

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 板单元 (Plate Element)

0.1.1 板单元的基本公式

我们选用的板单元为 4 节点 Kirchhoff 板单元，如图1所示：

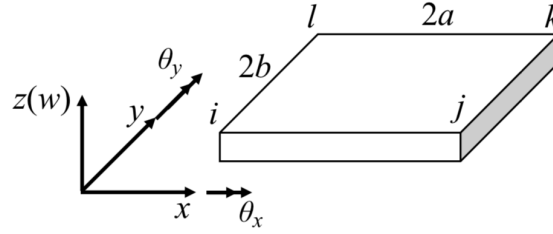


Figure 1 Kirchhoff 板单元结构示意图

其中每一个节点具有一个垂直于板的位移自由度 w 和绕板平面内两个垂直轴，即 x 轴和 y 轴的转角 θ_x, θ_y ，同时 Kirchhoff 板遵循平截面和直法线假设，因此板内部的位移场为：

$$\begin{aligned} u &= u_0 + z\theta_y \\ v &= v_0 - z\theta_x \\ w &= w_0(x, y) \end{aligned} \quad (1)$$

其中转角和挠度的关系为：

$$\theta_x = \frac{\partial w}{\partial y}, \quad \theta_y = -\frac{\partial w}{\partial x} \quad \gamma_{xz} = \gamma_{yz} = 0 \quad (2)$$

代入到应变和转角的关系中可以得到：

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = z \begin{Bmatrix} -\frac{\partial^2 w}{\partial x^2} \\ -\frac{\partial^2 w}{\partial y^2} \\ -2\frac{\partial^2 w}{\partial x \partial y} \end{Bmatrix} \quad (3)$$

可见在此假设下板单元的应力场只有三个分量

0.1.2 板单元的应力应变以及刚度阵构造

由公式 (3) 再代入应力应变关系中可以得到：

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} \quad (4)$$

由于三个应力分量和坐标 z 呈现线性关系，因此进行积分得到相应的力矩：

$$\begin{aligned}
 M &= \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \mathbf{D}\boldsymbol{\kappa} = \mathbf{D}\mathbf{L}w \\
 M_x &= \int_{-t/2}^{t/2} z\sigma_x \mathrm{d}z \\
 M_y &= \int_{-t/2}^{t/2} z\sigma_y \mathrm{d}z \\
 M_{xy} &= \int_{-t/2}^{t/2} z\tau_{xy} \mathrm{d}z
 \end{aligned} \tag{5}$$

而最终的方程：

$$\mathbf{L}^T \mathbf{D} \mathbf{L} w - q = 0 \tag{6}$$

最后构造 w 的差值函数（由于每一个板单元有 12 个自由度，因此构造一个具有 12 项的三次完备多项式）：

$$w^e = \mathbf{P}\boldsymbol{\alpha}^e \tag{7}$$

然后利用节点的位移以及转角得到：

$$\mathbf{d}^e = \mathbf{M}\boldsymbol{\alpha}^e \tag{8}$$

代入公式 (7) 可以得到：

$$\begin{aligned}
 w^e &= \mathbf{N}\mathbf{d}^e \\
 \mathbf{N} &= \mathbf{P}\mathbf{M}^{-1}
 \end{aligned} \tag{9}$$

进行简单的坐标变换可以得到矩形板单元的形函数表达式：

$$\mathbf{N}^T = \frac{1}{8} \begin{bmatrix} -(s-1)(t-1)(s^2+s+t^2+t-2) \\ -b(s-1)(t-1)^2(t+1) \\ a(s-1)^2(s+1)(t-1) \\ (s+1)(t-1)(s^2-s+t^2+t-2) \\ b(s+1)(t-1)^2(t+1) \\ a(s+1)^2(s-1)(t-1) \\ -(s+1)(t+1)(s^2-s+t^2-t-2) \\ b(s+1)(t+1)^2(t-1) \\ -a(s+1)^2(s-1)(t+1) \\ (s-1)(t+1)(s^2+s+t^2-t-2) \\ -b(s-1)(t+1)^2(t-1) \\ -a(s-1)^2(s+1)(t+1) \end{bmatrix} \quad (10)$$

对形函数 \mathbf{N} 进行求导得到：

$$\mathbf{B} = \mathbf{L}\mathbf{N} \quad (11)$$

相应的力矩 $\mathbf{M} = \mathbf{D}\mathbf{B}\mathbf{d}^e$ ，从而我们得到单元刚度阵（单元的长，宽分别为 $2a, 2b$ ）：

$$\mathbf{K} = ab \int_{-1}^1 \int_{-1}^1 \mathbf{B}^T \mathbf{D} \mathbf{B} ds dt \quad (12)$$

将公式 (10) 和公式 (11) 代入 (12) 中，并用 **matlab** 进行符号运算得到刚度阵的元素，然后利用 skyline 的存储方式输入到 **Stap++** 的 **Plate** 类的单元 **Stiffmatrix** 函数中

0.1.3 应力计算以及后处理

将单元刚度阵组装之后计算得到整体刚度阵中板单元的贡献；于此同时，对受力进行等效到节点上：

$$\mathbf{f}_i = \begin{Bmatrix} f_{w_i} \\ f_{\theta_{i_i}} \\ f_{\theta_{i_i}} \end{Bmatrix} = \int_{-b}^b \int_{-a}^a \mathbf{N}^T \mathbf{q} dx dy \quad (13)$$

对于重力的作用： $q = \rho g t$ (t 为板单元的厚度， ρ 为材料密度， g 为重力加速度)，代入公式 (13) 可以得到：

$$\begin{aligned} f_{1z} &= -\frac{1}{12}\rho g t a b \begin{Bmatrix} 3 \\ b \\ -a \end{Bmatrix} & f_2 &= -\frac{1}{12}\rho g t a b \begin{Bmatrix} 3 \\ -b \\ -a \end{Bmatrix} \\ f_3 &= -\frac{1}{12}\rho g t a b \begin{Bmatrix} 3 \\ b \\ a \end{Bmatrix} & f_4 &= -\frac{1}{12}\rho g t a b \begin{Bmatrix} 3 \\ -b \\ a \end{Bmatrix} \end{aligned} \quad (14)$$

同样将其组装到整体的方程中，可以求解相应的节点位移和转角。

利用求解得到的位移和转角代入公式 (9) 中可以得到相应的挠度分布, 仅以不由转角-挠度关系 (2) 可以得到单元内部的转角分布

关于应力，由 (3) 可知：应力和应变在中性面的值为 0，在表面最大，因此我们可以选择输出表面的应力 ($z = \pm \frac{t}{2}$) 或者 z 方向高斯点 ($z = \pm \frac{t}{2\sqrt{3}}$) 的应力：

$$\begin{aligned} \text{表面} \quad \sigma_{(\pm)}^e &= \pm \frac{6}{t^2} M^e \\ \text{Gauss点} \quad \sigma_{(\pm)}^e &= \pm \frac{6}{\sqrt{3}t^2} M^e \end{aligned} \quad (15)$$

最后针对后处理的输出，可以输出相应上下表面对应 xy 平面的 Gauss 点的 Mises 应力，然后取其平均值作为该单元的 Mises 应力。

0.1.4 输入文件格式和自由度约束

主体的输入文件格式基本不变，但是对于板单元而言，其节点只有三个自由度，也就是说其另外三个自由度是锁死的，那么对于倾斜的板单元，其对应的约束就较为复杂，需要利用旋转刚度阵将其进行坐标变换，由于最终的桥梁算例中，板单元都是位于 xy 平面内的，因此在 Stap++ 中并没有实现倾斜板的功能；在此情况下输入文件中 6 个 bcode 数中，前两个 (bcode[0],bcode[1]) 和最后一个 (bcode[5]) 都应该设为 1, 而其余几个 bcode 数则根据具体的模型约束情况而定，我们以一个板单元的输入文件为例子，如图2所示：

1	Cables to test STAP90									
2	4	1	1	1						
3	1	1	1	1	1	1	1	0.0	0.0	0.0
4	2	1	1	0	0	0	1	2.0	0.0	0.0
5	3	1	1	1	1	1	1	0.0	2.0	0.0
6	4	1	1	0	0	0	1	2.0	2.0	0.0
7	1	0								
8	7	1	1							
9	1	3E7	0.3	1	1E3					
10	1	1	2	4	3	1				

Figure 2 1 个单元的板单元输入文件示意图

而后续的节点力则按照公式 (14) 添加，格式和其他单元一致。

0.1.5 分片测试

我们在 $[-1, 1]^2$ 的正方形板构造这样的位移场 $w = x^2 - \nu y^2$, 由公式 (2) 可以得到:

$$\begin{aligned}\theta_x &= -2\nu y \\ \theta_y &= -2x\end{aligned}\tag{16}$$

由公式 (5) 可以得到均匀弯矩场:

$$\mathbf{M}^e = \begin{bmatrix} -\frac{1}{6}Et^3 \\ 0 \\ 0 \end{bmatrix}\tag{17}$$

其中 $E = 30, \nu = 0.2, t = 1$; 利用 $x = 0; y = 0; y = 0.5$ 三条直线将正方形板分割成六个分片, 如图3所示:

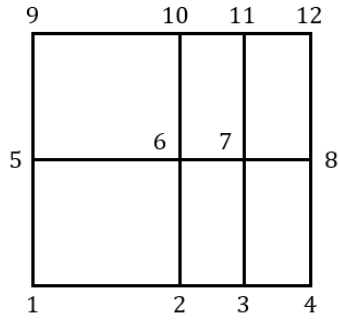


Figure 3 Plate 单元分片实验划分示意图

对应的将法线沿 x 轴方向边界 ($x = \pm 1$) 上的 y 向弯矩分配到边界上的每个点上:

$$\begin{aligned}M_{4y} &= -M_{1y} = -2.5 \\ M_{8y} &= -M_{5y} = -5 \\ M_{12y} &= -M_{9y} = -2.5\end{aligned}\tag{18}$$

具体输入文件见 Platepatchtest.dat, 在程序中的位移输出结果如图4所示:

D I S P L A C E M E N T S						
NODE	X-DISPLACEMENT	Y-DISPLACEMENT	Z-DISPLACEMENT	X-ROTATION	Y-ROTATION	Z-ROTATION
1	0.00000e+00	0.00000e+00	8.00000e-01	4.00000e-01	2.00000e+00	0.00000e+00
2	0.00000e+00	0.00000e+00	-2.00000e-01	4.00000e-01	4.10783e-15	0.00000e+00
3	0.00000e+00	0.00000e+00	5.00000e-02	4.00000e-01	-1.00000e+00	0.00000e+00
4	0.00000e+00	0.00000e+00	8.00000e-01	4.00000e-01	-2.00000e+00	0.00000e+00
5	0.00000e+00	0.00000e+00	1.00000e+00	-3.33067e-16	2.00000e+00	0.00000e+00
6	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
7	0.00000e+00	0.00000e+00	2.50000e-01	-2.67364e-16	-1.00000e+00	0.00000e+00
8	0.00000e+00	0.00000e+00	1.00000e+00	0.00000e+00	-2.00000e+00	0.00000e+00
9	0.00000e+00	0.00000e+00	8.00000e-01	-4.00000e-01	2.00000e+00	0.00000e+00
10	0.00000e+00	0.00000e+00	-2.00000e-01	-4.00000e-01	4.19977e-15	0.00000e+00
11	0.00000e+00	0.00000e+00	5.00000e-02	-4.00000e-01	-1.00000e+00	0.00000e+00
12	0.00000e+00	0.00000e+00	8.00000e-01	-4.00000e-01	-2.00000e+00	0.00000e+00

Figure 4 Plate 单元分片实验位移结果

, 可见位移 w 准确, 误差在浮点数范围内, 以 7 号节点为例:

$$\begin{aligned}
 w_7 &= x_7^2 - \nu y_7^2 = 0.25 \\
 \theta_{x7} &= -2\nu y_7 = 0 \\
 \theta_{y7} &= -2x_7 = -1
 \end{aligned} \tag{19}$$

从 θ_x 上的误差上来看, 误差基本上在 10^{-16} 量级上, 为计算机浮点数的误差

0.1.6 收敛率计算

考虑一个正方形板, $W = L = 2, t = 1$, 其中 $E = 3 \times 10^7, \nu = 0.3, t = 1$ 如图5所示:

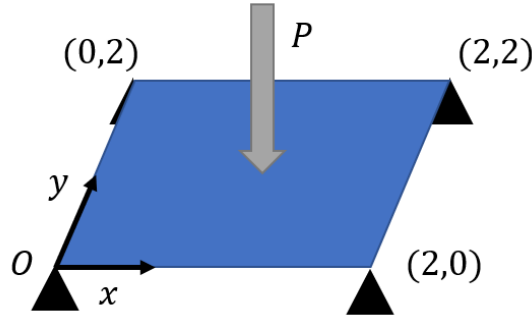


Figure 5 Plate 收敛率计算示意图

在中心点施加一个集中力 ($-z$ 方向) $P = 10^5$, 因此只需要在对应的节点施加这样一个力即可, 然后分别考虑划分 4, 16, 64 和 256 个等大小的矩形板单元进行计算 (对应单元尺寸 $h = 1, 0.5, 0.25, 0.125$), 相应的输入文件用 `data.m` 生成, 分别为: `Plate4ele.dat`, `Plate16ele.dat` 和 `Plate64ele.dat`, `Plate256ele.dat`, 计算相应的节点位移 w^e 与理论结果进行比较:

$$w = \frac{16P}{D\pi^4} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(-1)^{(m+n)}}{((2m+1)^2 + (2n+1)^2)^2} \sin \frac{m\pi x}{2} \sin \frac{n\pi y}{2} \tag{20}$$

计算误差范数：

$$error = (\int_0^L \int_0^W (w^e - w)^2 dx dy)^{\frac{1}{2}} \quad (21)$$

由于 w 至少是 4 次场，所以该积分至少需要采用 16*16 的 Gauss 积分才能准确积分，得到计算结果如图6所示, 并作出误差范数和单元尺寸的对数图

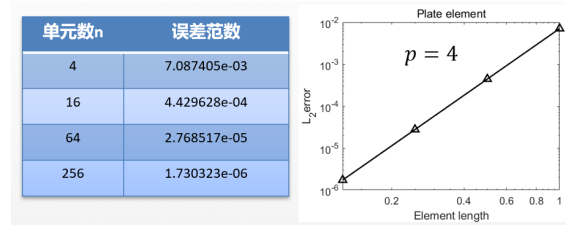


Figure 6 Plate 收敛率计算结果

可见板单元的挠度 w 是 4 阶收敛的，因为其构造挠度的多项式是 3 次完备的。

0.2 9 节点平面亚参元 (Subparameter Element)

0.2.1 9 节点平面亚参元基本公式

我们这里采用的 9 节点平面亚参元，其母单元示意图如图7所示：

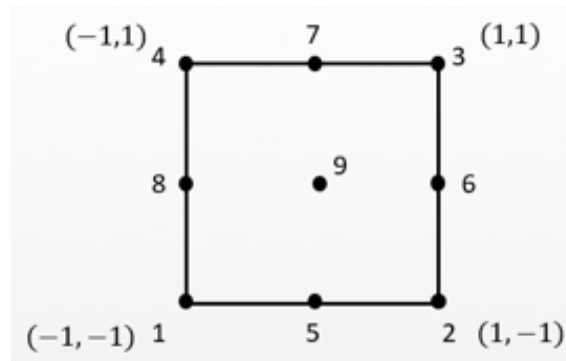


Figure 7 9 节点平面亚参元母单元示意图

其坐标差值点为 1, 2, 3, 4 四个节点, 而位移差值采用 1-9 九个节点:

$$\begin{aligned} x &= \sum_{I=1}^4 N'_I x_I \\ y &= \sum_{I=1}^4 N'_I y_I \\ \theta &= \sum_{K=1}^9 N_K \theta_K \end{aligned} \quad (22)$$

其中 $N'_I = \frac{1}{4}(1 + \xi_I \xi)(1 + \eta_I \eta)$, 而:

$$\begin{aligned} N_1 &= \frac{1}{4}\xi\eta(\xi-1)(\eta-1) \\ N_2 &= \frac{1}{4}\xi\eta(\xi+1)(\eta-1) \\ N_3 &= \frac{1}{4}\xi\eta(\xi+1)(\eta+1) \\ N_4 &= \frac{1}{4}\xi\eta(\xi-1)(\eta+1) \\ N_5 &= \frac{1}{2}\eta(\eta-1)(1-\xi^2) \\ N_6 &= \frac{1}{2}\xi(\xi+1)(1-\eta^2) \\ N_7 &= \frac{1}{2}\eta(\eta+1)(1-\xi^2) \\ N_8 &= \frac{1}{2}\xi(\xi-1)(1-\eta^2) \\ N_9 &= (1-\xi^2)(1-\eta^2) \end{aligned} \quad (23)$$

0.2.2 9 节点平面亚参元的应力应变以及刚度阵构造

其余过程和 4Q 单元一致, 首先应变由位移差值 (22) 可以得到:

$$\begin{aligned} \epsilon^e &= \sum_{i=1}^9 \mathbf{B}_i \mathbf{d}_i \\ \mathbf{d}_i &= \begin{bmatrix} u_i \\ v_i \end{bmatrix} \\ \mathbf{B}_i &= \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} \end{bmatrix} \end{aligned} \quad (24)$$

利用雅可比矩阵得到物理空间坐标和母单元坐标的转换关系 (但这里的坐标利用的是 4 节点差值), 进而得到单元刚度阵 (18×18):

$$\mathbf{K}^e = \int_{-1}^1 \int_{-1}^1 \mathbf{B}^T \mathbf{D} \mathbf{B} \det|\mathbf{J}| d\xi d\eta \quad (25)$$

采用 Skyline 的方式深书写到程序中, 并且采用 3×3 的 Gauss 积分计算每一个元素。

而相应的节点等效力由公式:

$$\mathbf{f}^e = \int_{\Gamma_t^e} \mathbf{N}^{eT} \bar{\mathbf{t}} d\Gamma + \int_{\Omega^e} \mathbf{N}^{eT} \mathbf{b} d\Omega \quad (26)$$

计算

0.2.3 输入文件格式和自由度约束

平面 9 节点亚参元的输入文件和 4Q 单元类似，其后三个转动自由度可以在 Stapp++ 中默认设置为 1，然后可以不需要在输入文件中体现，而对于一般的平面问题，我们可以不锁死 z 方向的自由度，但是为了求解简单，一般我们可以把坐标系的 xy 设置为单元所在的面，这样我们可以把 z 方向的自由度 (bcode[2]) 设置为 1，这样对应的平面 9 节点亚参元的总自由度为 18。

关于节点信息的输入，严格来说，我们可以用 1, 2, 3, 4 号节点自动生成节点 5-9 的坐标，但是在 Nodelist 中需要出现 5-9 对应的节点号，由于在 Stapp++ 中实现自动添加节点的功能还没有扩展（这也是可以继续扩展的方向），因此我们还是需要在输入文件中输入 5-9 号节点，只是相应的坐标全部可以写为 0，然后在 Stapp++ 中会计算其相应的坐标赋值给 XYZ

材料和受力和一般的单元输入格式一致，而单元对应的节点号部分应该按照图7对应的节点编号顺序填，图8展示了一个 9 节点亚参元的输入文件格式：

```

1 cables to test STAP90
2 9      1      1      1
3 1      1      0      1      0.0      0.0      0.0
4 2      0      0      1      2.0      0.0      0.0
5 3      1      0      1      0.0      2.0      0.0
6 4      0      0      1      2.0      2.0      0.0
7 5      0      0      1      0.0      0.0      0.0
8 6      0      0      1      0.0      0.0      0.0
9 7      0      0      1      0.0      0.0      0.0
10 8     1      1      1      0.0      0.0      0.0
11 9     0      0      1      0.0      0.0      0.0
12 1     9
13 1     1      0
14 2     1      2
15 3     1      0
16 4     1      2
17 5     1      4
18 6     1      8
19 7     1      4
20 8     1      0
21 9     1     16
22 9     1      1
23 1     3E7    0.3
24 1     1      2      4      3      5      6      7      8      9      1
25
26

```

Figure 8 1 个 9 节点亚参元的输入文件格式

0.2.4 分片测试

先考虑考虑一个平板的一端 (沿着 xy 平面), 长宽: $W = L = 2$, 其材料参数: $E = 3 \times 10^7$, $\nu = 0.3$, 其受到 x 方向的一个均匀拉力 $\sigma = 9$ 的情况，如图9所示：

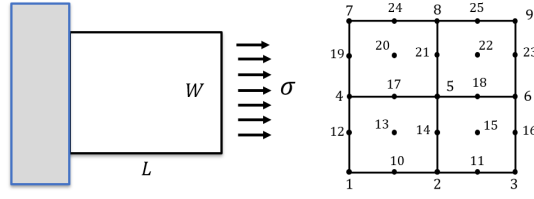


Figure 9 9节点平面亚参元的分片测试1示意图

并且我们划分4个等大的单元，节点划分如图9所示，其中平板的一段被约束住（即节点1，7，12，19被约束住 x 方向自由度，而4号节点约束住 x, y 方向自由度）；利用节点力等效公式 (26) 我们可以得到在该情况下 (对于矩形单元) 的节点等效力为： $f_{ix}|_{\Gamma} = \frac{\sigma h}{2} \int_{-1}^1 N_i(\xi = 1) d\eta$ （其中 h 为单元的长度）

$$\begin{aligned} f_{2x} &= f_{4x} = \frac{1}{6}\sigma h \\ f_{6x} &= \frac{2}{3}\sigma h \end{aligned} \quad (27)$$

注意这里的1-9对应的是母单元的编号，具体的编号要根据 Location matrix 对应的世纪编号来；相应的具体输入文件利用 matlab 程序 Subpara_force.m 生成，具体输入文件请见 Subparapatchtest.dat，程序计算的求解结果如图10所示：

D I S P L A C E M E N T S		
NODE	X-DISPLACEMENT	Y-DISPLACEMENT
1	0.00000e+00	9.00000e-08
2	3.00000e-07	9.00000e-08
3	6.00000e-07	9.00000e-08
4	0.00000e+00	0.00000e+00
5	3.00000e-07	3.11847e-22
6	6.00000e-07	2.05826e-22
7	0.00000e+00	-9.00000e-08
8	3.00000e-07	-9.00000e-08
9	6.00000e-07	-9.00000e-08
10	1.50000e-07	9.00000e-08
11	4.50000e-07	9.00000e-08
12	0.00000e+00	4.50000e-08
13	1.50000e-07	4.50000e-08
14	3.00000e-07	4.50000e-08
15	4.50000e-07	4.50000e-08
16	6.00000e-07	4.50000e-08
17	1.50000e-07	2.74624e-22
18	4.50000e-07	2.90297e-22
19	0.00000e+00	-4.50000e-08
20	1.50000e-07	-4.50000e-08
21	3.00000e-07	-4.50000e-08
22	4.50000e-07	-4.50000e-08
23	6.00000e-07	-4.50000e-08
24	1.50000e-07	-9.00000e-08
25	4.50000e-07	-9.00000e-08

Figure 10 9节点平面亚参元的分片测试1结果

其与理论的结果（单轴拉伸）：

$$\begin{aligned} u &= \frac{\sigma}{E}x \\ v &= -\frac{\nu\sigma}{E}(y-1) \end{aligned} \quad (28)$$

相比较，从 5 号节点的 y 方向位移，可以看出其误差基本在计算机浮点数范围 (10^{-22}) 内，因此通过分片测试 1；

进一步的提高分片测试的阶次，给平板一个沿着 x 方向均匀拉伸的体积力 $b_x = a = 9$, 其余参数不变，如图11所示：

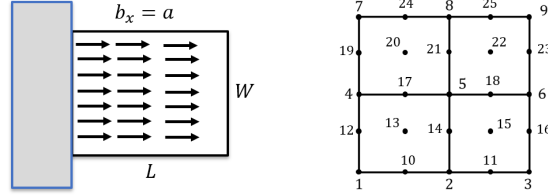


Figure 11 9 节点平面亚参元的分片测试 1 示意图

同样利用节点力等效公式 (26)，我们可以得到 (对于矩形单元) 每个单元等效的节点力为： $f_{ix} = \frac{ah^2}{4} \int_{-1}^1 \int_{-1}^1 N_i d\xi d\eta$ (其中 h 为单元的长度)

$$\begin{aligned} f_{1x} &= f_{2x} = f_{3x} = f_{4x} = \frac{ah^2}{36} \\ f_{5x} &= f_{6x} = f_{7x} = f_{8x} = \frac{ah^2}{9} \\ f_{9x} &= \frac{4ah^2}{9} \end{aligned} \quad (29)$$

其余的输入文件文件部分和 Subparapatchtest.dat 相同，见 Subparapatchtest2.dat 文件；程序的计算求解结果如图??所示：

D I S P L A C E M E N T S		
NODE	X-DISPLACEMENT	Y-DISPLACEMENT
1	0.00000e+00	1.80000e-07
2	4.50000e-07	9.00000e-08
3	6.00000e-07	7.70041e-22
4	0.00000e+00	0.00000e+00
5	4.50000e-07	9.18998e-22
6	6.00000e-07	1.44994e-21
7	0.00000e+00	-1.80000e-07
8	4.50000e-07	-9.00000e-08
9	6.00000e-07	-7.70041e-22
10	2.62500e-07	1.35000e-07
11	5.62500e-07	4.50000e-08
12	0.00000e+00	9.00000e-08
13	2.62500e-07	6.75000e-08
14	4.53326e-07	4.50000e-08
15	5.62500e-07	2.25000e-08
16	6.00000e-07	6.02671e-22
17	2.62500e-07	5.92261e-22
18	5.62500e-07	1.22241e-21
19	0.00000e+00	-9.00000e-08
20	2.62500e-07	-6.75000e-08
21	4.53326e-07	-4.50000e-08
22	5.62500e-07	-2.25000e-08
23	6.00000e-07	-6.02671e-22
24	2.62500e-07	-1.35000e-07
25	5.62500e-07	-4.50000e-08

Figure 12 9 节点平面亚参元的分片测试 2 结果

其与理论结果：

$$\begin{aligned} u &= \frac{a}{E}(2x - \frac{x^2}{2}) \\ v &= -\frac{a\nu}{E}(2 - x)(y - 1) \end{aligned} \quad (30)$$

比较的误差在计算浮点数范围 (10^{-22}), 因此其能通过分片测试 2, 所以其收敛率至少为 3。

0.2.5 收敛率计算

将分片测试 2 中的均匀体积力改为一个沿 x 方向线性分布体力的情况: $b_x = ax(a = 9)$, 其余几何参数, 材料参数以及单元划分不变, 如图13所示:

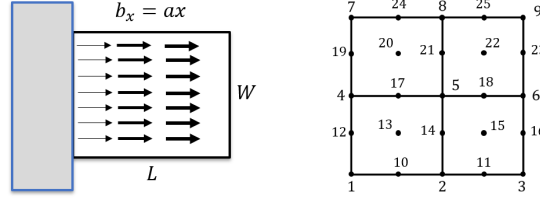


Figure 13 9 节点平面亚单元收敛率计算示意图

同样利用节点力等效公式 (26), 我们可以得到 (对于矩形单元) 每个单元等效的节点力为: $f_{ix} = \frac{ah^2}{4} \int_{-1}^1 \int_{-1}^1 N_i(\frac{x_1+x_2}{2} + \frac{h}{2}\xi) d\xi d\eta$ (其中 x_1 和 x_2 为母单元中 1, 2 号节点对应物理空间的坐标, h 为单元长度)

$$\begin{aligned} f_{1x} &= f_{4x} = \frac{ah^2}{36}(\frac{x_1+x_2}{2} - \frac{h}{2}) \\ f_{2x} &= f_{3x} = \frac{ah^2}{36}(\frac{x_1+x_2}{2} + \frac{h}{2}) \\ f_{5x} &= f_{7x} = \frac{ah^2}{9}\frac{x_1+x_2}{2} \\ f_{6x} &= \frac{ah^2}{9}(\frac{x_1+x_2}{2} + \frac{h}{2}) \\ f_{8x} &= \frac{ah^2}{9}(\frac{x_1+x_2}{2} - \frac{h}{2}) \\ f_{9x} &= \frac{4ah^2}{9}\frac{x_1+x_2}{2} \end{aligned} \quad (31)$$

注意这里的 1-9 对应的是母单元的编号, 具体的编号要根据 Location matrix 对应的世纪编号来; 相应的具体输入文件利用 matlab 程序 Subpara_force.m 生成; 我们分别计算划分 1, 4, 16 和 64 个单元时的结果 (对应单元尺寸 $h = 2, 1, 0.5, 0.25$), 具体输入文件请见 Subpara1ele.dat, Subpara4ele.dat, Subpara16ele.dat 和 Subpara64ele.dat; 计算相应的节点位移 w^e 与理论结果进行比较:

$$\begin{aligned} u &= \frac{3}{2E}(12x - x^3) \\ v &= \frac{9\nu}{2E}(x^2 - 4)(y - 1) \end{aligned} \quad (32)$$

我们这里计算 x 方向位移的误差范数：

$$error = (\int_0^L \int_0^W (u^e - u)^2 dx dy)^{\frac{1}{2}} \quad (33)$$

由于 u 是三次场，所以至少要用 9×9 节点 Gauss 积分，得到的计算结果如图14所示：

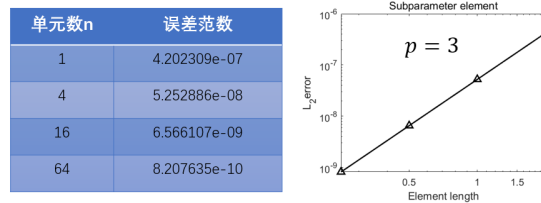


Figure 14 9 节点平面亚参元收敛率计算结果

可见其位移的收敛率为 3，和分片实验的结果一致，因为其位移差值能够精确重构 2 次多项式。

0.3 无限单元 (Infinite element)

0.3.1 无限单元的基本公式

我们这里采用的是 4 节点平面等参无限单元（之后简称为无限单元）；其主要适用于无限大问题的求解，其核心的思想是利用两个参考点 C 和 C_1 把边界点 P 和 P_1 还有对称的参考点 Q 和 Q_1 把无穷原点映射到母单元中 $\xi = 1$ 的两个点；最终在母单元中 P, Q, R 和 P_1, Q_1, R_1 对应平行与 ξ 轴的两条边，因此我们只需要构造一种一维差值 (ξ 方向) 能够满足把无穷远 R 点 ($x \rightarrow \infty$) 映射到 $\xi = 1$ 即可，如图15所示：

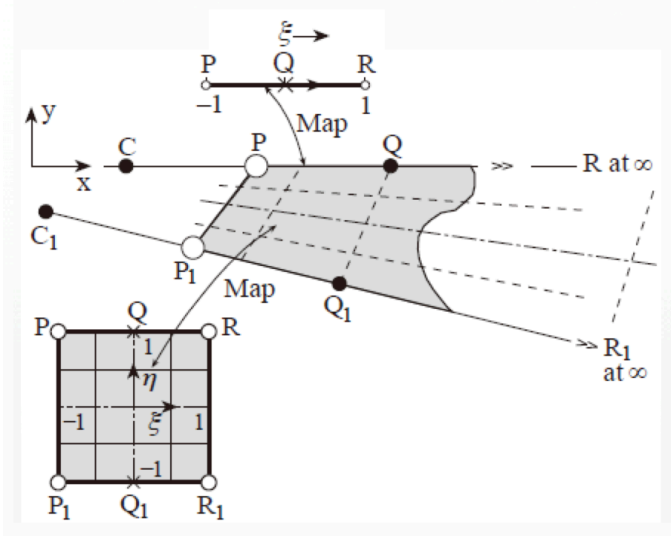


Figure 15 无限单元映射关系示意图

首先对于 ξ 方向的差值:

$$\begin{aligned}
 x &= \bar{N}_C x_C + \bar{N}_Q x_Q \\
 \bar{N}_C &= -\frac{\xi}{1-\xi} \\
 \bar{N}_Q &= 1 + \frac{\xi}{1-\xi}
 \end{aligned} \tag{34}$$

其可以使得 P 点从物理空间中的 $x = \frac{x_C + x_Q}{2}$ 映射到 $\xi = -1$, 使得 Q 点从物理空间中的 $x = x_Q$ 映射到 $\xi = 0$, 使得 R 点从物理空间中的 $x \rightarrow \infty$ 映射到 $\xi = 1$, 而 η 方向的差值可直接采用一般的两节点线性差值: $N_1 = \frac{1}{2}\eta(1 + \eta)$, $N_0 = \frac{1}{2}\eta(1 - \eta)$, 于是整个 4 节点无限单元的形函数和差值关系为:

$$\begin{aligned}
 x &= \sum_{i=1}^4 N_i^* x_i \\
 N_1^* &= N_1(\eta) \bar{N}_C \\
 N_2^* &= N_1(\eta) \bar{N}_Q \\
 N_3^* &= N_0(\eta) \bar{N}_C \\
 N_4^* &= N_0(\eta) \bar{N}_Q
 \end{aligned} \tag{35}$$

其中 $i = 1$ 对应 C, $i = 2$ 对应 Q, $i = 3$ 对应 C_1 , $i = 4$ 对应 Q_1 。

0.3.2 无限单元的应力应变以及刚度阵构造

其余有关应力和刚度阵的后续推导和平面 4 节点等参元一致，但是由于形函数不同 (分式多项式)，会导致形函数的导数会有变化：

$$\begin{aligned} \frac{\partial N_1^*}{\partial \xi} &= -\frac{1}{2} \frac{(1+\eta)}{(1-\xi)^2} & \frac{\partial N_1^*}{\partial \eta} &= -\frac{1}{2} \frac{\xi}{1-\xi} \\ \frac{\partial N_2^*}{\partial \xi} &= \frac{1}{2} \frac{(1+\eta)}{(1-\xi)^2} & \frac{\partial N_2^*}{\partial \eta} &= \frac{1}{2} \left(1 + \frac{\xi}{1-\xi} \right) \\ \frac{\partial N_3^*}{\partial \xi} &= -\frac{1}{2} \frac{(1-\eta)}{(1-\xi)^2} & \frac{\partial N_3^*}{\partial \eta} &= \frac{1}{2} \frac{\xi}{1-\xi} \\ \frac{\partial N_4^*}{\partial \xi} &= \frac{1}{2} \frac{(1-\eta)}{(1-\xi)^2} & \frac{\partial N_4^*}{\partial \eta} &= -\frac{1}{2} \left(1 + \frac{\xi}{1-\xi} \right) \end{aligned} \quad (36)$$

从而代入到 \mathbf{B} 矩阵的计算中：

$$[\mathbf{B}] = \begin{bmatrix} \frac{\partial N_i^*}{\partial x} & 0 \\ 0 & \frac{\partial N_i^*}{\partial y} \\ \frac{\partial N_i^*}{\partial y} & \frac{\partial N_i^*}{\partial x} \end{bmatrix} = [\mathbf{J}]^{-1} [\mathbf{GN}^*] \quad [\mathbf{J}] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (37)$$

进一步由公式 (25) 计算刚度阵列，程序中采用 2×2 的 Gauss 积分（但是由于这里雅可比矩阵中存在分式多项式，因此 Gauss 积分不一定能够做到准确积分）计算并且用 Skyline 的方式存储。

而其余的节点力等效和 Gauss 点应力计算和 4Q 单元一致，程序实现上的思路也一致，只是需要把形函数改为 (35)

0.3.3 无限单元的收敛性分析

首先无限单元作为物理尺寸无限大的单元无法像常规的单元一样用单元的尺来考虑收敛率，但是从等参元的基本性质出发，无限单元是满足等参元的相容性和连续性：（因为其对应的节点 C,Q 不是有限单元边界上的点 P），于此同时其同样满足刚体位移条件：

$$\sum_{i=1}^4 N_i^* = 1 \quad (38)$$

因此该单元是收敛的。但是需要注意，由于无限单元节点的特殊性，这使得输入文件的构造变得很困难，首先需要给定有限单元的边界，然后去寻找关于边界的对称点，这对于程序上实现会有一些难度。

0.4 Euler-Bernoulli Beam 单元

0.4.1 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：

$$d^e = \left(u_{x1} \quad u_{y1} \quad u_{z1} \quad u_{x1} \quad u_{y1} \quad u_{z1} \quad u_{x2} \quad u_{y2} \quad u_{z2} \quad \theta_{x2} \quad \theta_{y2} \quad \theta_{y3} \right)^T \quad (39)$$

$$f^e = \left(F_{x1} \quad F_{y1} \quad F_{z1} \quad M_{x1} \quad M_{y1} \quad M_{z1} \quad F_{x2} \quad F_{y2} \quad F_{z2} \quad M_{x2} \quad M_{y3} \right)^T \quad (40)$$

其中 u 表示位移， θ 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。

$$K^e = \begin{pmatrix} \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{l} & 0 & 0 & 0 & 0 & 0 \\ & \frac{12EI_z}{l^3} & 0 & 0 & 0 & \frac{6EI_z}{l^2} & 0 & -\frac{12EI_z}{l^3} & 0 & 0 & 0 & \frac{6EI_z}{l^2} \\ & & \frac{12EI_y}{l^3} & 0 & -\frac{6EI_z}{l^2} & 0 & 0 & 0 & -\frac{12EI_y}{l^3} & 0 & -\frac{6EI_z}{l^2} & 0 \\ & & & \frac{GJ_z}{I} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ_z}{I} & 0 & 0 \\ & & & & \frac{4EI_y}{l} & 0 & 0 & 0 & \frac{6EI_y}{l} & 0 & \frac{2EI_y}{l} & 0 \\ & & & & & \frac{4EI_z}{l} & 0 & -\frac{6EI_z}{l} & 0 & 0 & 0 & \frac{2EI_z}{l} \\ & & \text{Symmetry} & & & & \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & \frac{12EI_z}{l^3} & 0 & 0 & 0 & -\frac{6EI_z}{l^2} \\ & & & & & & & & \frac{12EI_y}{l^3} & 0 & \frac{6EI_y}{l^2} & 0 \\ & & & & & & & & & \frac{GJ_z}{I} & 0 & 0 \\ & & & & & & & & & & \frac{4EI_y}{l} & 0 \\ & & & & & & & & & & & \frac{4EI_z}{l} \end{pmatrix} \quad (41)$$

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：

$$K^e d^e = f^e \quad (42)$$

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：

$$K^e R d^e = R f^e \quad (43)$$

因此有

$$K^{global} = R^T K^{element} R \quad (44)$$

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

$$R = \begin{bmatrix} A & & & \\ & A & & \\ & & A & \\ & & & A \end{bmatrix} \quad (45)$$

其中 A=

$$\hat{A} = \begin{bmatrix} a_x a'_x & a_x a'_y & a_x a'_z \\ a_y a'_x & a_y a'_y & a_y a'_z \\ a_z a'_x & a_z a'_y & a_z a'_z \end{bmatrix} \quad (46)$$

将如此经过正交变换的单元刚度阵组装到总刚度阵即可

0.4.2 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 μ ，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)

0.4.3 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。

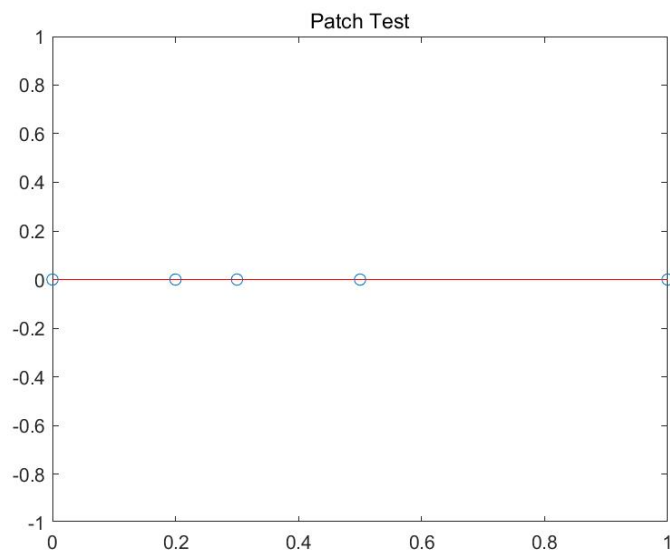


Figure 16 Patch Test

由于梁有四个互相独立解耦的模式，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：

$$w = x^2 - x, \theta = 2x - 1 \quad (47)$$

分片试验结果如下，

Table 1 DISPLACEMENT-WZ

NODE	X-D	Y-D	Z-D	θ_X -D	θ_Y -D	θ_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

$$w = x^2 - x, \theta = 2x - 1 \quad (48)$$

分片试验结果如下，

Table 2 **DISPLACEMENT-WZ**

NODE	X-D	Y-D	Z-D	θ_X -D	θ_Y -D	θ_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, θ_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	θ_X -D	θ_Y -D	θ_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

0.4.4 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。

$$w = 0.01x^4 \quad (49)$$

误差范数采用位移范数, 可以表示为

$$\|e\|^2 = \int A(u - u^h)^2 dx \quad (50)$$

计算收敛率时网格划分如下:

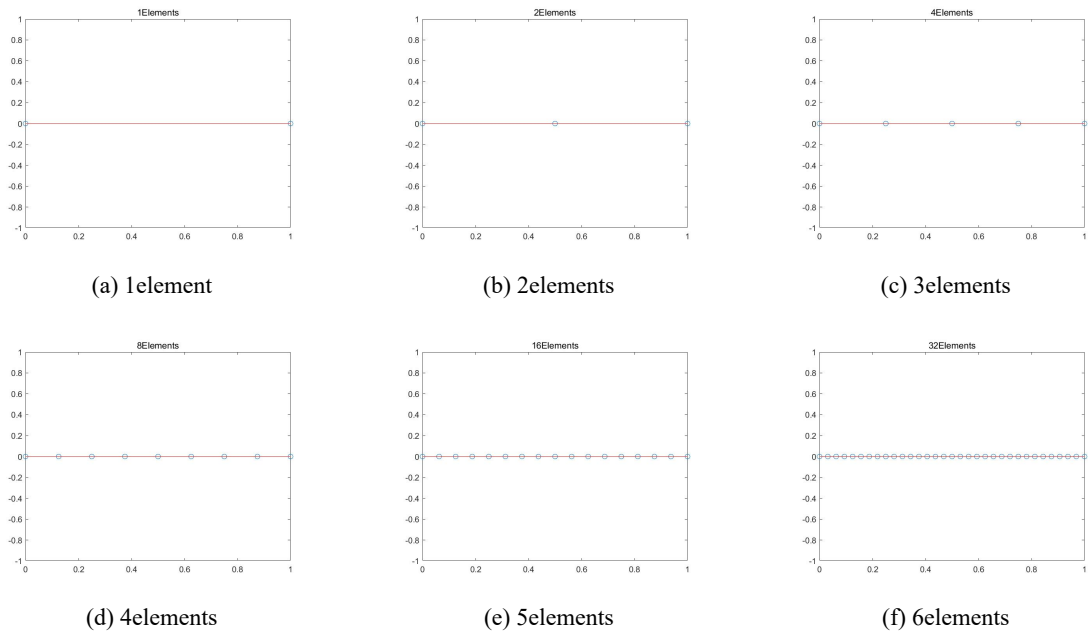


Figure 17 Convergence Mesh

最终收敛率结果如图所示:

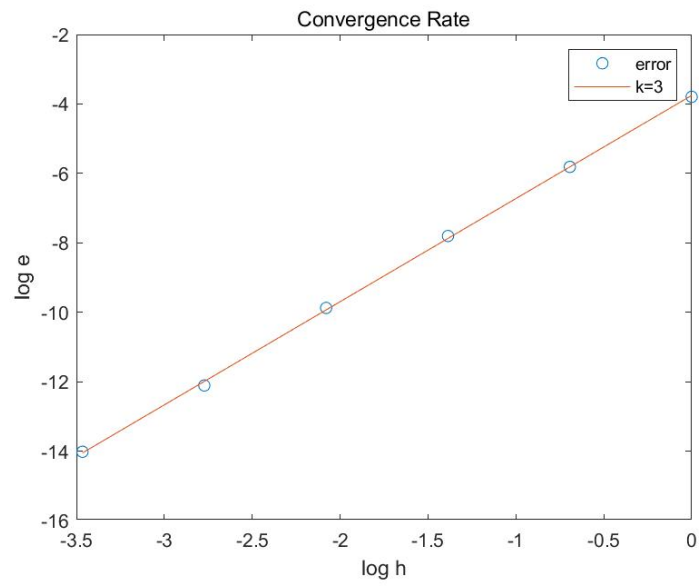


Figure 18 Convergence Rate

0.5 Timoshenko Beam

0.5.1 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。

0.5.2 Timoshenko Beam 的控制方程

从变分原理出发有，系统的势能泛函为

$$\Pi_P = \int_0^l \frac{1}{2} EI \left(\frac{d\theta}{dx} \right)^2 dx + \int_0^l \frac{1}{2} \frac{GA}{k} \gamma^2 dx - \int_0^l q w dx \quad (51)$$

其中，

$$\varepsilon_x = -z \frac{d\theta}{dx} \quad \gamma = \frac{dw}{dx} - \theta \quad \kappa = -\frac{d\theta}{dx} \quad (52)$$

按照有限元格式离散之后可以写成

$$\begin{aligned} \Pi_P &= \int_0^l \frac{1}{2} EI \left(\frac{d\theta}{dx} \right)^2 dx + \int_0^l \frac{1}{2} \frac{GA}{k} \gamma^2 dx - \int_0^l q w dx \\ &= \sum_e \left[\int_{\Omega^e} \frac{1}{2} EI \left(\frac{d\theta}{dx} \right)^2 dx + \int_{\Omega^e} \frac{1}{2} \frac{GA}{k} \gamma^2 dx - \int_{\Omega^e} q w dx \right] \\ &= \frac{1}{2} \sum_e (\mathbf{d}^e)^T [(\mathbf{K}_b^e + \mathbf{K}_s^e) \mathbf{d}^e - \mathbf{f}^e] \end{aligned} \quad (53)$$

其中

$$\begin{aligned} \mathbf{K}_b^e &= \int_{\Omega^e} EI \mathbf{B}_b^{eT} \mathbf{B}_b^e dx \\ \mathbf{K}_s^e &= \int_{\Omega^e} \frac{GA}{k} \mathbf{B}_s^{eT} \mathbf{B}_s^e dx \\ \mathbf{f}^e &= \int_{\Omega^e} \mathbf{N}^{eT} \begin{bmatrix} q \\ 0 \end{bmatrix} dx \end{aligned} \quad (54)$$

根据最小势能原理有

$$\delta \Pi_P = 0 \quad (55)$$

从而

$$\begin{aligned} EI_z \frac{d\kappa_z}{dx} + \frac{GA}{k} \gamma_{xy} + \overline{M}_{zk} \delta(x - x_k) &= 0 \\ \frac{GA}{k} \frac{d\gamma_{xy}}{dx} + \overline{q} + \overline{N}_{yj} \delta(x - x_j) &= 0 \end{aligned} \quad (56)$$

即

$$\delta \mathbf{d}^T \left(\sum_e \mathbf{L}^{eT} (\mathbf{K}_b^e + \mathbf{K}_s^e) \mathbf{L}^e \mathbf{d} - \sum_e \mathbf{L}^{eT} \mathbf{f}^e \right) = 0 \quad \forall \mathbf{d}_F \quad (57)$$

0.5.3 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：

$$\begin{aligned}
 a &= a_b + a_s \\
 a_i &= (u_i, v_i, w_i, \varphi_i, \theta_{yi}, \theta_{zi})^T \\
 a_b &= (a_{b1}, a_{b2})^T \\
 a_s &= (a_{s1}, a_{s2})^T \\
 a_{bi} &= (u_i, v_{bi}, w_{bi}, \varphi_i, \theta_{yi}, \theta_{zi})^T \\
 a_{si} &= (0, v_{si}, w_{si}, 0, 0, 0)^T
 \end{aligned} \tag{58}$$

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：

$$\begin{aligned}
 \frac{dv_{bi}}{dx} &= \theta_{zi} \\
 \frac{dw_{bi}}{dx} &= -\theta_{yi}
 \end{aligned} \tag{59}$$

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

$$v_b = N_1 v_{b1} + N_2 \theta_1 + N_3 v_{b2} + N_4 \theta_4 \tag{60}$$

对于剪切部分，采用两点线性插值

$$v_s = N_5 v_{s1} + N_6 v_{s2} \tag{61}$$

其中 ξ 为母单元坐标值

$$\begin{aligned}
 N_1 &= 1 - 3\xi^2 + 2\xi^3 & N_2 &= (\xi - 2\xi^2 + \xi^3)l \\
 N_3 &= 3\xi^2 - 2\xi^3 & N_4 &= (\xi^3 - \xi^2)l \\
 N_5 &= 1 - \xi & N_6 &= \xi
 \end{aligned} \tag{62}$$

将插值函数代入有

$$\mathbf{B}_b^e = \begin{bmatrix} \frac{dN_1}{dx} & \dots & \frac{dN_n}{dx} \end{bmatrix} \tag{63}$$

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

$$\begin{pmatrix} v_{b2} - v_{b1} \\ v_{s2} - v_{s1} \end{pmatrix} = \begin{pmatrix} \frac{1}{1+b_z} & \frac{lb_z}{2(1+b_z)} \\ \frac{b_z}{1+b_z} & -\frac{lb_z}{2(1+b_z)} \end{pmatrix} \begin{pmatrix} v_2 - v_1 \\ \theta_{z1} + \theta_{z2} \end{pmatrix} \tag{64}$$

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

$$\mathbf{K} = \frac{EI_z}{(1+b_z)l^3} \begin{pmatrix} 12 & -12 & 6l & 6l \\ & 12 & -6l & -6l \\ & & (4+b_z)l^2 & (2-b_z)l^2 \\ \text{[symmetry]} & & & (4+b_z)l^2 \end{pmatrix} \quad (65)$$

考虑所有分量的问题, 同时加入扭转以及拉压方向的问题

$$\mathbf{a} = \mathbf{a}_b + \mathbf{a}_s \quad (66)$$

有如下刚度阵

$$\begin{pmatrix} \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{l} & 0 & 0 & 0 & 0 & 0 \\ & \frac{12EI_z}{(1+b_z)l^3} & 0 & 0 & 0 & \frac{(1+b_z)6EI_z}{l^2} & 0 & -\frac{12EI_z}{l^3} & 0 & 0 & 0 & \frac{6EI_z}{(2-b_z)l^2} \\ & & \frac{12EI_y}{(1+b_y)l^3} & 0 & -\frac{6EI_z}{(1+b_y)l^2} & 0 & 0 & 0 & -\frac{12EI_y}{(1+b_y)l^3} & 0 & -\frac{6EI_z}{(1+b_y)l^2} & 0 \\ & & & \frac{GJ_z}{I} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ_z}{I} & 0 & 0 \\ & & & & \frac{(4+b_y)EI_y}{(1+b_y)l} & 0 & 0 & 0 & \frac{6EI_y}{(1+b_y)l} & 0 & \frac{(2-b_y)EI_y}{(1+b_y)l} & 0 \\ & \text{Symmetry} & & & & \frac{(4+b_z)EI_z}{(1+b_z)l} & 0 & -\frac{6EI_z}{(1+b_z)l} & 0 & 0 & 0 & \frac{(2-b_z)EI_z}{(1+b_z)l} \\ & & & & & & \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & \frac{12EI_z}{(1+b_z)l^3} & 0 & 0 & 0 & -\frac{6EI_z}{(1+b_z)l^2} \\ & & & & & & & & \frac{12EI_y}{(1+b_y)l^3} & 0 & \frac{6EI_y}{(1+b_y)l^2} & 0 \\ & & & & & & & & & \frac{GJ_z}{I} & 0 & 0 \\ & & & & & & & & & & \frac{(4-b_y)EI_y}{(1+b_y)l} & 0 \\ & & & & & & & & & & & \frac{(4-b_z)EI_z}{(1+b_z)l} \end{pmatrix} \quad (67)$$

同时单元应力的具体计算公式如下:

$$\begin{aligned} \sigma_{xx} &= E\varepsilon_{xx} = \frac{E(u_2 - u_1)}{l} \\ \sigma_{xy} &= \frac{G}{k}\gamma_{xy} \\ &= \frac{G}{kl} \frac{b_z}{1+b_z} \left((v_2 - v_1) - \frac{1}{2}(\theta_{z1} + \theta_{z2})l \right) \\ \sigma_{xz} &= \frac{G}{k}\gamma_{xz} \\ &= \frac{G}{kl} \frac{b_y}{1+b_y} \left((w_2 - w_1) + \frac{1}{2}(\theta_{y1} + \theta_{y2})l \right) \end{aligned} \quad (68)$$

0.5.4 程序实现

在 Stapp 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

$$K^e R d^e = R f^e \quad (69)$$

因此有

$$K^{global} = R^T K^{element} R \quad (70)$$

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

$$R = \begin{bmatrix} A & & & \\ & A & & \\ & & A & \\ & & & A \end{bmatrix} \quad (71)$$

其中 A=

$$\hat{A} = \begin{bmatrix} a_x a'_x & a_x a'_y & a_x a'_z \\ a_y a'_x & a_y a'_y & a_y a'_z \\ a_z a'_x & a_z a'_y & a_z a'_z \end{bmatrix} \quad (72)$$

将如此经过正交变换的单元刚度阵组装到总刚度阵即可

0.5.5 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：

$$\begin{aligned} v &= \frac{1}{2} a x^2 + c x + d \\ \theta_z &= a x + b \end{aligned} \quad (73)$$

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

$$w = 0.5x^2, \theta = x \quad (74)$$

分片试验结果如下：

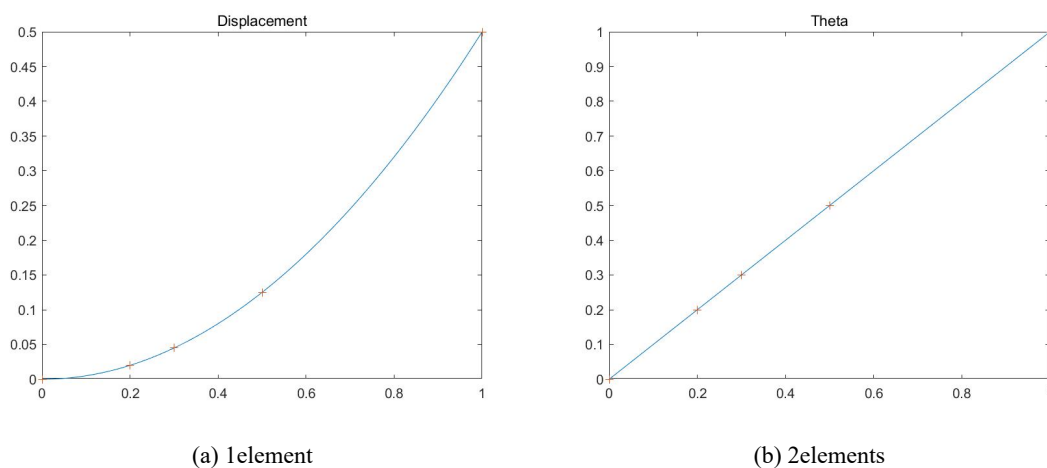


Figure 19 Timoshenko Beam Patch Test 1

其次对于纯剪应力状态，精确解为

$$w = -x^3 + 3x^2 + x, \theta = -3x^2 + 6x \quad (75)$$

分片试验结果如下：

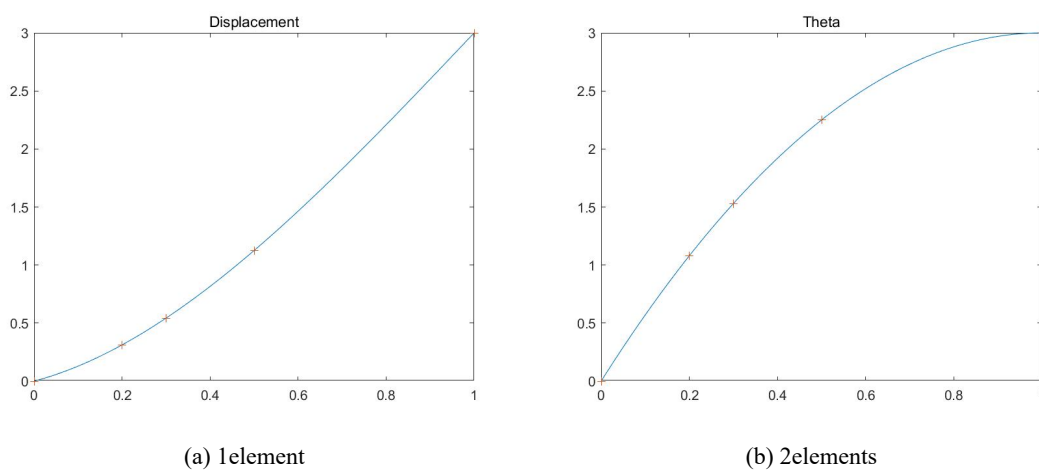


Figure 20 Timoshenko Beam Patch Test 2

0.5.6 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

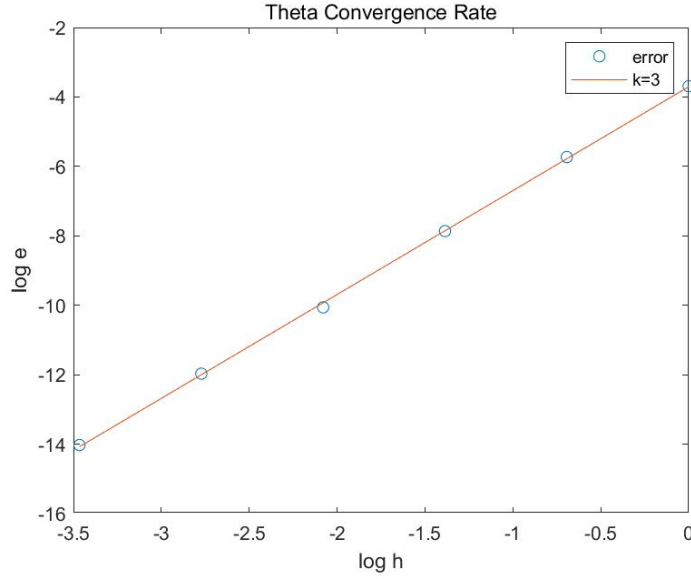


Figure 21 Convergence Rate

0.6 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。

0.6.1 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。

$$A_{\text{man}} = \begin{bmatrix} 0 & 4 & 0 & 7 & 0 \\ 2 & 0 & 3 & 0 & 6 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 6 & 0 \end{bmatrix} \quad (76)$$

$$Data = \begin{bmatrix} 4 & 7 & 2 & 3 & 6 & 5 & 2 & 1 & 6 \end{bmatrix} \quad (77)$$

$$Column - indices = \begin{bmatrix} 1 & 3 & 0 & 2 & 4 & 1 & 4 & 0 & 3 \end{bmatrix} \quad (78)$$

$$Ptr = \begin{bmatrix} 0 & 2 & 5 & 6 & 7 & 9 \end{bmatrix} \quad (79)$$

CSR 占据的存储空间为

$$CSR_{storage} = 2 \times NZV + m + 1 \quad (80)$$

0.6.2 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。

0.6.3 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。