

**【知识点 7.6】 门型框架结构的 MATLAB 编程****【MATLAB 程序】 2D 梁单元的有限元分析程序(Beam2D2Node)**

编写平面梁单元的单元刚度矩阵、单元组装、单元应力的计算程序。

解答：

编写的平面 2 节点梁单元的三个 MATLAB 函数如下。

Beam2D2Node_Stiffness(E,I,A,L)

以上函数计算单元的刚度矩阵，输入弹性模量 E，横截面积 A，惯性矩 I，长度 L，输出单元刚度矩阵 $k(6 \times 6)$ 。

Beam2D2Node_Assemble(KK,k,i,j)

以上函数进行单元刚度矩阵的组装，输入单元刚度矩阵 k，单元的节点编号 i、j，输出整体刚度矩阵 KK。

Beam2D2Node_Forces(k,u)

以上函数计算单元的应力，输入单元刚度矩阵 k，节点位移 u，输出单元节点力 forces。

可以编写出具体实现以上每个函数的 MATLAB 程序如下。

```

%%%%%%%%%%%%%% Beam2D2Node %% begin %%%%%%%%%%%%%%%
function k =Beam2D2Node_Stiffness(E,I,A,L)
% 以上函数计算单元的刚度矩阵
% 输入弹性模量 E，横截面积 A，惯性矩 I，长度 L
% 输出单元刚度矩阵 k(6×6)
%-----
k=[E*A/L,0,0,-E*A/L,0,0;0,12*E*I/(L^3),6*E*I/(L^2),0,-12*E*I/(L^3),6*E*I/(L^2);0,6*E*I/(L^2),4*E*I/L,0,-
  6*E*I/(L^2),2*E*I/L;-E*A/L,0,0,E*A/L,0,0;0,-12*E*I/(L^3),-6*E*I/(L^2),0,12*E*I/(L^3),-
  6*E*I/(L^2);0,6*E*I/(L^2),2*E*I/L,0,-6*E*I/(L^2),4*E*I/L]
%%%%%%%%%%%%%%
function y =Beam2D2Node_Assemble(KK,k,i,j)
% 以上函数进行单元刚度矩阵的组装
% 输入单元刚度矩阵 k，单元的节点编号 i、j
% 输出整体刚度矩阵 KK
%-----
DOF(1)=3*i-2;
DOF(2)=3*i-1;
DOF(3)=3*i;
DOF(4)=3*j-2;
DOF(5)=3*j-1;
DOF(6)=3*j;
for n1=1:6
    for n2=1:6
        KK(DOF(n1),DOF(n2))=KK(DOF(n1),DOF(n2))+k(n1,n2);
    end
end
y = KK;
%%%%%%%%%%%%%%
function forces = Beam2D2Node_Forces(k,u)
% 以上函数计算单元的应力，输入单元刚度矩阵 k，节点位移 u，
% 输出单元节点力 forces
%-----

```



```
forces = k * u;
%%%%%%%%%%%% Beam2D2Node %%% end %%%%%%%%%%
```

【MATLAB 算例】 门型框架结构的 MATLAB 编程

如图所示的框架结构，其顶端受均布力作用，用有限元方法分析该结构的位移。结构中各个截面的参数都为： $E = 3.0 \times 10^{11} \text{ Pa}$ ， $I = 6.5 \times 10^{-7} \text{ m}^4$ ， $A = 6.8 \times 10^{-4} \text{ m}^2$ 。

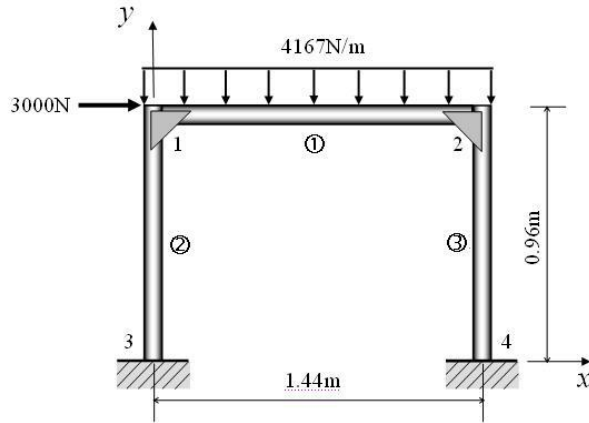
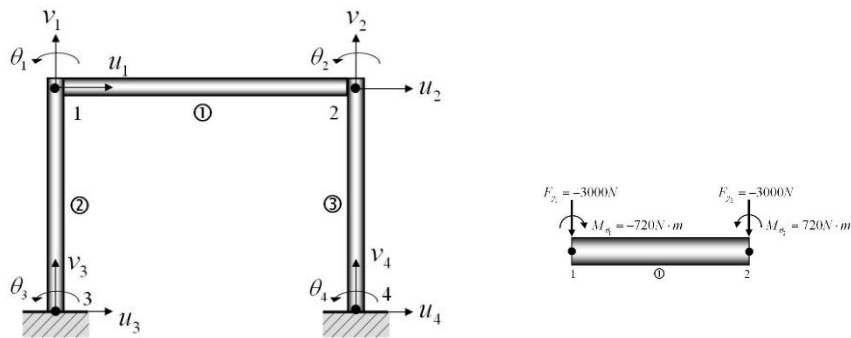


图 1 框架结构受一均布力作用



(a) 节点位移及单元编号 (b) 等效在节点上的外力

图 2 单元划分、节点位移及节点上的外载

解答：对该问题进行有限元分析的过程如下。

(1) 结构的离散化与编号

将该结构离散为三个单元，节点位移及单元编号如图 2 所示，有关节点和单元的信息见知识点 7.6。

(2) 各个单元的描述

首先在 MATLAB 环境下，输入弹性模量 E、横截面积 A、惯性矩 I、长度 L，然后针对单元 1 和单元 2 单元 3，分别二次调用函数 Beam2D2Node_Stiffness，就可以得到单元的刚度矩阵 k1(6×6)和 k2(6×6)，且单元 2 和单元 3 的刚度矩阵相同。

```
>> E=3E11;
>> I=6.5E-7;
>> A=6.8E-4;
```



```
>> L1=1.44;
>> L2=0.96;
>> k1=Beam2D2Node_Stiffness(E,I,A,L1);
>> k2=Beam2D2Node_Stiffness(E,I,A,L2);
```

(3) 建立整体刚度方程

将单元 2 和单元 3 的刚度矩阵转换成整体坐标下的形式。由于该结构共有 4 个节点，则总共的自由度数为 12，因此，结构总的刚度矩阵为 $KK(12 \times 12)$ ，对 KK 清零，然后两次调用函数 `Beam2D2Node_Assemble` 进行刚度矩阵的组装。

```
>> T=[0,1,0,0,0,0;-1,0,0,0,0,0;0,0,1,0,0,0;0,0,0,1,0,0;0,0,0,-1,0,0;0,0,0,0,0,1];
>> k3=T'*k2*T;
>> KK=zeros(12,12);
>> KK=Beam2D2Node_Assemble(KK,k1,1,2);
>> KK=Beam2D2Node_Assemble(KK,k3,3,1);
>> KK=Beam2D2Node_Assemble(KK,k3,4,2)
KK = 1.0e+008 *
    1.4431    0    0.0127   -1.4167    0    0   -0.0264    0    0.0127    0    0    0
         0    2.1328    0.0056    0   -0.0078    0.0056    0   -2.1250    0    0    0    0
    0.0127    0.0056    0.0135    0   -0.0056    0.0027   -0.0127    0    0.0041    0    0    0
   -1.4167    0    0    1.4431    0    0.0127    0    0    0   -0.0264    0    0.0127
         0   -0.0078   -0.0056    0    2.1328   -0.0056    0    0    0    0   -2.1250    0
         0    0.0056    0.0027    0.0127   -0.0056    0.0135    0    0    0   -0.0127    0    0.0041
   -0.0264    0   -0.0127    0    0    0    0.0264    0   -0.0127    0    0    0
         0   -2.1250    0    0    0    0    0    2.1250    0    0    0    0
    0.0127    0    0.0041    0    0    0   -0.0127    0    0.0081    0    0    0
         0    0    0   -0.0264    0   -0.0127    0    0    0    0.0264    0   -0.0127
         0    0    0    0   -2.1250    0    0    0    0    0    2.1250    0
         0    0    0    0.0127    0    0.0041    0    0    0   -0.0127    0    0.0081
```

(4) 边界条件的处理及刚度方程求解

该问题的位移边界条件为 $u_3 = v_3 = \theta_3 = u_4 = v_4 = \theta_4 = 0$ 。因此，将针对节点 1 和节点 2 的位移进行求解，节点 1 和节点 2 的位移将对应 KK 矩阵中的前 6 行和前 6 列，则需从 $KK(12 \times 12)$ 中提出，置给 k ，然后生成对应的载荷列阵 p ，再采用高斯消去法进行求解，注意：MATLAB 中的反斜线符号“\”就是采用高斯消去法。

```
>> k=KK(1:6,1:6);
>> p=[3000;-3000;-720;0;-3000;720];
>> u=k\p
u = 0.0009    -0.0000    -0.0014    0.0009    -0.0000    -0.0000    [将列排成了行]
```

(5) 支反力的计算

在得到整个结构的节点位移后，由原整体刚度方程就可以计算出对应的支反力；先将上面得到的位移结果与位移边界条件的节点位移进行组合(注意位置关系)，可以得到整体的位移列阵 $U(12 \times 1)$ ，再代回原整体刚度方程，计算出所有的节点力 $P(12 \times 1)$ ，按式(3-180)的对应关系就可以找到对应的支反力。

```
>> U=[u;0;0;0;0;0]
U = 0.0009    -0.0000    -0.0014    0.0009    -0.0000    -0.0000    [将列排成了行]
         0    0    0    0    0    0    [将列排成了行]
>> P=KK*U
P = 1.0e+003 *
     3.0000    -3.0000    -0.7200    0.0000    -3.0000    0.7200    [将列排成了行]
    -0.6658    2.2012    0.6014    -2.3342    3.7988    1.1283    [将列排成了行]
```