学　　号 2016011560

# Foundation of Finite Elements Method
# Assignment FEM model


院 (系) 名 称：　航天航空学院

专 业 名 称：　工程力学

学 生 姓 名：　易泽吉


二〇一八年三月

## 0.1 Euler-Bernoulli Beam 单元

### 0.1.1 单元构造

每个梁单元有两个节点首先确定梁单元的位移向量与载荷向量。规定沿梁的轴线方向为 x 方向，单元的位移向量与载荷向量可以表示为：

$$d^e = \begin{pmatrix} u_{x1} & u_{y1} & u_{z1} & u_{x1} & u_{y1} & u_{z1} & u_{x2} & u_{y2} & u_{z2} & \theta_{x2} & \theta_{y2} & \theta_{y3} \end{pmatrix}^T \tag{1}$$

$$f^e = \begin{pmatrix} F_{x1} & F_{y1} & F_{z1} & M_{x1} & M_{y1} & M_{z1} & F_{x2} & F_{y2} & F_{z2} & M_{x2} & M_{y3} \end{pmatrix}^T \tag{2}$$

其中 $u$ 表示位移，$\theta$ 表示转角，$f$ 表示剪力，$M$ 表示弯矩，下标中 1，2 分别表示两端节点。

对于梁单元有四个独立的模态，第一个为轴向拉伸，与杆单元相同，第二个为沿轴扭转，第三四个分别为垂直于轴线的两个方向的弯曲，单元的刚度矩阵如下，矩阵中各元素的值不需要高斯积分，可直接导出。

$$K^e = \begin{pmatrix} \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{l} & 0 & 0 & 0 & 0 & 0 \\ & \frac{12EI_z}{l^3} & 0 & 0 & 0 & \frac{6EI_z}{l^2} & 0 & -\frac{12EI_z}{l^3} & 0 & 0 & 0 & \frac{6EI_z}{l^2} \\ & & \frac{12EI_y}{l^3} & 0 & -\frac{6EI_z}{l^2} & 0 & 0 & 0 & -\frac{12EI_y}{l^3} & 0 & -\frac{6EI_z}{l^2} & 0 \\ & & & \frac{GJ_z}{I} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ_z}{I} & 0 & 0 \\ & & & & \frac{4EI_y}{l} & 0 & 0 & 0 & \frac{6EI_y}{l} & 0 & \frac{2EI_y}{l} & 0 \\ & & & & & \frac{4EI_z}{l} & 0 & -\frac{6EI_z}{l} & 0 & 0 & 0 & \frac{2EI_z}{l} \\ & & \text{Symmetry} & & & & \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & \frac{12EI_z}{l^3} & 0 & 0 & 0 & -\frac{6EI_z}{l^2} \\ & & & & & & & & \frac{12EI_y}{l^3} & 0 & \frac{6EI_y}{l^2} & 0 \\ & & & & & & & & & \frac{GJ_z}{I} & 0 & 0 \\ & & & & & & & & & & \frac{4EI_y}{l} & 0 \\ & & & & & & & & & & & \frac{4EI_z}{l} \end{pmatrix} \tag{3}$$

在写出单元刚度阵以及位移向量之后，我们就得到了单元的控制方程：

$$K^e d^e = f^e \tag{4}$$

这样就可以准备将单元刚度阵组装到总体刚度阵，当然在此之前需要将单元刚度阵做坐标变换，这源自于在全局的坐标系，真实的单元方程是：

$$K^e R d^e = R f^e \tag{5}$$

因此有

$$K^{global} = R^T K^{element} R \tag{6}$$

其中 R 为 12*12 的矩阵，其中有四个分块，每个分块都为方向余弦阵

$$R = \begin{bmatrix} A & & & \\ & A & & \\ & & A & \\ & & & A \end{bmatrix} \tag{7}$$

其中 A=

$$\hat{A} = \begin{bmatrix} a_x a'_x & a_x a'_y & a_x a'_z \\ a_y a'_x & a_y a'_y & a_y a'_z \\ a_z a'_x & a_z a'_y & a_z a'_z \end{bmatrix} \tag{8}$$

将如此经过正交变换的单元刚度阵组装到总刚度阵即可

### 0.1.2 梁单元的输入规定

梁单元的在单元类型中编号为 4，需要输入的材料性质包括基本的杨氏模量 E，泊松比 $\mu$，另外本工程项目中使用的是空心矩形截面梁，为保证对称性，即保证不移轴，上下壁面的厚度需要相同，左右壁面的厚度需要相同，因此依次输入的参数是：矩形的长和宽（a,b），上下壁面的厚度，左右壁面的厚度，以及一组单位向量（n1,n2,n3）来表征局部坐标系的 y 轴在全局坐标系中的方向。考虑到桥梁中的梁单元都存在于 xz 平面，该向量可以被置为（0，1，0）

### 0.1.3 Patch Test

首先对于梁进行不规则的划分，如下图所示：

梁的总长度为 1m, 共划分为 4 个单元，5 个节点，分别在 $x = 0, 0.2, 0.3, 0.5, 1$ 处。

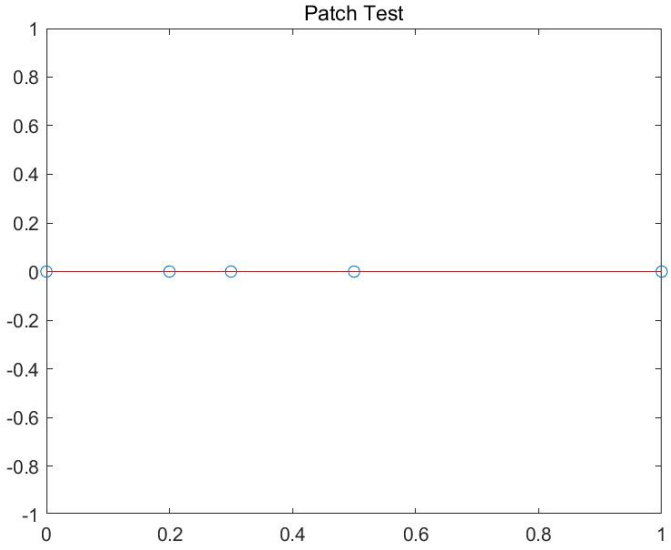



Figure 1　Patch Test

由于梁有四个互相独立解耦的模态，对其分别约束测试。

首先验证 y 方向挠度，即 z 方向弯曲，如下构造精确解。本质边界条件为两端简支，自然边界条件为两边施加等大反向弯矩，梁处于纯弯曲状态且弯矩相等，位移的精确解为：

$$w = x^2 - x, \theta = 2x - 1 \tag{9}$$

分片试验结果如下，

Table 1　**DISPLACEMENT-WZ**

| NODE | X-D | Y-D | Z-D | $\theta_X$-D | $\theta_Y$-D | $\theta_Z$-D |
|---|---|---|---|---|---|---|
| 1 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 0.000e+00 | -1.000e+01 |
| 2 | 0.000e+00 | -1.600e+01 | 0.000e+00 | 0.000e+00 | 0.000e+00 | -6.000e+00 |
| 3 | 0.000e+00 | -2.100e+01 | 0.000e+00 | 0.000e+00 | 0.000e+00 | -1.000e+01 |
| 4 | 0.000e+00 | -2.500e+01 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 1.237e-15 |
| 5 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 1.000e+01 |

之后验证 z 方向挠度，y 方向弯曲，边界条件及载荷相同，位移的精确解也相同，节点和分片方式相同，结果如下

$$w = x^2 - x, \theta = 2x - 1 \tag{10}$$

分片试验结果如下，

Table 2 **DISPLACEMENT-WZ**

| NODE | X-D | Y-D | Z-D | $\theta_X$-D | $\theta_Y$-D | $\theta_Z$-D |
|---|---|---|---|---|---|---|
| 1 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 0.000e+00 | -1.000e+01 | 0.000e+00 |
| 2 | 0.000e+00 | 0.000e+00 | -1.600e+01 | 0.000e+00 | -6.000e+00 | 0.000e+00 |
| 3 | 0.000e+00 | 0.000e+00 | -2.100e+01 | 0.000e+00 | -1.000e+01 | 0.000e+00 |
| 4 | 0.000e+00 | 0.000e+00 | -2.500e+01 | 0.000e+00 | 9.877e-16 | 0.000e+00 |
| 5 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 1.000e+01 | 0.000e+00 |

x 方向位移与杆单元的 patch test 相同，$\theta_x$ 方向应满足线性收敛率，结果如下

Table 3 **DISPLACEMENT-THETAX**

| NODE | X-D | Y-D | Z-D | $\theta_X$-D | $\theta_Y$-D | $\theta_Z$-D |
|---|---|---|---|---|---|---|
| 1 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 0.000e+00 |
| 2 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 2.000e+00 | 0.000e+00 | 0.000e+00 |
| 3 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 3.000e+00 | 0.000e+00 | 0.000e+00 |
| 4 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 5.000e+00 | 0.000e+00 | 0.000e+00 |
| 5 | 0.000e+00 | 0.000e+00 | 0.000e+00 | 1.000e+01 | 0.000e+00 | 0.000e+00 |

### 0.1.4 收敛性分析

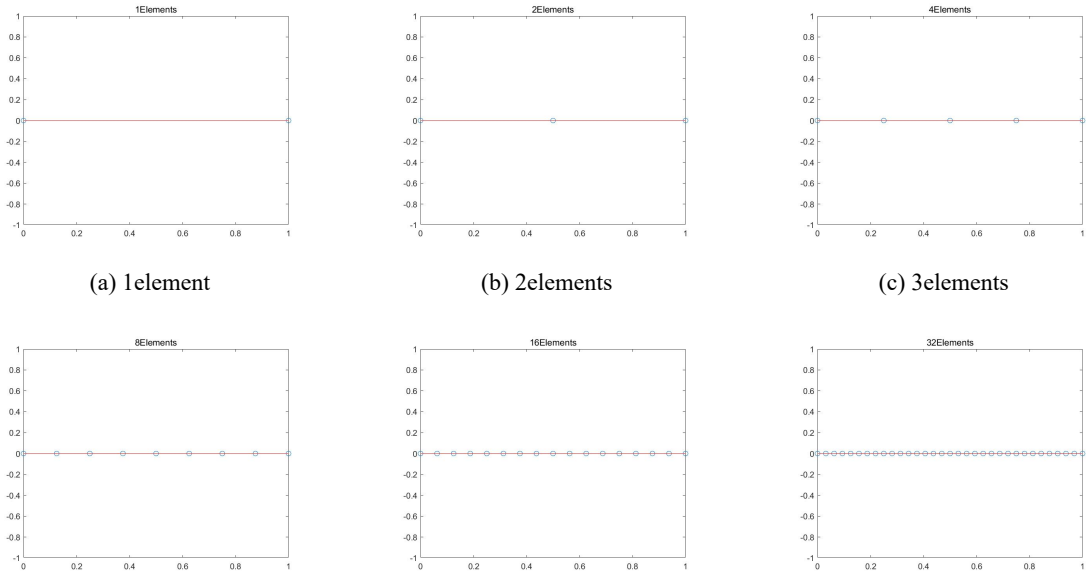
选取悬臂梁，调整重力大小，即均布载荷大小，构造出精确解：梁单元可以有效重构三次位移场，因此，带有均布载荷的悬臂梁可以较好的反应误差。

$$w = 0.01x^4 \tag{11}$$

误差范数采用位移范数，可以表示为

$$\|e\|^2 = \int A\left(u - u^h\right)^2 dx \tag{12}$$

计算收敛率时网格划分如下：




(a) 1element (b) 2elements (c) 3elements

(d) 4elements (e) 5elements (f) 6elements

Figure 2 Convergence Mesh

最终收敛率结果如图所示:

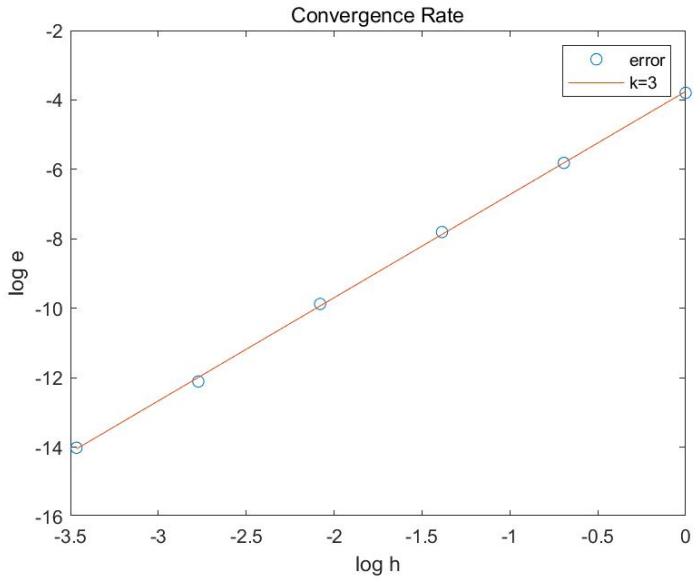



Figure 3 Convergence Rate

## 0.2 Timoshenko Beam

### 0.2.1 Timoshenko Beam 的基本假设

Euler-Bernoulli 梁适用于薄梁，忽略横向剪切变形，横向剪切应力由平衡方程确定，不适用于有效长度较短的梁或组合梁。Timoshenko 梁仍然假设与梁轴线垂直的横截面在变形后保持平面（假设为刚性横截面平面），但放松了一些约束，即放弃了直法线假设，采用剪应变沿截面均匀分布的假设，在一定程度上解决了欧拉伯努利梁的问题。

### 0.2.2 Timoshenko Beam 的控制方程

从变分原理出发有，系统的势能泛函为

$$\Pi_P = \int_0^l \frac{1}{2} EI \left(\frac{\mathrm{d}\theta}{\mathrm{d}x}\right)^2 \mathrm{d}x + \int_0^l \frac{1}{2}\frac{GA}{k}\gamma^2 \mathrm{d}x - \int_0^l qw \mathrm{d}x \tag{13}$$

其中，

$$\varepsilon_x = -z\frac{\mathrm{d}\theta}{\mathrm{d}x} \quad \gamma = \frac{\mathrm{d}w}{\mathrm{d}x} - \theta \quad \kappa = -\frac{\mathrm{d}\theta}{\mathrm{d}x} \tag{14}$$

按照有限元格式离散之后可以写成

$$\begin{aligned}
\Pi_P &= \int_0^l \frac{1}{2} EI \left(\frac{\mathrm{d}\theta}{\mathrm{d}x}\right)^2 \mathrm{d}x + \int_0^l \frac{1}{2}\frac{GA}{k}\gamma^2 \mathrm{d}x - \int_0^l qw \mathrm{d}x \\
&= \sum_e \left[ \int_{\Omega^e} \frac{1}{2} EI \left(\frac{\mathrm{d}\theta}{\mathrm{d}x}\right)^2 \mathrm{d}x + \int_{\Omega^e} \frac{1}{2}\frac{GA}{k}\gamma^2 \mathrm{d}x - \int_{\Omega^e} qw \mathrm{d}x \right] \\
&= \frac{1}{2}\sum_e \left(\boldsymbol{d}^e\right)^{\mathrm{T}} \left[\left(\boldsymbol{K}_b^e + \boldsymbol{K}_s^e\right)\boldsymbol{d}^e - f^e\right]
\end{aligned} \tag{15}$$

其中

$$\begin{aligned}
\boldsymbol{K}_b^e &= \int_{\Omega^e} EI \boldsymbol{B}_b^{e\mathrm{T}} \boldsymbol{B}_b^e \mathrm{d}x \\
\boldsymbol{K}_s^e &= \int_{\Omega^e} \frac{GA}{k} \boldsymbol{B}_s^{e\mathrm{T}} \boldsymbol{B}_s^e \mathrm{d}x \\
\boldsymbol{f}^e &= \int_{\Omega^e} \boldsymbol{N}^{e\mathrm{T}} \begin{bmatrix} q \\ 0 \end{bmatrix} \mathrm{d}x
\end{aligned} \tag{16}$$

根据最小势能原理有

$$\delta\Pi_P = 0 \tag{17}$$

从而

$$\begin{aligned}
EI_z \tfrac{d\kappa_z}{dx} + \tfrac{GA}{k}\gamma_{xy} + \overline{M}_{zk}\delta\left(x - x_k\right) = 0 \\
\tfrac{GA}{k}\tfrac{d\gamma_{xy}}{dx} + \overline{q} + \overline{N}_{yj}\delta\left(x - x_j\right) = 0
\end{aligned} \tag{18}$$

即

$$\delta\boldsymbol{d}^{\mathrm{T}} \left( \sum_e \boldsymbol{L}^{e\mathrm{T}} \left(\boldsymbol{K}_b^e + \boldsymbol{K}_s^e\right) \boldsymbol{L}^e \boldsymbol{d} - \sum_e \boldsymbol{L}^{e\mathrm{T}} \boldsymbol{f}^e \right) = 0 \quad \forall \boldsymbol{d}_F \tag{19}$$

### 0.2.3 有限元离散与插值函数

我们选取的插值方式为位移转角的一致插值，即考虑两种模式，一种为欧拉伯努利梁，只有弯曲位移，没有剪切位移，另一种为简单修正，只有剪切位移，没有弯曲位移：

$$\begin{aligned} a &= a_b + a_s \\ a_i &= (u_i, v_i, w_i, \varphi_i, \theta_{yi}, \theta_{zi})^T \\ a_b &= (a_{b1}, a_{b2})^T \\ a_s &= (a_{s1}, a_{s2})^T \\ a_{bi} &= (u_i, v_{bi}, w_{bi}, \varphi_i, \theta_{yi}, \theta_{zi})^T \\ a_{si} &= (0, v_{si}, w_{si}, 0, 0, 0)^T \end{aligned} \tag{20}$$

对于欧拉伯努利梁的模式，弯曲所产生的变形与转角有如下关系：

$$\begin{aligned} \frac{dv_{bi}}{dx} &= \theta_{zi} \\ \frac{dw_{bi}}{dx} &= -\theta_{yi} \end{aligned} \tag{21}$$

下面我们选取 y，z 中的任意方向，进行插值和推导，对于欧拉伯努利梁部分，采用 Hermite 插值

$$v_b = N_1 v_{b1} + N_2 \theta_1 + N_3 v_{b2} + N_4 \theta_4 \tag{22}$$

对于剪切部分，采用两点线性插值

$$v_s = N_5 v_{s1} + N_6 v_{s2} \tag{23}$$

其中 $\xi$ 为母单元坐标值

$$\begin{aligned} N_1 &= 1 - 3\xi^2 + 2\xi^3 & N_2 &= (\xi - 2\xi^2 + \xi^3) l \\ N_3 &= 3\xi^2 - 2\xi^3 & N_4 &= (\xi^3 - \xi^2)\, l \\ N_5 &= 1 - \xi & N_6 &= \xi \end{aligned} \tag{24}$$

将插值函数代入有

$$\boldsymbol{B}_b^e = \begin{bmatrix} \frac{\mathrm{d}N_1}{\mathrm{d}x} & \cdots & \frac{\mathrm{d}N_n}{\mathrm{d}x} \end{bmatrix} \tag{25}$$

由于单元内部平衡方程 $Q = \frac{dM}{dx}$，欧拉伯努利梁部分与剪力部分有如下关系

$$\begin{pmatrix} v_{b2} - v_{b1} \\ v_{s2} - v_{s1} \end{pmatrix} = \begin{pmatrix} \frac{1}{1+b_z} & \frac{lb_z}{2(1+b_z)} \\ \frac{b_z}{1+b_z} & -\frac{lb_z}{2(1+b_z)} \end{pmatrix} \begin{pmatrix} v_2 - v_1 \\ \theta_{z1} + \theta_{z2} \end{pmatrix} \tag{26}$$

其中，$b_z := \frac{12EI_z k}{GAl^2}$，对于矩形截面梁的近似有 $k = \frac{6}{5}$。整理之后可以得到，$\boldsymbol{Kd} = \boldsymbol{F}$ 其中

$$\boldsymbol{K} = \frac{EI_z}{(1+b_z)\,l^3}\begin{pmatrix} 12 & -12 & 6l & 6l \\ & 12 & -6l & -6l \\ & & (4+b_z)\,l^2 & (2-b_z)\,l^2 \\ [\text{symmetry}] & & & (4+b_z)\,l^2 \end{pmatrix} \tag{27}$$

考虑所有分量的问题，同时加入扭转以及拉压方向的问题

$$\boldsymbol{a} = \boldsymbol{a}_b + \boldsymbol{a}_s \tag{28}$$

有如下刚度阵

$$\begin{pmatrix} \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{l} & 0 & 0 & 0 & 0 & 0 \\ & \frac{12EI_z}{(1+b_z)l^3} & 0 & 0 & 0 & \frac{(1+b_z)6EI_z}{l^2} & 0 & -\frac{12EI_z}{l^3} & 0 & 0 & 0 & \frac{6EI_z}{(2-b_z)l^2} \\ & & \frac{12EI_y}{(1+b_y)l^3} & 0 & -\frac{6EI_z}{(1+b_y)l^2} & 0 & 0 & 0 & -\frac{12EI_y}{(1+b_y)l^3} & 0 & -\frac{6EI_z}{(1+b_y)l^2} & 0 \\ & & & \frac{GJ_z}{I} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ_z}{I} & 0 & 0 \\ & & & & \frac{(4+b_y)EI_y}{(1+b_y)l} & 0 & 0 & 0 & \frac{6EI_y}{(1+b_y)l} & 0 & \frac{(2-b_y)EI_y}{(1+b_y)l} & 0 \\ & & & & & \frac{(4+b_z)EI_z}{(1+b_z)l} & 0 & -\frac{6EI_z}{(1+b_z)l} & 0 & 0 & 0 & \frac{(2-b_z)EI_z}{(1+b_z)l} \\ & & & & & & \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & \frac{12EI_z}{(1+b_z)l^3} & 0 & 0 & 0 & -\frac{6EI_z}{(1+b_z)l^2} \\ & & & & & & & & \frac{12EI_y}{(1+b_y)l^3} & 0 & \frac{6EI_y}{(1+b_y)l^2} & 0 \\ & & & & & & & & & \frac{GJ_z}{I} & 0 & 0 \\ & & & & & & & & & & \frac{(4-b_y)EI_y}{(1+b_y)l} & 0 \\ & & & & & & & & & & & \frac{(4-b_y)EI_z}{(1+b_z)l} \end{pmatrix} \tag{29}$$

Symmetry

同时单元应力的具体计算公式如下：

$$\begin{aligned} \sigma_{xx} &= E\varepsilon_{xx} = \frac{E\,(u_2-u_1)}{l} \\ \sigma_{xy} &= \frac{G}{k}\gamma_{xy} \\ &= \frac{G}{kl}\frac{b_z}{1+b_z}\left((v_2-v_1) - \frac{1}{2}\,(\theta_{z1}+\theta_{z2})\,l\right) \\ \sigma_{xz} &= \frac{G}{k}\gamma_{xz} \\ &= \frac{G}{kl}\frac{b_y}{1+b_y}\left((w_2-w_1) + \frac{1}{2}\,(\theta_{y1}+\theta_{y2})\,l\right) \end{aligned} \tag{30}$$

### 0.2.4 程序实现

在 Stappp 程序中新增了 Timoshenko 单元，在基类 Material 的基础上派生了 CTimoshenkoMaterial 类，Timoshenko 梁的裁量性质与 Beam 相同，同时在程序实

现时需要注意坐标的转换，我们得出的刚度阵是在单元坐标系中得出的，我们需要将其转换到全局坐标系组装。与 Beam 单元类似，有

$$K^e R d^e = R f^e \tag{31}$$

因此有

$$K^{global} = R^T K^{element} R \tag{32}$$

其中 R 为 12*12 的矩阵，其中有四个分块，每个分块都为方向余弦阵

$$R = \begin{bmatrix} A & & & \\ & A & & \\ & & A & \\ & & & A \end{bmatrix} \tag{33}$$

其中 A=

$$\hat{A} = \begin{bmatrix} a_x a'_x & a_x a'_y & a_x a'_z \\ a_y a'_x & a_y a'_y & a_y a'_z \\ a_z a'_x & a_z a'_y & a_z a'_z \end{bmatrix} \tag{34}$$

将如此经过正交变换的单元刚度阵组装到总刚度阵即可

### 0.2.5 Patch Test

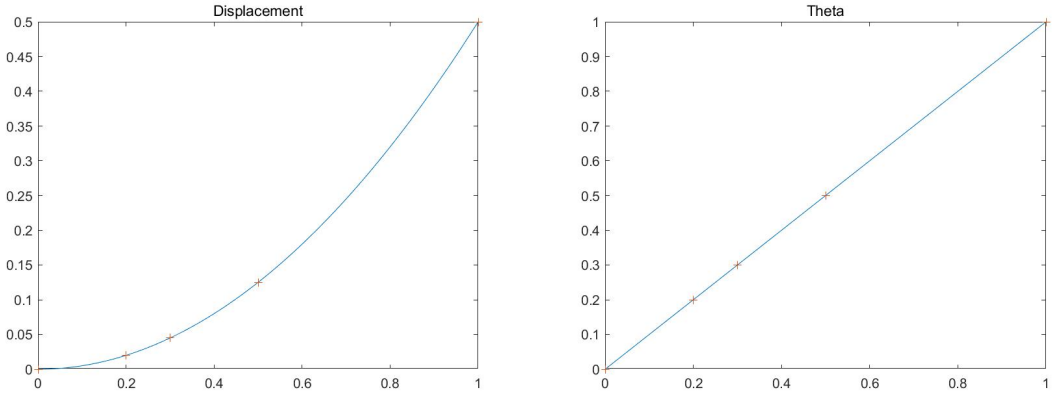
对于 Timoshenko 单元，我们考虑其常应变模态为常曲率与常剪应变状态，对于位移与转角有：

$$\begin{aligned} v &= \frac{1}{2}ax^2 + cx + d \\ \theta_z &= ax + b \end{aligned} \tag{35}$$

注意到此处转角并非单纯位移的积分，因为剪力项对于位移有线性贡献儿对于转角无贡献。这一状态实际上对应于一端集中剪力载荷与弯矩载荷共同作用的悬臂梁。此处选取两种精确解进行 patch test 的测试。首先为常弯曲状态，即悬臂梁一端施加弯矩。具有如下精确解

$$w = 0.5x^2, \theta = x \tag{36}$$

分片试验结果如下：



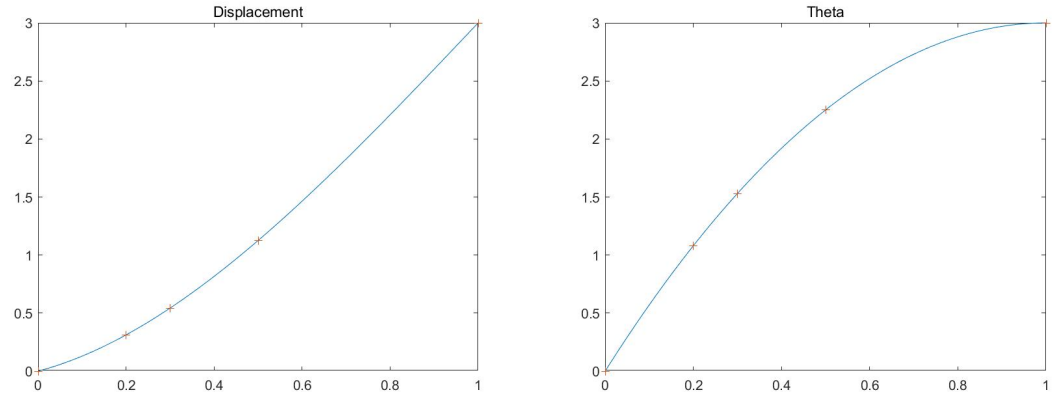

(a) 1element (b) 2elements

Figure 4 Timoshenko Beam Patch Test 1

其次对于纯剪应力状态，精确解为

$$w = -x^3 + 3x^2 + x, \theta = -3x^2 + 6x \tag{37}$$

分片试验结果如下：




(a) 1element (b) 2elements

Figure 5 Timoshenko Beam Patch Test 2

### 0.2.6 收敛率验证

选取均布载荷，对于转角验证收敛率，精确解为三次函数，选取位移范数，收敛率如图

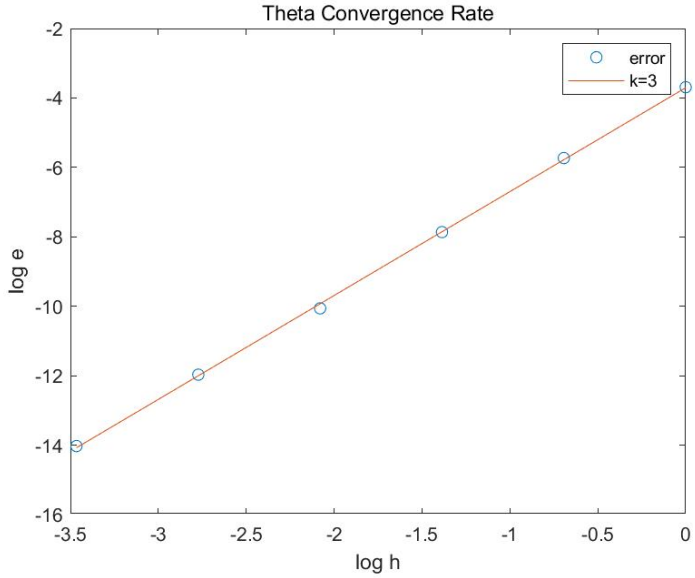



Figure 6　Convergence Rate

## 0.3　稀疏矩阵求解器

我们选取了 MKL 库中的稀疏矩阵求解器 PARDISO，PARDISO 的基本原理是根据稀疏特性优化的 LDLT 分解算法，基于共享内存，同时具有多线程的并行性，是一种 Left-looking Right-looking 结合的算法。

### 0.3.1　稀疏矩阵的存储方式

CSR 是最流行和通用的格式，可为高性能架构中的结构化和非结构化稀疏矩阵提供出色的压缩比。具有 CSR 格式的向量运算在 CPU 上实现时显示出良好的性能改进，并且所有算法（如 BLAS，LAPACK 和 CUSparse）均支持此格式。它使用三个 1 维数组，一个用于保存非零值，第二个用于保存每行非零值的数量，第三个用于保存非零值的列索引。此格式的大小主要取决于矩阵中的非零值的数量。CSR 较好的反映了矩阵的稀疏特性，并且受每行非零值的分布的影响。

$$A_{\text{man}} = \begin{bmatrix} 0 & 4 & 0 & 7 & 0 \\ 2 & 0 & 3 & 0 & 6 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 6 & 0 \end{bmatrix} \tag{38}$$

$$Data = \begin{bmatrix} 4 & 7 & 2 & 3 & 6 & 5 & 2 & 1 & 6 \end{bmatrix} \tag{39}$$

$$Column-indices=\begin{bmatrix}1&3&0&2&4&1&4&0&3\end{bmatrix} \tag{40}$$

$$Ptr=\begin{bmatrix}0&2&5&6&7&9\end{bmatrix} \tag{41}$$

CSR 占据的存储空间为

$$CSR_{\text{storage}}=2\times NZV+m+1 \tag{42}$$

### 0.3.2 稀疏矩阵的具体实现

在原有的 STAP++ 的 LDLT 求解器中适配的是 Skyline 格式的矩阵，对于 Pardiso 求解器，适配上述的 CSR 格式的求解器。CSR 格式需要矩阵每一行的长度来分配内存，但在组装过程结束前并不能完全清楚每一行的长度，因此这设计到可变长度的存储非零元素的索引值，因此我们考虑到使用 c++ 的 <vector> 容器来进行存储。

在最终使用 vector 之前我们对比了其他 c++ 标准 STL 容器。他们具有一些共同优点, 如接口简单，排序速度快，使用方便，内存分配科学。但在这一问题中，vector 的特点适配的较好。vector 是典型的序列容器，唯一可以和标准 C 兼容的 stl 容器，任意元素的读取、修改具有常数时间复杂度，在序列尾部进行插入、删除是常数时间复杂度，但在序列的头部插入、删除的时间复杂度是 O(n)，可以在任何位置插入新元素，有随机访问功能，插入删除操作需要考虑。排序的时间复杂度为 O(NlogN)。在程序的编写过程中，首先扫描全部元素，标记所有非零元的位置，为 CSR 求解器预先分配内存之后对存储的索引值进行排序，去掉重复的索引值，之后分配矩阵元素值的存储空间。

### 0.3.3 PARDISO 的参数调节

根据系统的环境可以调节系统的并行线程数，在 16 核的系统上性能得到了大幅度的提升。同时打开 OOC 模式开关，MKL 带有 IC/OOC 模式的开关，IC 指的是 IN-CORE 的存储，OOC 指的是 out-of-core 的存储方式，通俗的说 IC 模式将原本全部存储在内存中的文件中超出允许使用的内存上限的部分存储在硬盘中，因此解决了内存溢出的问题。OOC 可以通过将矩阵因子保存在磁盘上的文件中来解决非常大的问题，与 IC 相比，这减小了对于内存的需求。通过 MKL-PARDISO-OOC-MAX-CORE-SIZE 这一变量规定 OOC 模式 PARDISO 能够使用的最大内存量。同时在工作目录下配置 pardiso-ooc.cfg 配置文件。我们在本地电脑上对于第二个算例进行了 OOC 的尝试，配置文件为：

ooc-max-core-size got from config file=200

ooc-max-swap-size got from config file=0

ooc-keep-file got from config file=1

显示成功读取相关配置文件最终结果如下：

Table 4 **PARDISO STATISTICS**

| TIMES | | |
|---|---|---|
| Time spent in calculations of symmetric matrix portrait (fulladj) | : | 0.115191 s |
| Time spent in reordering of the initial matrix (reorder) | : | 0.394719 s |
| Time spent in symbolic factorization (symbfct) | : | 0.589268 s |
| Time spent in data preparations for factorization (parlist) | : | 0.013076 s |
| Time spent in copying matrix to internal data structure (A to LU) | : | 0.000000 s |
| Factorization: Time for writing to files | : | 0.000000 |
| Factorization: Time for reading from files | : | 0.000000 |
| Time spent in factorization step (numfct) | : | 76.839788 s |
| Solution: Time for reading from files | : | 0.505685 |
| Time spent in direct solver at solve step (solve) | : | 0.623790 s |
| Time spent in allocation of internal data structures (malloc) | : | 3.694331 s |
| Time spent in additional calculations | : | 0.765964 s |
| Total time spent | : | 83.036128 s |

Table 5 **OUT OF CORE STATISTICS**

| ———— Out of core time (in percent) ———— | | |
|---|---|---|
| Factorization step(100): | | |
| write to files | : | 0 |
| read from files | : | 0 |
| factorization - write and read | : | 100 |
| Solution step (100): | | |
| read from files | : | 81 |
| solve - writeread | : | 19 |
| Total time (100): | | |
| read from files | : | 0 |
| total - write and read | : | 100 |
| ———— Out of core Mb ———— | | |
| Factorization step: | | |
| write to files | : | 0.000 Mb |
| read from files | : | 0.000 Mb |
| Solution step: | | |
| read from files | : | 414.116 Mb |
| Total size of data transferred: | | |
| write and read | : | 414.116 Mb |

同时 OOC 模式解决了第四个算例的内存溢出的问题，通过将大部分的存储内容放在硬盘中，用时间换空间，具备了解决大型问题的能力。例如：原本第四个算例我们的计算会发生内存溢出的问题，无法求解，现在通过 OOC 模式，将矩阵存储在硬盘中，完成了第四个算例的求解，可能是全班唯一具备求解第四个算例的能力的小组，但很遗憾的是，我们现在第四个算例的求解时间大约在 30 分钟左右，超出了求解时间，这可以在未来通过进一步的优化来解决，但 OOC 模式使得求解超大型的问题具备了理论可能，当然对于矩阵的结构等可能也需要进一步的优化，以减轻对存储容量的需求。

另外在求解过程中，我们也通过任务管理器对程序运行所占据的内存和 CPU 使用率进行了监控，在实验室的测试平台中，第四个算例中，程序 8 线程运行，CPU 基本一直接近满负荷运行，内存最高峰达到 50GB，磁盘的读写速度最高达到了 500MB/s 级别的速度，在本地测试的第二个算例中，内存高峰达到了 400MB，在 IC 模式下为 700MB，但程序运行时磁盘的读写速度大幅度上升，达到了 50MB/s，整体运行时间增长了 4 倍左右。

如果运行环境的磁盘的读写速度有较大幅度的提升，如更换为 SSD 固态硬盘，可能对程序整体的运行时间有较好的增益，同时可以进一步分配更大的可使用内存量，当前考虑到系统的安全，我们只分配了大约 45G 的可使用内存空间。这在未来都是可以考虑优化的方面。