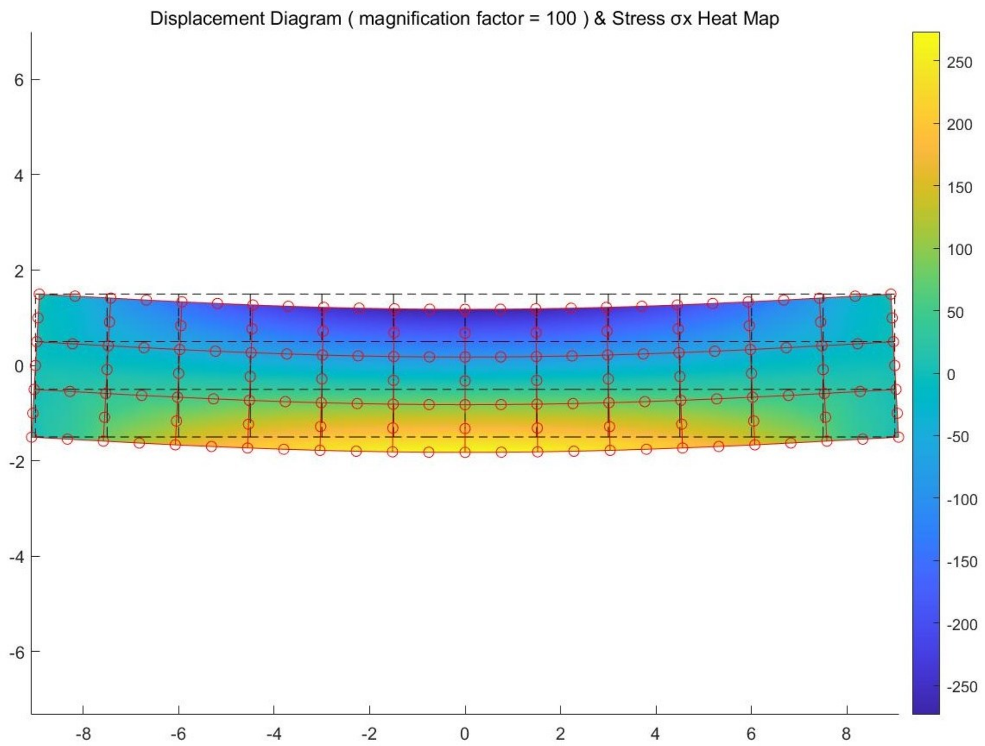


Fig 3.3 Displacement and Stress Diagram of Example 1

Compared to the results given in textbook ***Variational principle and Finite Element Method***, the computational error is within a reasonable range.

Table 3.1 results given in the textbook

截面高度 / 单元类型	1.5	1.0	0.5	0	−0.5	−1.0	−1.5
常应变三角形单元	−262	−184	−106	−28	50	131	212
四节点矩形单元	−265	−174	−85	0.0	85	174	265
六节点三角形单元	−272.9	−180.6	−89.2	−0.64	89.2	179.7	271.4
八节点等参单元	−272.6	−180.7	−89.7	0.0	89.7	180.7	272.6
弹性力学解析解	−272	−179.5	−89.2	0.0	89.2	179.5	272
材料力学解	−270	−180	−90	0.0	90	180	270

表 6−3　$x=0$ 截面上 σ_y 的分布(单位:t/m)

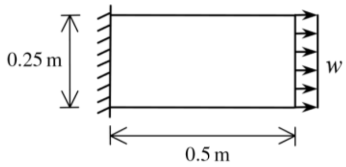
截面高度 / 单元类型	1.5	1.0	0.5	0	−0.5	−1.0	−1.5
四节点矩形单元	−10.1	−8.9	−7.2	−5.0	−2.8	−1.0	−0.2
六节点三角形单元	−10.0	−9.11	−7.75	−4.96	−2.5	−0.77	−0.58
八节点等参单元	−10.38	−9.07	−7.34	−5.0	−2.65	−0.93	0.38
弹性力学解析解	−10.1	−9.3	−7.4	−5.0	−2.6	−0.7	0.0

3.3 Additional Example

Example 2 is from the page 325 of ***MATLAB Guide to Finite Elements.***

Consider the thin plate subjected to a uniformly distributed load as shown in Fig. 14.3. This problem was solved in Example 13.1 using bilinear quadrilateral elements. Solve this problem again using one quadratic quadrilateral element as shown in Fig. 14.4. Given $E = 210\,\mathrm{GPa}$, $\nu = 0.3$, $t = 0.025\,\mathrm{m}$, and $w = 3000\,\mathrm{kN/m^2}$, determine:

1. the global stiffness matrix for the structure.
2. the horizontal and vertical displacements at nodes 3, 5, and 8.
3. the reactions at nodes 1, 4, and 6.
4. the stresses in the element.
5. the principal stresses and principal angle for the element.



The displacement diagram of Example 2 was plotted by the

EightPointsQuadrilateralElementMain function

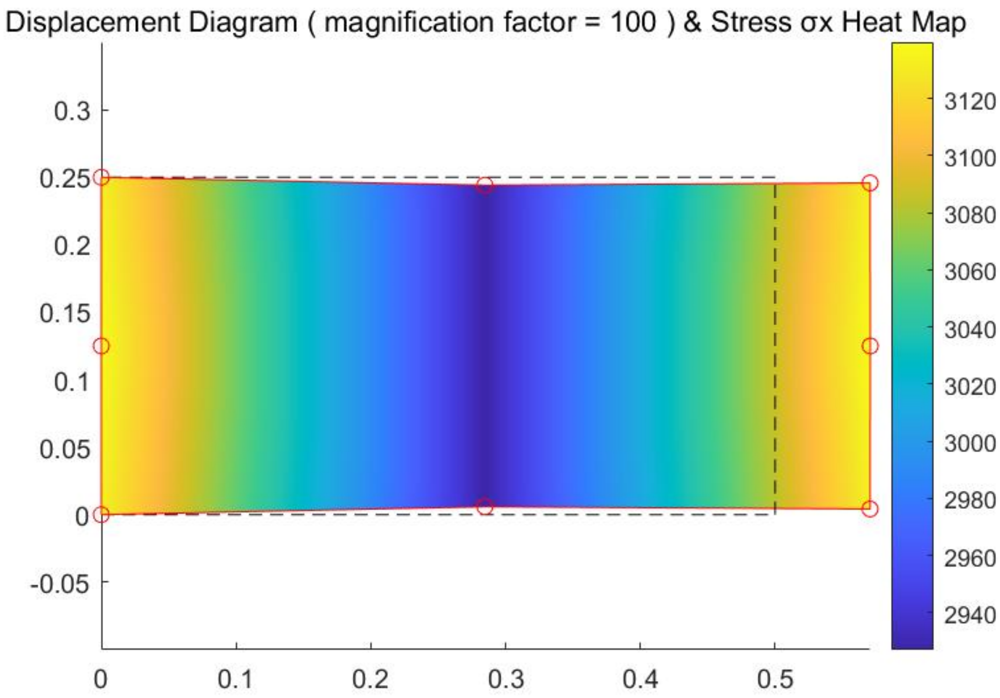


Fig 3.4 Displacement and Stress Diagram of Example 2