湖南大學

HUNAN UNIVERSITY

有限元课程大作业

作业题目	使用有限元方法确定钓鱼时鱼竿偏转角
学生姓名	夏彬海
学生学号	S230200199
专业班级	机械工程 2306
学院名称	机械与运载工程学院
课程老师	王琥
学院院长	丁荣军

2024 年 2 月 3 日

目录

第一章 作业任务

在本作业中，我们的任务是利用有限元课程中学到的知识和 MATLAB 来确定钓鱼时鱼竿的偏转（如图 1 所示）。假设鱼竿的长度为 2 米。半径为 5 厘米，鱼竿的横截面为原型，其惯性为$\frac{\pi r^4}{4}$，假设杨氏模量为 100MPa

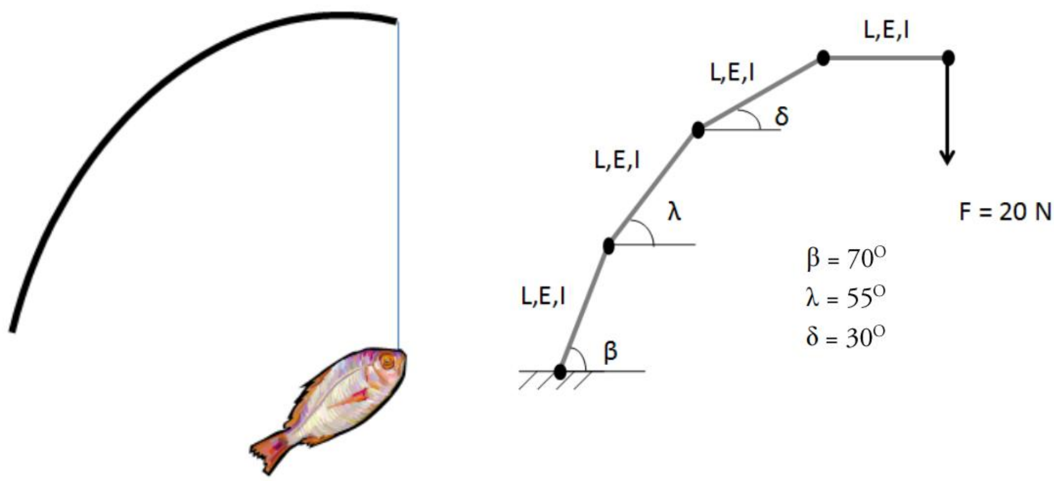

图 1 鱼竿示意图及鱼竿模型

第二章 解决方法

在本节中，对鱼竿进行离散化，以构建包括 4 个梁单元的模型，如图所示 2 所示。然后对这些梁进行了受力分析，推导出了鱼竿的整体刚度矩阵。在已知边界值的情况下，使用 MATLAB（MathWorks，Natick，MA，USA）进行计算。使用 HYPERMESH（Altair，Troy，MI，USA）和 OPTISTRUCT（Altair、Troy，MI，USA）进行静力学模拟，验证了计算结果。

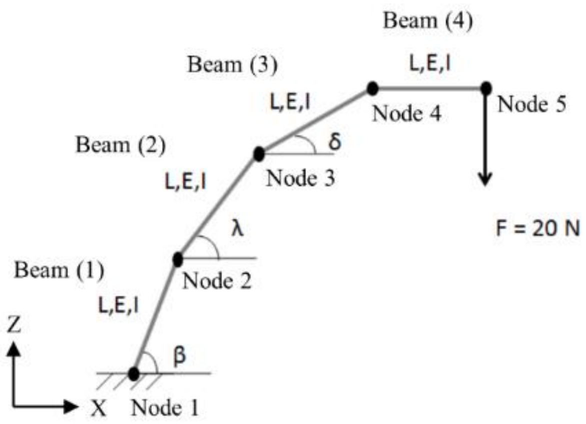

图 2 包含 4 个梁单元的鱼竿模型

第三章 代数推理过程

3.1 元素的力分析

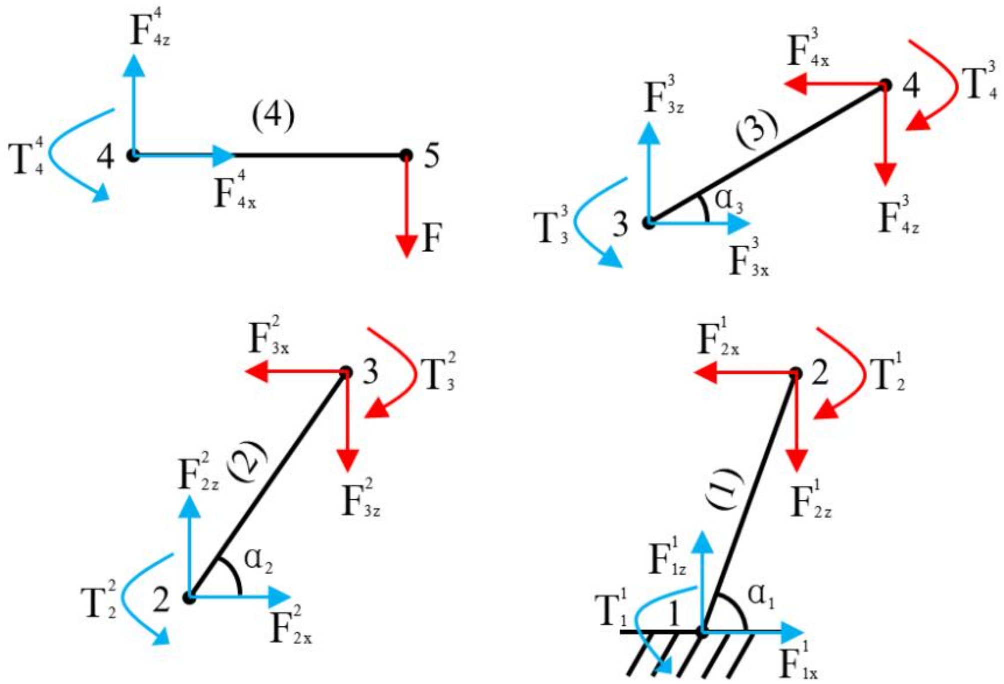

图 3 梁元件的力分析

为了便于后续的解决，我们对鱼竿模型中的每个梁元件进行了力分析。对于梁 4，力平衡为

$$\begin{cases} F_{4x}^4 = 0 \\ F_{4z}^4 + F = 0 \\ T_4^4 + F \cdot L_4 = 0 \end{cases} \tag{3-1}$$

其中 L 是梁的长度，F 是线性力，T 是力矩。对于力和力矩，下标表示节点数，上标表示元素数。对于梁 3、2 和 1，力平衡为

$$\begin{cases} F_{ix}^i + F_{(i+1)x}^i = 0 \\ F_{iz}^i + F_{(i+1)z}^i = 0 \\ T_i^i + T_{i+1}^i + F_{(i+1)z}^i \cdot L_i \cdot \cos\alpha_i - F_{(i+1)x}^i \cdot L_i \cdot \sin\alpha_i = 0 \\ F_{(i+1)x}^i + F_{(i+1)x}^{i+1} = 0 \\ F_{(i+1)z}^i + F_{(i+1)z}^{i+1} = 0 \\ T_{i+1}^i + T_{i+1}^{i+1} = 0 \end{cases} , i \in \{1,2,3\} \tag{3-2}$$

假设鱼竿平均分为 4 部分，每束梁的长度（L）为总长度的四分之一，即 500 毫米。已经给出了光束的角度（α），因此可以计算出全局坐标系中的节点力和力矩，如下表 1 所示。

表 1 全局坐标系中的节点力和力矩

Node\Force and moment	F_x / N	F_z / N	T / N·mm
Node 5	0	-20	0
Node 4	0	0	0
Node 3	0	0	0
Node 2	0	0	0
Node 1	0	20	27816.220

3.2 全局刚度矩阵

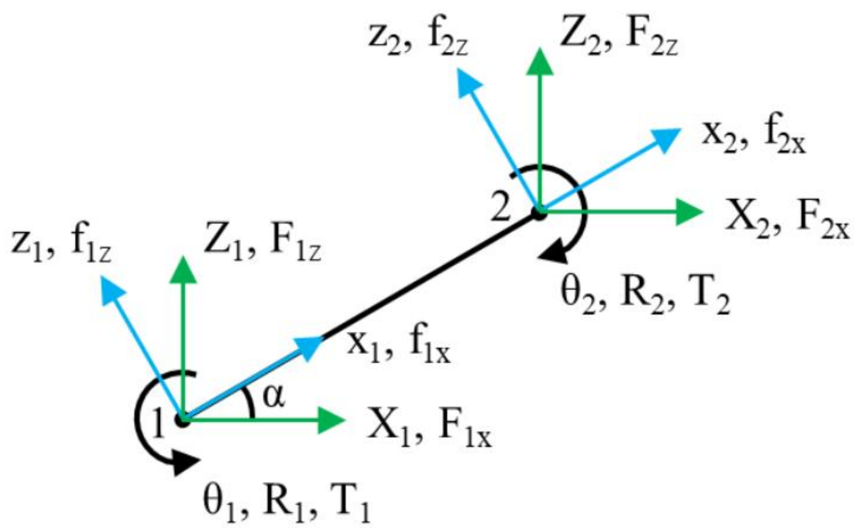

图 4 梁元件的变量

图 4 所示的单束的力平衡为

$$\begin{bmatrix} f_{1x} \\ f_{1z} \\ T_1 \\ f_{2x} \\ f_{2z} \\ T_2 \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ z_1 \\ \theta_1 \\ x_2 \\ z_2 \\ \theta_2 \end{bmatrix} \tag{3-3}$$

公式 3-3 可以缩写为

$$f = kd \tag{3-4}$$

其中k为刚度矩阵，d为变形矩阵，f为力矩阵。它们都在局部坐标系中。

变形的全局形式可以转化为局部形式

$$\begin{bmatrix} x_1 \\ z_1 \\ \theta_1 \\ x_2 \\ z_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} C & S & 0 & 0 & 0 & 0 \\ -S & C & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & C & S & 0 \\ 0 & 0 & 0 & -S & C & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Z_1 \\ R_1 \\ X_2 \\ Z_2 \\ R_2 \end{bmatrix} \tag{3-5}$$

其中 C 代表$\cos\alpha$，S 代表$\sin\alpha$。公式 3-5 可以缩写为

$$\mathbf{d} = \mathbf{TD} \tag{3-6}$$

其中，T 定义为变换矩阵，D 为全局坐标系中的变形。

对力也是如此

$$\mathbf{f} = \mathbf{TF} \tag{3-7}$$

其中 F 是全局坐标系中的力。

替换等式 3-6 和等式 3-7 年到等式 3-4 中

$$\mathbf{TF} = \mathbf{kTD} \tag{3-8}$$

因为 T 它是一个正交矩阵，它的逆矩阵和置换矩阵是相同的，公式 3-8 两边同时乘

T^T 以

$$\mathbf{F}=\mathbf{T}^{\mathrm{T}}\mathbf{kTD} \tag{3-9}$$

D 和 F 之间的关系是

$$\mathbf{F}=\mathbf{KD} \tag{3-10}$$

将公式 3-9 和公式 3-10 比较得

$$\mathbf{K}=\mathbf{T}^{\mathrm{T}}\mathbf{kT} \tag{3-11}$$

因此，利用等式可以将梁元件的局部刚度矩阵利用公式 3-11 转化为全局形式。

在该鱼杆模型中，4 个梁元件的杨氏模量 E、面积惯性矩 I、横截面面积 A 和长度 L 相同，因此编号为 i ($\mathbf{K}^{(\mathrm{i})}$) 的梁的刚度矩阵的全局形式为

$$\mathbf{K}^{(\mathrm{i})}=\frac{E}{L^3}\times$$

$$\begin{bmatrix} AL^2C_i^2+12IS_i^2 & AL^2C_iS_i-12IC_iS_i & -6ILS_i & -AL^2C_i^2-12IS_i^2 & -AL^2C_iS_i+12IC_iS_i & -6ILS_i \\ & AL^2S_i^2+12IC_i^2 & 6ILC_i & -AL^2C_iS_i+12IC_iS_i & -AL^2S_i^2-12IC_i^2 & 6ILC_i \\ & & 4IL^2 & 6ILS_i & -6ILC_i & 2IL^2 \\ & & & AL^2C_i^2+12IS_i^2 & AL^2C_iS_i-12IC_iS_i & 6ILS_i \\ & & & & AL^2S_i^2+12IC_i^2 & -6ILC_i \\ \mathit{Symmetric} & & & & & 4IL^2 \end{bmatrix} \tag{3-12}$$

$$,i\in\{1,2,3,4\}$$

其中 C_i 代表 $\cos\alpha_i$， S_i 代表 $\sin\alpha_i$， K^i 可以表示为

$$\mathbf{K}^{(\mathrm{i})}=\frac{E}{L^3}\begin{bmatrix}\mathbf{N}_{11}^{(\mathrm{i})} & \mathbf{N}_{12}^{(\mathrm{i})}\\ \mathbf{N}_{21}^{(\mathrm{i})} & \mathbf{N}_{22}^{(\mathrm{i})}\end{bmatrix},i\in\{1,2,3,4\} \tag{3-13}$$

其中 $\mathbf{N}_{11}^{(i)},\mathbf{N}_{12}^{(\mathrm{i})},\mathbf{N}_{21}^{(\mathrm{i})}\,\mathrm{and}\,\mathbf{N}_{22}^{(\mathrm{i})}$ 都为三阶方阵。

该鱼竿模型由 4 个梁元和 5 个节点组成，每根梁连接到前一个端到端，因此该模型的自由度（DOF）为 15。因此，通过结合各梁元件的刚度矩阵，可以得到钓竿的整体刚度矩阵 K，即为

$$\mathbf{K}=\frac{E}{L^3}\begin{bmatrix}\mathbf{M}_{11} & \mathbf{M}_{12} & \mathbf{O} & \mathbf{O} & \mathbf{O}\\ \mathbf{M}_{21} & \mathbf{M}_{22} & \mathbf{M}_{23} & \mathbf{O} & \mathbf{O}\\ \mathbf{O} & \mathbf{M}_{32} & \mathbf{M}_{33} & \mathbf{M}_{34} & \mathbf{O}\\ \mathbf{O} & \mathbf{O} & \mathbf{M}_{43} & \mathbf{M}_{44} & \mathbf{M}_{45}\\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{M}_{54} & \mathbf{M}_{55}\end{bmatrix} \tag{3-14}$$

其中

$$\begin{aligned}
&\mathbf{M}_{11}=\mathbf{N}_{11}^{(1)},\\
&\mathbf{M}_{55}=\mathbf{N}_{22}^{(4)},\\
&\mathbf{M}_{\mathrm{m,m}}=\mathbf{N}_{22}^{(\mathrm{m-1})}+\mathbf{N}_{11}^{(\mathrm{m})},m\in\{2,3,4\},\\
&\mathbf{M}_{\mathrm{n,n+1}}=\mathbf{M}_{\mathrm{n+1,n}}^{\mathrm{T}}=\mathbf{N}_{12}^{(\mathrm{n})},n\in\{1,2,3,4\}
\end{aligned} \tag{3-15}$$

因此，通过求解以下方程，可以得到节点的变形。

$$\begin{bmatrix} F_{1x} \\ F_{1z} \\ T_1 \\ \vdots \\ F_{5x} \\ F_{5z} \\ T_5 \end{bmatrix}=\mathbf{K}\begin{bmatrix} X_1 \\ Z_1 \\ R_1 \\ \vdots \\ X_5 \\ Z_5 \\ R_5 \end{bmatrix} \tag{3-16}$$

第四章 使用 MATLAB 计算

在本章中使用 MATLAB 编写了一个脚本来解决这个任务。该脚本使用杨氏模量、杆半径、节点和元素信息（节点号、节点坐标、元素数、元素包含的节点数）作为输入，计算节点变形并绘制变形钓杆的图形。 该代码见附录 1。

4.1 验证

为了验证计算结果，我们使用超网格方法对鱼竿模型进行了重构，如图 4 所示。根据任务描述应用约束和加载。然后使用光学进行模拟。

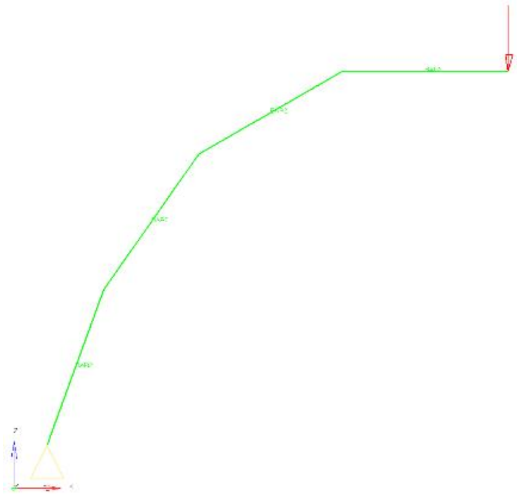

图 5 建立鱼杆模型进行验证

4.2 计算结果

本节给出了计算和仿真的结果。为了提高准确性，我们对这两个结果进行了交叉检验。

鱼竿的变形情况如图 6 所示。

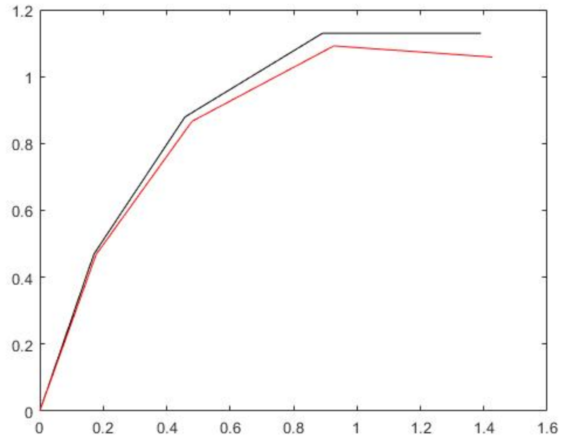

图 6 变形（红色）和未变形（黑色）钓竿

通过 MATLAB 计算出的节点变形如表 2 所示。

表 2 全局坐标系中的节点变形

Node\Deformation	Displacement/mm		Rotation/rad
	X	Z	
Node 5	36.0874	-71.0006	-0.0682
Node 4	36.0874	-37.7447	-0.0631
Node 3	21.9546	-13.2533	-0.0485
Node 2	6.3793	-2.3346	-0.0266
Node 1	0	0	0

4.3 验证结果

第 4.1 节构建的静力学分析模型采用光学结构法进行求解。仿真结果如图 5 所示，该单位为毫米级。

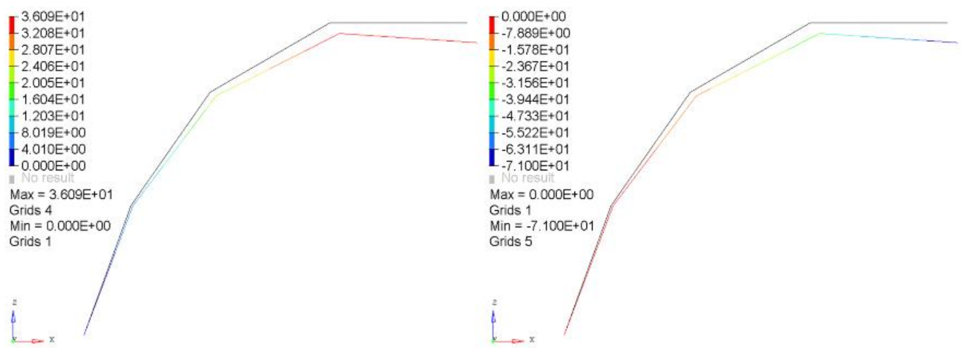

图 7 X 方向（左）和 Z 方向（右）上的位移

由光学结构给出的节点位移如表 3 所示。

表 3 节点位移，单位为毫米

Node\Displacement	X-direction	Z-direction
Node 5	36.0873642	-71.00057983
Node 4	36.0873642	-37.7447319
Node 3	21.95457268	-13.2532711
Node 2	6.379256249	-2.334592342
Node 1	0	0

第五章 结论

本文介绍了用于解决该任务的方法。对 MATLAB 脚本进行计算，并通过超网格和光学结构进行仿真验证了结果的准确性。

用 MATLAB 计算的鱼竿变形与光学结构提供的结果一致。因此，本报告中提出的结果可以被认为是准确的。

附录 1-用于计算的 MATLAB 代码

```
%---main---
clc
clear;

E = 100; % Young's modulus in MPa,
r = 50; % radius of the rod in mm,
I = pi * r^4 / 4; % area moment of inertia in mm^4,
A = pi * r^2; % cross-section area in mm^2,

% node information. 1st column is node number,
% 2nd, 3rd and 4th column are X, Y, Z coordinate.
node = [1 0 0 0;
2 171.010 0 469.846;
3 457.798 0 879.422;
4 890.811 0 1129.422;
5 1390.811 0 1129.422];

% element information. 1st column is element number,
% 2nd and 3rd column are the numbers of the nodes it includes.
ele = [1 1 2;
2 2 3;
3 3 4;
4 4 5];

n_ele = length(ele(:, 1)); % total number of the elements
L = zeros(1, n_ele); % beam length
ang = zeros(1, n_ele); % beam horizontal angle

for i = 1 : n_ele
L(i) = sqrt((node(ele(i, 3), 2) - node(ele(i, 2), 2))^2 ...
+ (node(ele(i, 3), 4) - node(ele(i, 2), 4))^2);
ang(i) = atand((node(ele(i, 3), 4) - node(ele(i, 2), 4)) ...
/ (node(ele(i, 3), 2) - node(ele(i, 2), 2)));
end

% obtain the global stiffness matrix of the rod
dof = length(node(:, 1)) * 3; % degree of freedom
F = zeros(dof, 1) * nan; % global force vector
D = ones(dof, 1) * nan; % deformation vector in global form
K = zeros(dof); % global stiffness matrix
```

```
for i = 1 : n_ele
k_ele = transform(E, I, A, L(i), ang(i));
K = combine(K, k_ele, ele(i, 2), ele(i, 3));
end

% Force boundary conditions, units N, mm.
F5x = 0; F5z = -20; T5 = 0;
F4x = 0; F4z = 0; T4 = 0;
F3x = 0; F3z = 0; T3 = 0;
F2x = 0; F2z = 0; T2 = 0;
F(4 : end) = [F2x F2z T2 F3x F3z T3 F4x F4z T4 F5x F5z T5];

% Displacement boundary conditions, units mm, degree.
D1x = 0; D1z = 0; M1 = 0;
D(1 : 3) = [D1x D1z M1];

% Solve the equation
index = D ~= 0;
D(index) = K(index, index) \ F(index);
F = K * D;

% print the nodal deformation
disp('The nodal deformations are:');
disp(D);

% plot the deformed and undefromed fishing rod
A = 0.001 * node(:, 2);
B = 0.001 * node(:, 4);
A1 = zeros(1, length(A));
B1 = zeros(1, length(B));
for i = 1 : length(D(:, 1))/3
A1(i) = A(i) + 0.001 * D(3 * i - 2, 1);
B1(i) = B(i) + 0.001 * D(3 * i - 1, 1);
end
plot(A, B, '-k', A1, B1, '-r');
axis([0, 1.6, 0, 1.2]);
set(gca, 'XTick', [0 : 0.2 : 1.6]);
set(gca, 'YTick', [0 : 0.2 : 1.2]);

function K_ele = transform(E, I, A, L, ang)
% This is to transform local stiffness matrix into global form.
% E is Young's modulus,
% I is area moment of inertia,
% A is cross-section area,
```

```
% L is length,
% ang is horizontal angle.
% the size of the beam stiffness matrix is 6*6.

% local stiffness matrix
k1 = A * E / L;
k2 = E * I / L^3;
k_local = [k1, 0, 0, -k1, 0, 0;
0, 12 * k2, 6 * k2 * L, 0, -12 * k2, 6 * k2 * L;
0, 6 * k2 * L, 4 * k2 * L^2, 0, -6 * k2 * L, 2 * k2 * L^2;
-k1, 0, 0, k1, 0, 0;
0, -12 * k2, -6 * k2 * L, 0, 12 * k2, -6 * k2 * L;
0, 6 * k2 * L, 2 * k2 * L^2, 0, -6 * k2 * L, 4 * k2 * L^2];

% transformation matrix
c = cosd(ang);
s = sind(ang);
T = [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];

% transform local stiffness matrix to global form
K_ele = (T') * k_local * T;

end

function k_t = combine(k_t, k_ele, n1, n2)
% This is to obtain the global stiffness matrix.
% stiffness matrix of a beam with node i and j are used as input.

d = zeros(1, 6);
d(1) = 3 * n1 - 2;
d(2) = 3 * n1 - 1;
d(3) = 3 * n1;
d(4) = 3 * n2 - 2;
d(5) = 3 * n2 - 1;
d(6) = 3 * n2;
for m = 1 : 6
for n = 1 : 6
k_t(d(m), d(n)) = k_t(d(m), d(n)) + k_ele(m, n);
end
```

```
end

end
```