

基于 STAP90 的有限元分析程序

小组成员： 黄维啸 祝凌霄 袁博
潘哲鑫 舒炫博

目录

一、 引言	3
二、 理论及程序分析	4
(一) 基础单元分析	4
1. 四节点四边形单元 (4Q)	4
2. 三节点三角形单元 (3T)	14
3. 八节点六面体单元 (8H)	21
4. 梁单元 (beam)	29
5. 板、壳单元 (plate&shell)	31
(二) 高级单元	35
1. 六节点三角形亚单元	35
2. 八节点四边形亚单元	37
(三) 其他功能	38
1. 模态的计算	38
2. 超收敛分片应力恢复 (SPR)	40
(四) 前处理	42
1. 简介	42
2. 实现过程	42
3. 具体示例	42
(五) 后处理	45
1. 简介	45
2. 实现过程	45
3. 具体示例	45
三、 分工与合作	46
四、 总结与反思	47

一、引言

STAP90 是有限元计算的经典程序，而现在我们得到的 STAP90 程序只能计算轴力杆单元，这对于应用是非常不便的。

为了使得程序更加强大，我们对程序进行了优化，添加了 4Q、3T、8H、BEAM、PLATE、SHELL 等基础单元计算功能，同时在基础单元基础上，我们添加了 6T、8Q 亚单元等高级单元的计算功能，还增加计算动力学模块，可以进行模态的计算与 SPR（应力恢复）处理。此外，还增加了针对 ANSYS 的前处理与针对 TECPLOT 的后处理，可直接在 ANSYS 中进行建模后导入程序算出结果，再由 TECPLOT 进行后处理绘制位移（应力）云图。

二、理论及程序分析

(一) 基础单元分析

1. 四节点四边形单元 (4Q)

1.1 理论基础

对于平面问题，其控制方程为

$$\sigma_{ij,j} + b_i = 0 \quad \text{in } \Omega$$

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})$$

$$\sigma_{ij} = C_{ijkl} \varepsilon_{kl}$$

$$\sigma_{ij} n_j = \bar{t}_i \quad \text{on } \Gamma_t$$

$$u_i = \bar{u}_i \quad \text{on } \Gamma_u$$

由此可以构造弱形式为

$$\int_{\Omega} w_{i,j} \sigma_{i,j} d\Omega = \int_{\Gamma_t} w_i \bar{t}_i d\Gamma + \int_{\Omega} w_i b_i d\Omega \quad \forall w_i \in U_0$$

由二维问题可知

$$u = \begin{bmatrix} u_x \\ u_y \end{bmatrix} \quad w = \begin{bmatrix} w_x \\ w_y \end{bmatrix} \quad \bar{t} = \begin{bmatrix} \bar{t}_x \\ \bar{t}_y \end{bmatrix} \quad b = \begin{bmatrix} b_x \\ b_y \end{bmatrix}$$

$$\varepsilon = [\varepsilon_{xx} \quad \varepsilon_{yy} \quad \gamma_{xy}]^T = \nabla_s u \quad \sigma = [\sigma_{xx} \quad \sigma_{yy} \quad \sigma_{xy}]^T$$

其中

$$\nabla_s = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}$$

又

$$\sigma = D\varepsilon$$

而对于不同问题而言则有不同的 D 矩阵：

平面应力问题

$$D = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$$

平面应变问题

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

轴对称问题

$$\begin{aligned} \varepsilon &= [\varepsilon_r \quad \varepsilon_z \quad \gamma_{rz} \quad \varepsilon_\theta]^T = \nabla_s u \\ \sigma &= [\sigma_r \quad \sigma_z \quad \tau_{rz} \quad \sigma_\theta]^T = D\varepsilon \end{aligned} \quad \nabla_s^T = \begin{bmatrix} \frac{\partial}{\partial r} & 0 & \frac{\partial}{\partial z} & \frac{1}{r} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial r} & 0 \end{bmatrix}$$

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

则弱形式可化为

$$\begin{aligned} \int_{\Omega} (\nabla_s w) D \nabla_s u d\Omega &= \int_{\Gamma_t} w^T \bar{t} d\Gamma + \int_{\Omega} w^T b d\Omega \\ \sum_{e=1}^{n_{el}} \left\{ \int_{\Omega} (\nabla_s w^e) D \nabla_s u^e d\Omega - \int_{\Gamma_t} w^{eT} \bar{t} d\Gamma - \int_{\Omega} w^{eT} b d\Omega \right\} &= 0 \end{aligned}$$

最后得到有限元求解的方程

$$w^T (Kd - f) = 0 \quad \forall w_F \quad K = \sum_{e=1}^{n_{el}} L^{eT} K^e L^e, \quad f = \sum_{e=1}^{n_{el}} L^{eT} f^e$$

对于单元的形状，在有限元中可以使用四边形单元，其本质上即为 4 结点线性二维单元。

1.2 程序实现

具体实现过程可见 QuadrElem.f90 文件

利用 4 结点单元的拉格朗日形函数

$$\begin{aligned} N_1^{4Q}(\xi, \eta) &= \frac{1}{4}(1-\xi)(1-\eta) & N_3^{4Q}(\xi, \eta) &= \frac{1}{4}(1+\xi)(1+\eta) \\ N_2^{4Q}(\xi, \eta) &= \frac{1}{4}(1+\xi)(1-\eta) & N_4^{4Q}(\xi, \eta) &= \frac{1}{4}(1-\xi)(1+\eta) \end{aligned}$$

既可以作为从母单元变换到具体单元的变换函数，也作为插值位移场的形函数。而位移的导数便可以用形函数的导数来表示，即

$$\frac{du}{dx} = B^e d^e$$

其中

$$B^e = \begin{bmatrix} B_1^e & B_2^e & B_3^e & B_4^e \end{bmatrix}$$

$$B_i^e = \begin{bmatrix} \frac{\partial N_i^{4Q}}{\partial x} & 0 \\ 0 & \frac{\partial N_i^{4Q}}{\partial y} \\ \frac{\partial N_i^{4Q}}{\partial y} & \frac{\partial N_i^{4Q}}{\partial x} \end{bmatrix}$$

由此可以得到

$$\nabla_s u^e = B^e d^e = B^e L^e d$$

$$\nabla_s w^e = B^e w^e = B^e L^e w$$

则单元刚度阵可以表示为

$$K = \sum_{e=1}^{n_{el}} L^{eT} K^e L^e \quad f = \sum_{e=1}^{n_{el}} L^{eT} f^e$$

$$K^e = \int_{\Omega} B^{eT} D B^e d\Omega \quad f^e = \int_{\Gamma_r^e} N^{eT} t d\Omega + \int_{\Omega} N^{eT} b d\Omega$$

由于形函数的阶次是给定的，因此在单元上利用能够精确积分形函数阶次的多项式的高斯积分，并可以得到单元刚度矩阵，即单元上的载荷。

利用 **Gather Matrixs** L ，再将单元刚度阵组装到，总体的刚度阵，以及总体的结点载荷。并可以通过求解矩阵

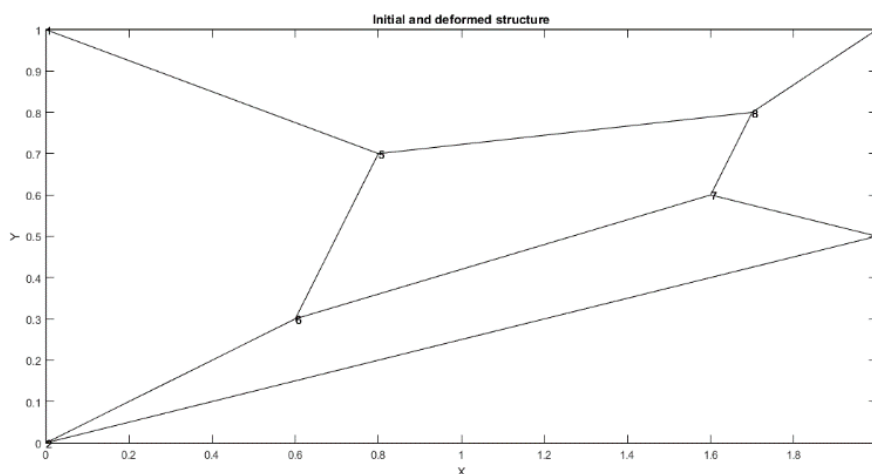
$$Kd = f$$

来得到各结点的位移。

1.3 验证

1.3.1 分片实验

我们对于4Q单元进行分片实验。我们考虑如下图所示结构，我们可以做如图所示划分。



对于此问题，我们构造一个线性的位移场如下

$$u = 2x \times 10^{-7}$$

$$v = (3x + y - 1) \times 10^{-7}$$

由此，我们可以算出每个节点各个自由度的位移值 d 。然后我们利用现有的正确的的结果如下：

d =	f2=	f1 =	f0 =
1.0e-06 *			
0	-0.3297	-1.4423	14.4997
0	3.5440	5.7692	40.2426
0	-5.3571	-5.9135	-0.6042
-0.1000	-6.1401	-8.3654	-46.6406
0.4000	0.3297	1.4423	40.4782
0.5500	-3.5440	-5.7692	-76.1715
0.4000	5.3571	5.9135	70.6858
0.6000	6.1401	8.3654	99.9147
0.1600	-0.0000	0.0000	0.0000
0.2100	-0.0000	-0.0000	-5.9992
0.1200	-0.0000	0.0000	-0.0000
0.1100	-0.0000	0.0000	-4.8030
0.3200	-0.0000	-0.0000	-0.0000
0.4400	0.0000	-0.0000	-3.1718
0.3400	0.0000	0.0000	-0.0000
0.4900	0.0000	0.0000	-3.3712

其中 f2、f1、f0 分别为平面应力、平面应变、轴对称问题的载荷信息。

根据得到的载荷信息，我们把此载荷信息写入输入文件中，即可以得到分片试验的算例

PT4Q_2.in、PT4Q_1.in、PT4Q_0.in

1.3.2 结果

1.3.2.1 平面应力

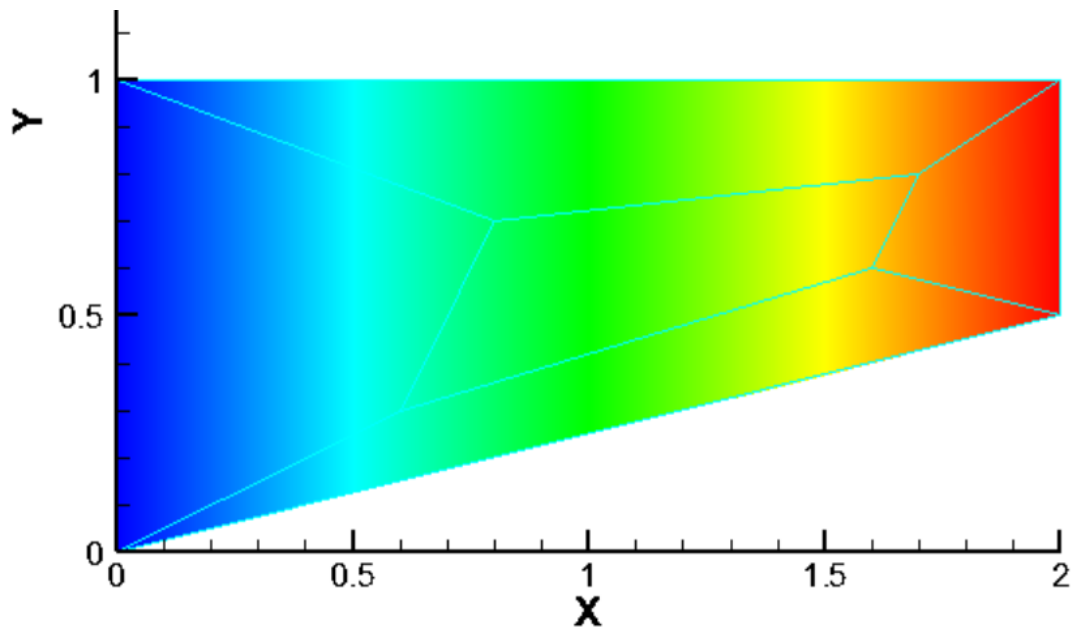
我们使用 PT4Q_2.in 作为输入文件，可以求解出如下所示位移信息

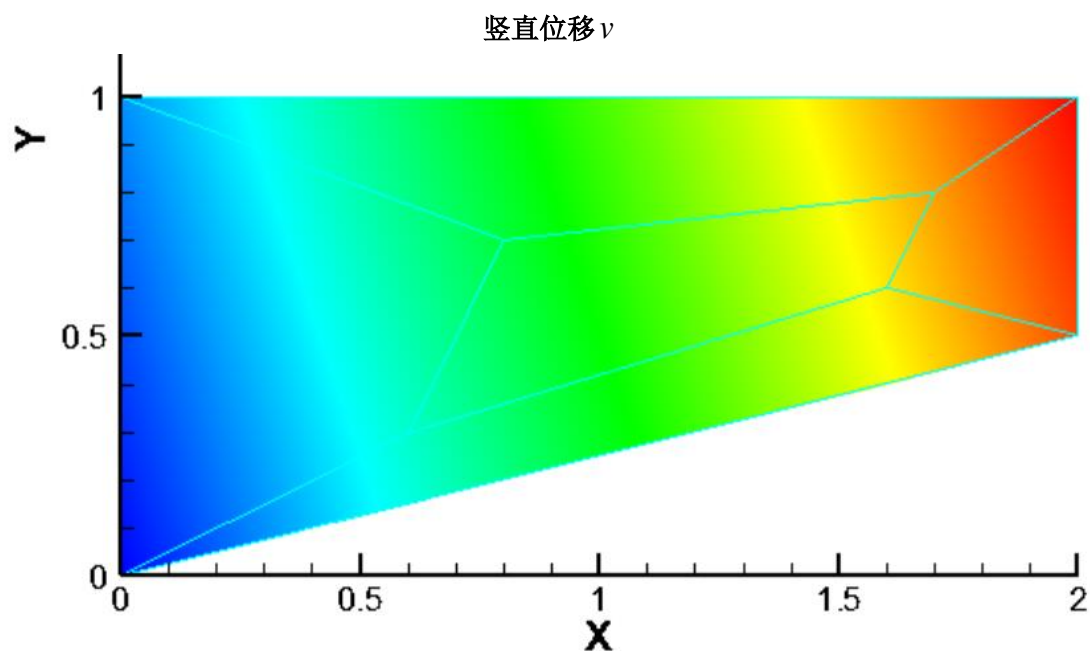
D I S P L A C E M E N T S

NODE	X-DISPLACEMENT	Y-DISPLACEMENT	Z-DISPLACEMENT
1	0.000000E+00	0.000000E+00	0.000000E+00
2	0.000000E+00	-0.100000E-01	0.000000E+00
3	0.399997E-01	0.549981E-01	0.000000E+00
4	0.400002E-01	0.599981E-01	0.000000E+00
5	0.160000E-01	0.209993E-01	0.000000E+00
6	0.119999E-01	0.109994E-01	0.000000E+00
7	0.319999E-01	0.439985E-01	0.000000E+00
8	0.340000E-01	0.489984E-01	0.000000E+00

利用此计算结果，比对位移向量 d ，可以看到结果非常的精确。同时利用 Tecplot 画图可以得到如下的位移云图

水平位移 u





可以看出线性性有了很大的满足。同时我们可以得到应力的信息如下

STRESS CALCULATIONS FOR ELEMENT GROUP 1

ELEMENT NUMBER	GAUSS POINT		StressXX	StressYY	StressXY
1	-0.577	-0.577	0.758242E+01	0.527473E+01	0.346143E+01
1	-0.577	0.577	0.758239E+01	0.527473E+01	0.346143E+01
1	0.577	-0.577	0.758242E+01	0.527474E+01	0.346145E+01
1	0.577	0.577	0.758238E+01	0.527473E+01	0.346145E+01
2	-0.577	-0.577	0.758237E+01	0.527474E+01	0.346145E+01
2	-0.577	0.577	0.758234E+01	0.527475E+01	0.346148E+01
2	0.577	-0.577	0.758239E+01	0.527474E+01	0.346144E+01
2	0.577	0.577	0.758235E+01	0.527475E+01	0.346149E+01
3	-0.577	-0.577	0.758237E+01	0.527474E+01	0.346153E+01
3	-0.577	0.577	0.758240E+01	0.527475E+01	0.346153E+01
3	0.577	-0.577	0.758237E+01	0.527474E+01	0.346152E+01
3	0.577	0.577	0.758243E+01	0.527476E+01	0.346151E+01
4	-0.577	-0.577	0.758244E+01	0.527472E+01	0.346150E+01
4	-0.577	0.577	0.758244E+01	0.527472E+01	0.346147E+01
4	0.577	-0.577	0.758241E+01	0.527471E+01	0.346151E+01
4	0.577	0.577	0.758242E+01	0.527472E+01	0.346145E+01
5	-0.577	-0.577	0.758241E+01	0.527474E+01	0.346146E+01
5	-0.577	0.577	0.758239E+01	0.527474E+01	0.346146E+01
5	0.577	-0.577	0.758240E+01	0.527474E+01	0.346149E+01
5	0.577	0.577	0.758236E+01	0.527473E+01	0.346150E+01

可以看出的确是接近于常应变单元，而且与理论值很吻合

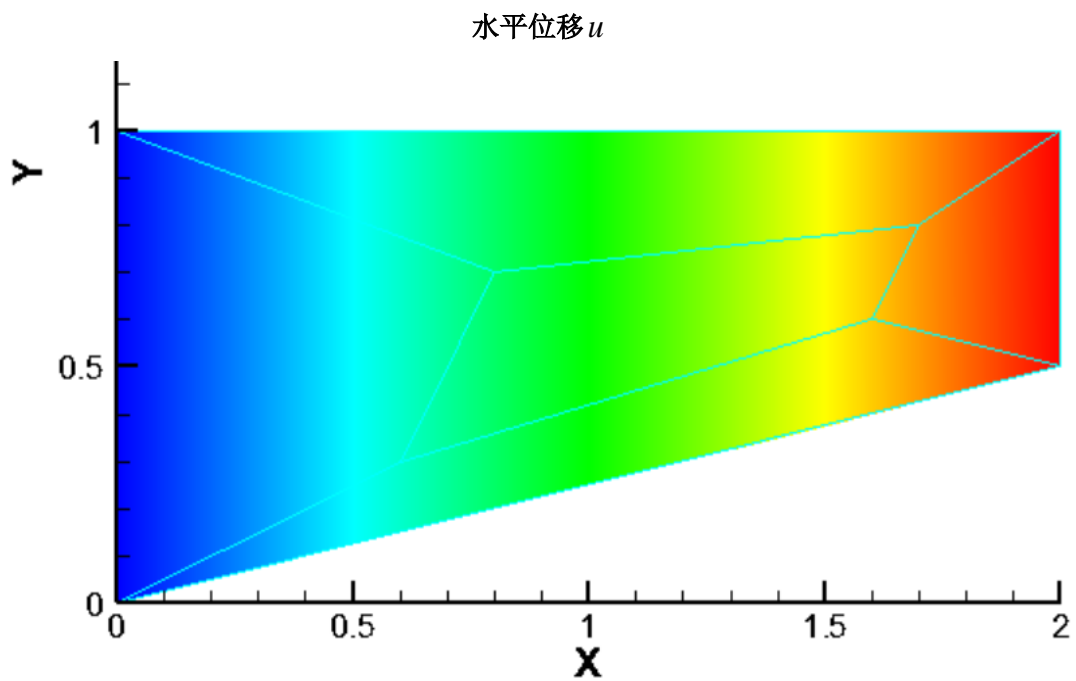
1.3.2.2 平面应变

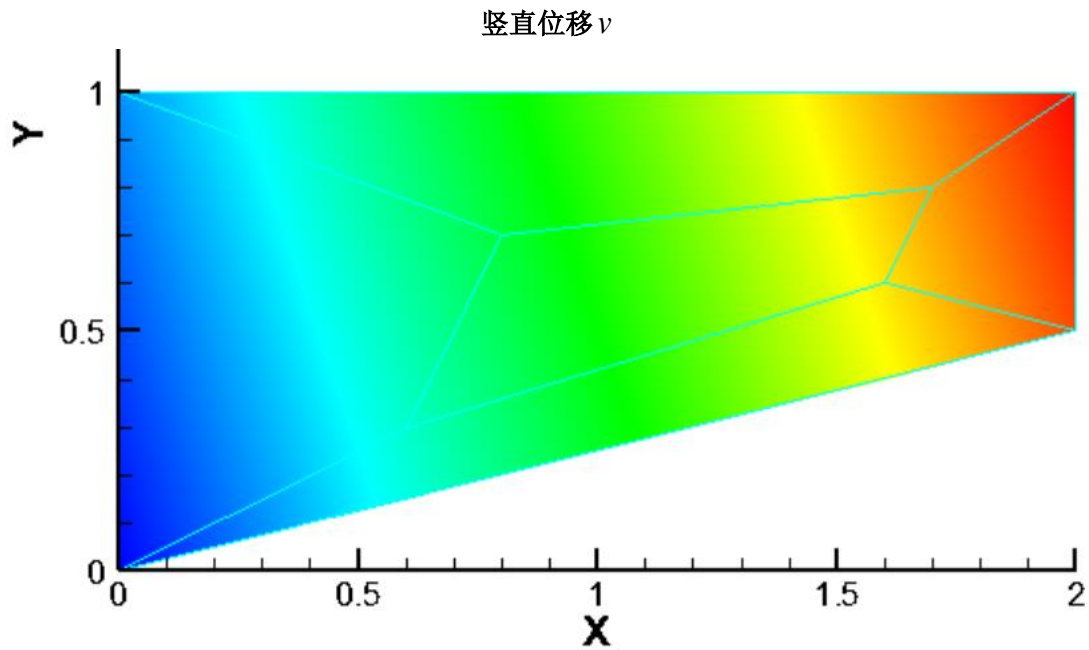
我们使用 PT4Q_1.in 作为输入文件，可以求解出如下所示位移信息

D I S P L A C E M E N T S

NODE	X-DISPLACEMENT	Y-DISPLACEMENT	Z-DISPLACEMENT
1	0.000000E+00	0.000000E+00	0.000000E+00
2	0.000000E+00	-0.999997E-02	0.000000E+00
3	0.400004E-01	0.550017E-01	0.000000E+00
4	0.399999E-01	0.600017E-01	0.000000E+00
5	0.160000E-01	0.210007E-01	0.000000E+00
6	0.120001E-01	0.110005E-01	0.000000E+00
7	0.320002E-01	0.440014E-01	0.000000E+00
8	0.340001E-01	0.490014E-01	0.000000E+00

利用此计算结果，比对位移向量 d ，可以看到结果非常的精确。同时利用 Tecplot 画图可以得到如下的位移云图





可以看出线性性有了很大的满足。同时我们可以得到应力的信息如下

STRESS CALCULATIONS FOR ELEMENT GROUP 1

ELEMENT NUMBER	GAUSS POINT		StressXX	StressYY	StressXY
1	-0.577	-0.577	0.980772E+01	0.750000E+01	0.346163E+01
1	-0.577	0.577	0.980775E+01	0.750001E+01	0.346163E+01
1	0.577	-0.577	0.980771E+01	0.749999E+01	0.346162E+01
1	0.577	0.577	0.980776E+01	0.750001E+01	0.346162E+01
2	-0.577	-0.577	0.980777E+01	0.749999E+01	0.346162E+01
2	-0.577	0.577	0.980779E+01	0.749999E+01	0.346158E+01
2	0.577	-0.577	0.980774E+01	0.749999E+01	0.346163E+01
2	0.577	0.577	0.980778E+01	0.749998E+01	0.346158E+01
3	-0.577	-0.577	0.980777E+01	0.750000E+01	0.346154E+01
3	-0.577	0.577	0.980773E+01	0.749998E+01	0.346154E+01
3	0.577	-0.577	0.980777E+01	0.750000E+01	0.346155E+01
3	0.577	0.577	0.980770E+01	0.749997E+01	0.346156E+01
4	-0.577	-0.577	0.980769E+01	0.750001E+01	0.346156E+01
4	-0.577	0.577	0.980769E+01	0.750001E+01	0.346160E+01
4	0.577	-0.577	0.980772E+01	0.750002E+01	0.346156E+01
4	0.577	0.577	0.980771E+01	0.750002E+01	0.346161E+01
5	-0.577	-0.577	0.980772E+01	0.749999E+01	0.346161E+01
5	-0.577	0.577	0.980775E+01	0.750000E+01	0.346160E+01
5	0.577	-0.577	0.980774E+01	0.749999E+01	0.346158E+01
5	0.577	0.577	0.980778E+01	0.750001E+01	0.346157E+01

可以看出的确是常应变单元，而且与理论值很吻合

1.3.2.3 轴对称问题

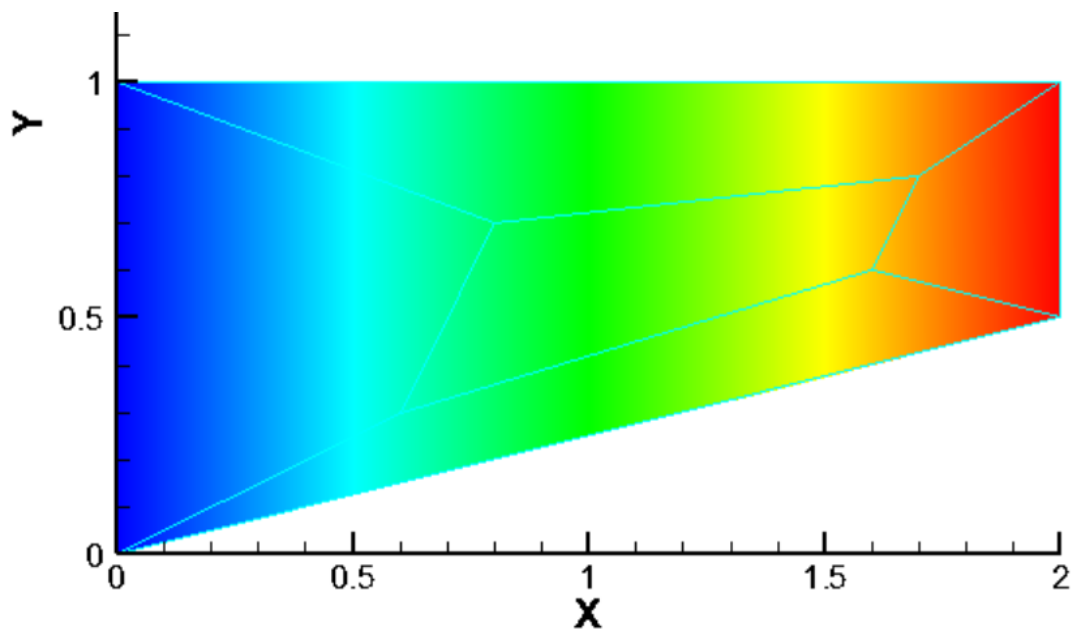
我们使用 PT4Q_0.in 作为输入文件，可以求解出如下所示位移信息

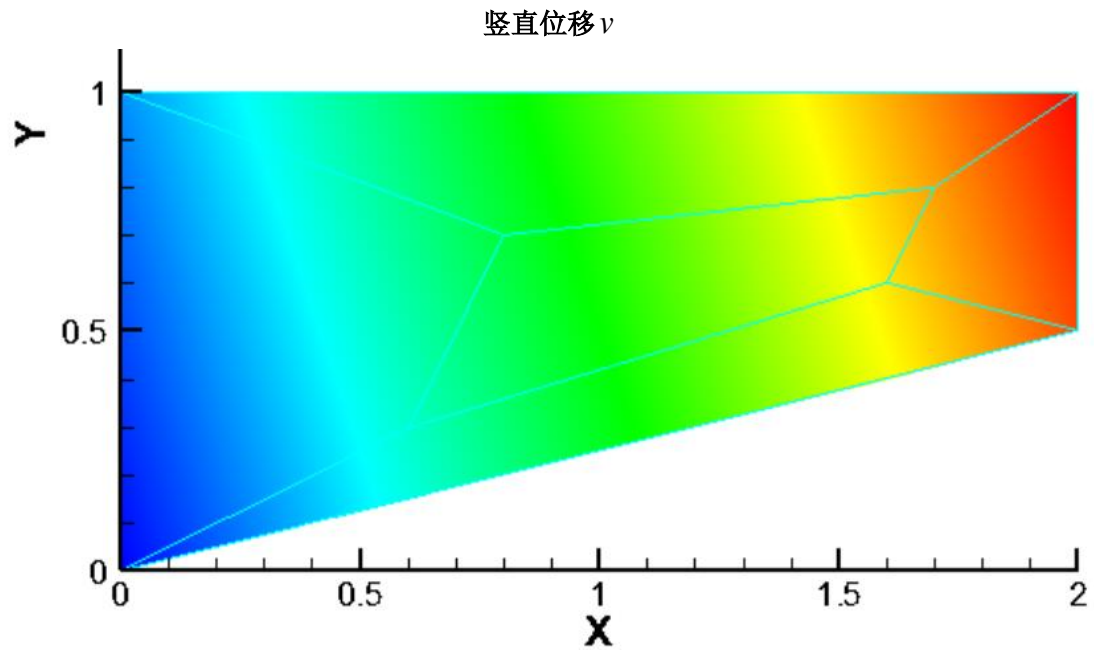
DISPLACEMENTS

NODE	X-DISPLACEMENT	Y-DISPLACEMENT	Z-DISPLACEMENT
1	0.000000E+00	0.000000E+00	0.000000E+00
2	0.000000E+00	-0.100001E-06	0.000000E+00
3	0.400000E-06	0.550001E-06	0.000000E+00
4	0.399999E-06	0.600001E-06	0.000000E+00
5	0.160000E-06	0.210000E-06	0.000000E+00
6	0.120000E-06	0.110000E-06	0.000000E+00
7	0.320000E-06	0.440000E-06	0.000000E+00
8	0.340000E-06	0.490000E-06	0.000000E+00

利用此计算结果，比对位移向量 d ，可以看到结果非常的精确。同时利用 Tecplot 画图可以得到如下的位移云图

水平位移 u





可以看出线性性有了很大的满足。同时我们可以得到应力的信息如下

STRESS CALCULATIONS FOR ELEMENT GROUP 1

ELEMENT NUMBER	GAUSS POINT		StressXX	StressYY	StressXY
1	-0.577	-0.577	0.132692E+02	0.109616E+02	0.346154E+01
1	-0.577	0.577	0.132692E+02	0.109616E+02	0.346154E+01
1	0.577	-0.577	0.132692E+02	0.109615E+02	0.346154E+01
1	0.577	0.577	0.132692E+02	0.109616E+02	0.346154E+01
2	-0.577	-0.577	0.132692E+02	0.109616E+02	0.346154E+01
2	-0.577	0.577	0.132692E+02	0.109615E+02	0.346154E+01
2	0.577	-0.577	0.132692E+02	0.109616E+02	0.346154E+01
2	0.577	0.577	0.132692E+02	0.109615E+02	0.346154E+01
3	-0.577	-0.577	0.132692E+02	0.109615E+02	0.346154E+01
3	-0.577	0.577	0.132692E+02	0.109615E+02	0.346154E+01
3	0.577	-0.577	0.132692E+02	0.109615E+02	0.346154E+01
3	0.577	0.577	0.132692E+02	0.109615E+02	0.346154E+01
4	-0.577	-0.577	0.132692E+02	0.109615E+02	0.346154E+01
4	-0.577	0.577	0.132692E+02	0.109615E+02	0.346154E+01
4	0.577	-0.577	0.132692E+02	0.109615E+02	0.346154E+01
4	0.577	0.577	0.132692E+02	0.109615E+02	0.346154E+01
5	-0.577	-0.577	0.132692E+02	0.109615E+02	0.346154E+01
5	-0.577	0.577	0.132692E+02	0.109615E+02	0.346154E+01
5	0.577	-0.577	0.132692E+02	0.109615E+02	0.346154E+01
5	0.577	0.577	0.132692E+02	0.109615E+02	0.346154E+01

可以看出的确是常应变单元，而且与理论值很吻合

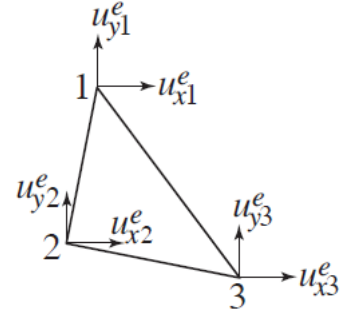
2. 三节点三角形单元（3T）

2.1 理论基础

对于平面问题，另一种思路可以用三角形单元作为网格划分的单元。我们考虑如右图所示的三角形单元，有

$$\mathbf{d}^e = [u_{x1}^e \ u_{y1}^e \ u_{x2}^e \ u_{y2}^e \ u_{x3}^e \ u_{y3}^e]^T$$

为了求解单元刚度阵，我们必须求出形函数。通过求解，我们得到了如下所示的形函数



$$N_1^e = \frac{1}{2A^e} [(x_2^e y_3^e - x_3^e y_2^e) + y_{23}^e x + x_{32}^e y]$$

$$N_2^e = \frac{1}{2A^e} [(x_3^e y_1^e - x_1^e y_3^e) + y_{31}^e x + x_{13}^e y]$$

$$N_3^e = \frac{1}{2A^e} [(x_1^e y_2^e - x_2^e y_1^e) + y_{12}^e x + x_{21}^e y]$$

对于形函数求导，我们就可以得到应变矩阵

$$\mathbf{B}^e = \frac{1}{2A^e} \begin{bmatrix} y_{23}^e & 0 & y_{31}^e & 0 & y_{12}^e & 0 \\ 0 & x_{32}^e & 0 & x_{13}^e & 0 & x_{21}^e \\ x_{32}^e & y_{23}^e & x_{13}^e & y_{31}^e & x_{21}^e & y_{12}^e \end{bmatrix}$$

于是可以得到单元的刚度矩阵

$$\mathbf{K}^e = \int_{\Omega^e} \mathbf{B}^{eT} \mathbf{D} \mathbf{B}^e d\Omega = t^e A^e \mathbf{B}^{eT} \mathbf{D} \mathbf{B}^e$$

2.2 程序实现

对于3节点的三角形单元，我们采用的是1阶的高斯积分，高斯点为三角形单元的中心。具体实现步骤为先根据不同的 NPAR4 计算 D 矩阵，然后根据以上的推导编写子程序

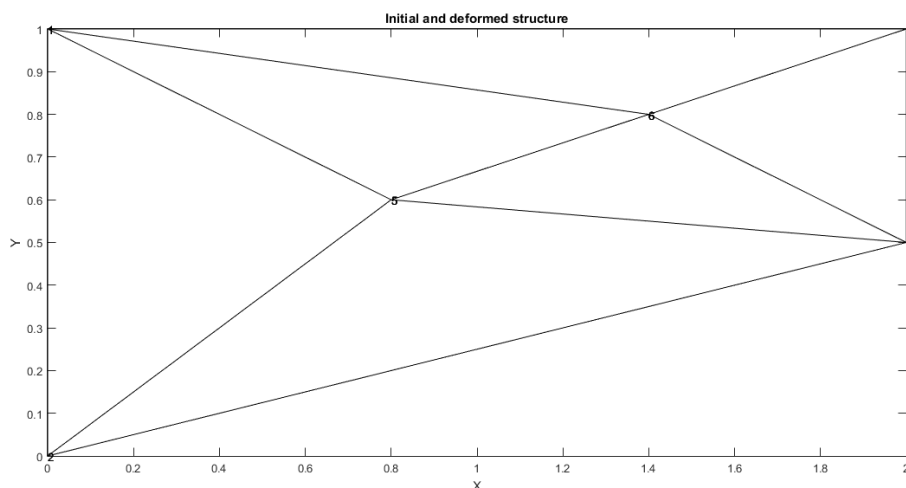
`BNmat (xy,ri,si,Be,Ne,detJ,r)`

来计算出 N 矩阵和 B 矩阵，然后可以得到单元刚度矩阵 S 的值，最后调用 `ADDBAN` 函数即可以组装刚度阵。

2.3 验证结果

2.3.1 分片实验

我们对于 3T 单元进行分片实验。我们考虑如右图所示结构，我们可以做如图所示划分，



对于此问题，我们构造一个线性的位移场如下

$$u = 2x \times 10^{-7}$$

$$v = (3x + y - 1) \times 10^{-7}$$

由此，我们可以算出每个节点各个自由度的位移值 d 。然后我们利用现有的正确的 MATLAB 程序算出此结构的刚度矩阵 K ，于是可以得到各自由度载荷 f ，在 MATLAB 中得到的结果如下：

$d =$	$f_2 =$	$f_1 =$	$f_0 =$
$1.0e-06 *$			
0	-0.6593	-2.8846	28.9993
0	7.0879	11.5385	80.2314
0	-10.7143	-11.8269	-1.2083
-0.1000	-12.2802	-16.7308	-96.1811
0.4000	0.6593	2.8846	80.9564
0.5500	-7.0879	-11.5385	-157.5630
0.4000	10.7143	11.8269	141.3717
0.6000	12.2802	16.7308	200.3369
0.1600	0.0000	0.0000	0.0000
0.2000	0.0000	0.0000	-16.6746
0.2800	0.0000	0.0000	0.0000
0.4000	0.0000	0.0000	-10.1498

其中 f_2 、 f_1 、 f_0 分别为平面应力、平面应变、轴对称问题的载荷信息。

根据得到的载荷信息，我们把此载荷信息写入输入文件中，即可以得到分片试验的算例 PT3T_2.in、PT3T_1.in、PT3T_0.in

2.3.2 结果

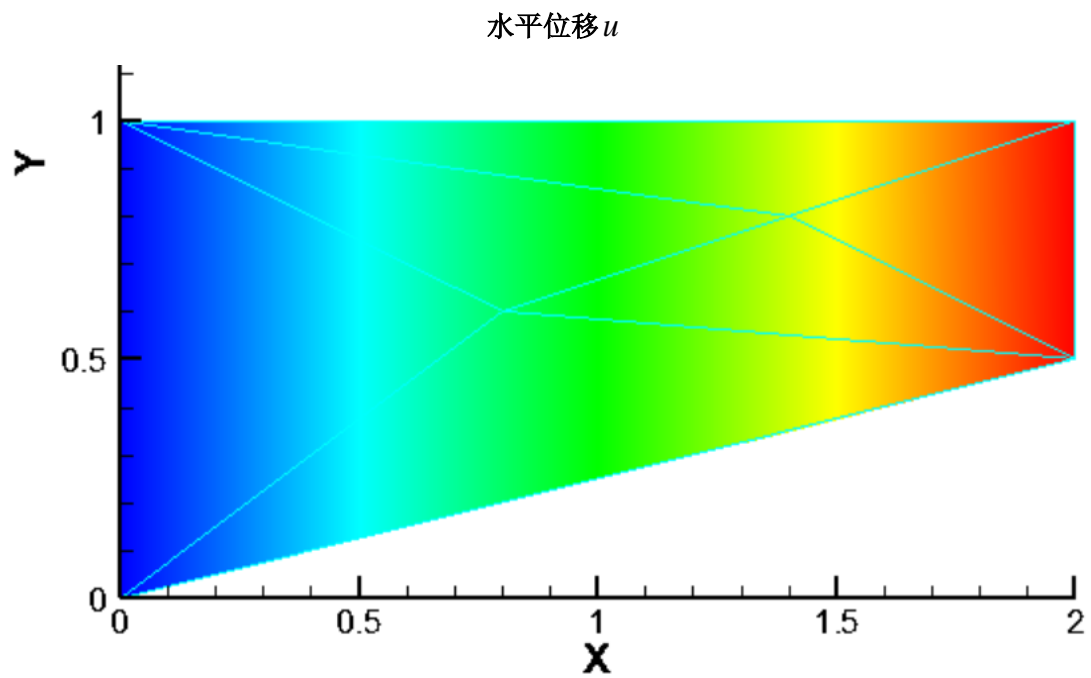
2.3.2.1 平面应力

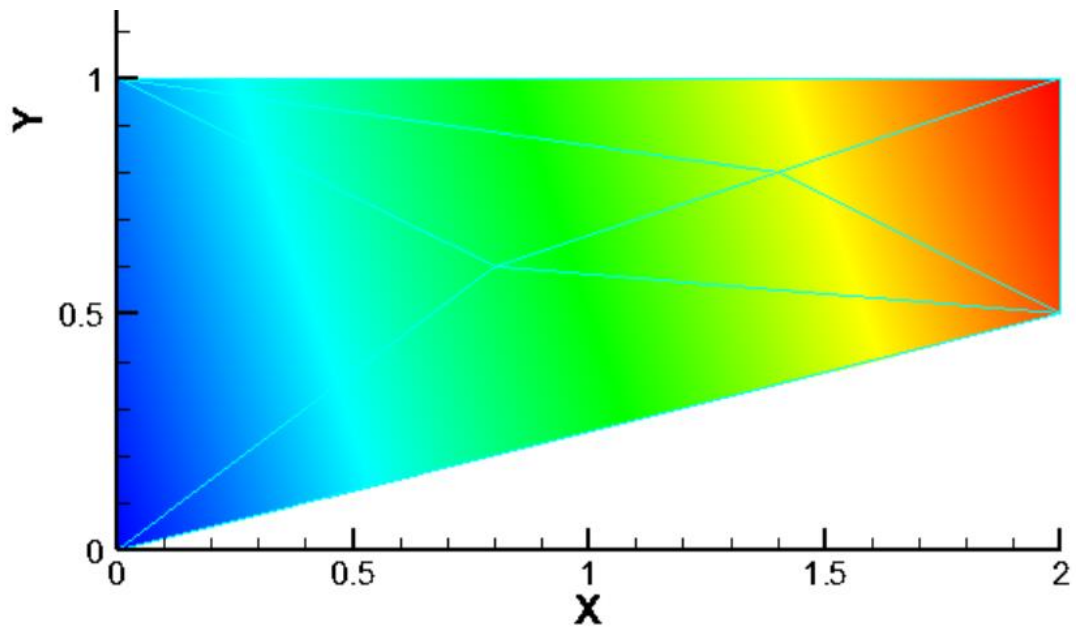
我们使用 PT3T_2.in 作为输入文件，可以求解出如下所示位移信息

D I S P L A C E M E N T S

NODE	X-DISPLACEMENT	Y-DISPLACEMENT	Z-DISPLACEMENT
1	0.000000E+00	0.000000E+00	0.000000E+00
2	0.000000E+00	-0.999999E-07	0.000000E+00
3	0.399998E-06	0.549997E-06	0.000000E+00
4	0.400000E-06	0.599997E-06	0.000000E+00
5	0.160000E-06	0.199999E-06	0.000000E+00
6	0.280000E-06	0.399998E-06	0.000000E+00

利用此计算结果，比对位移向量 d ，可以看到结果非常的精确。同时利用 Tecplot 画图可以得到如下的位移云图



竖直位移 v 

可以看出线性性有了很好的满足。同时我们可以得到应力的信息如下

STRESS CALCULATIONS FOR ELEMENT GROUP 1

ELEMENT NUMBER	GAUSS POINT		StressXX	StressYY	StressXY
1	0.267	0.533	0.758240E+01	0.527472E+01	0.346152E+01
2	0.933	0.367	0.758239E+01	0.527472E+01	0.346153E+01
3	1.400	0.633	0.758239E+01	0.527472E+01	0.346155E+01
4	1.800	0.767	0.758240E+01	0.527472E+01	0.346156E+01
5	1.133	0.933	0.758242E+01	0.527471E+01	0.346154E+01
6	0.733	0.800	0.758241E+01	0.527471E+01	0.346153E+01

可以看出的确是常应变单元，而且与理论值很吻合

2.3.2.2 平面应变

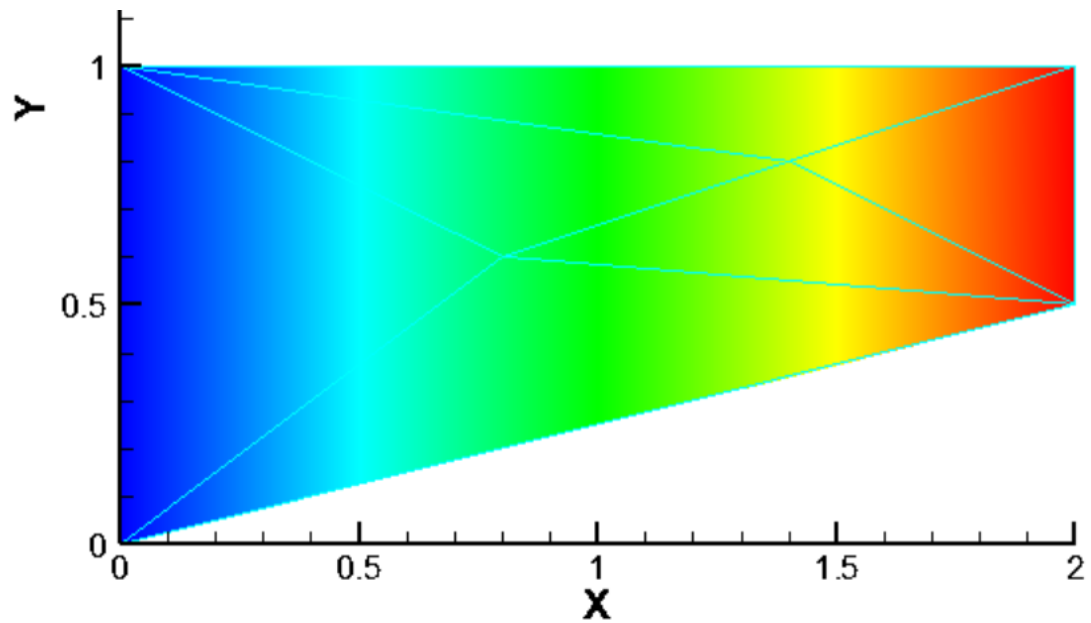
我们使用 PT3T_1.in 作为输入文件，可以求解出如下所示位移信息

DISPLACEMENTS

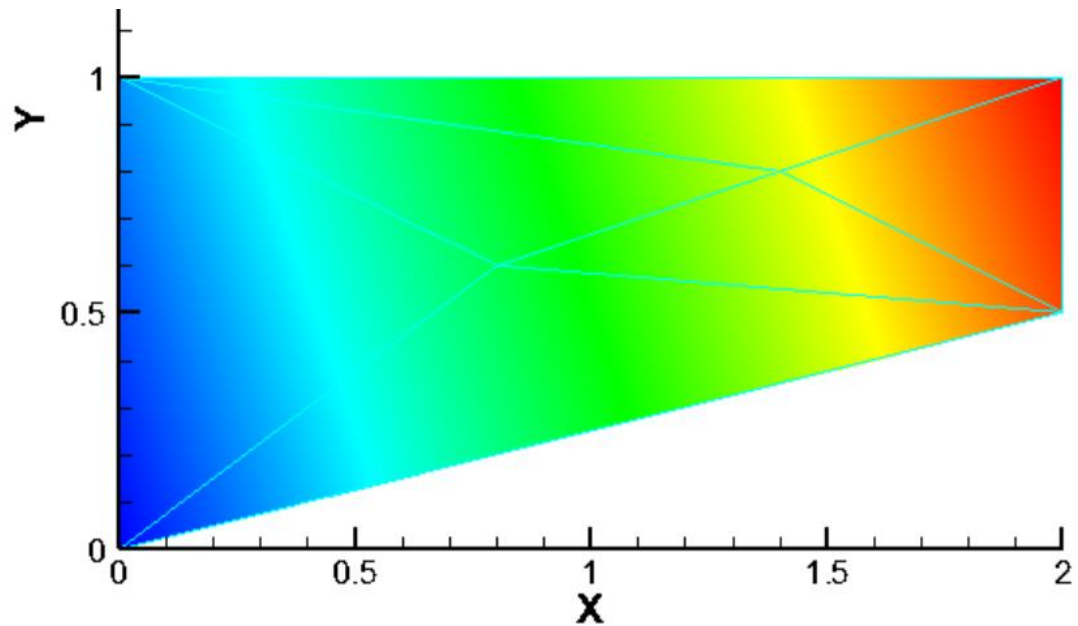
NODE	X-DISPLACEMENT	Y-DISPLACEMENT	Z-DISPLACEMENT
1	0.000000E+00	0.000000E+00	0.000000E+00
2	0.000000E+00	-0.100001E-06	0.000000E+00
3	0.399998E-06	0.549999E-06	0.000000E+00
4	0.399998E-06	0.599999E-06	0.000000E+00
5	0.159999E-06	0.199999E-06	0.000000E+00
6	0.279999E-06	0.399999E-06	0.000000E+00

利用此计算结果，比对位移向量 \mathbf{d} ，可以看到结果非常的精确。同时利用 Tecplot 画图可以得到如下的位移云图

水平位移 u



竖直位移 v



可以看出线性性有了很大的满足。同时我们可以得到应力的信息如下

STRESS CALCULATIONS FOR ELEMENT GROUP 1

ELEMENT NUMBER	GAUSS POINT		StressXX	StressYY	StressXY
1	0.267	0.533	0.980767E+01	0.750002E+01	0.346153E+01
2	0.933	0.367	0.980767E+01	0.750002E+01	0.346153E+01
3	1.400	0.633	0.980767E+01	0.750002E+01	0.346153E+01
4	1.800	0.767	0.980767E+01	0.750002E+01	0.346153E+01
5	1.133	0.933	0.980767E+01	0.750002E+01	0.346153E+01
6	0.733	0.800	0.980767E+01	0.750002E+01	0.346153E+01

可以看出的确是常应变单元，而且与理论值很吻合

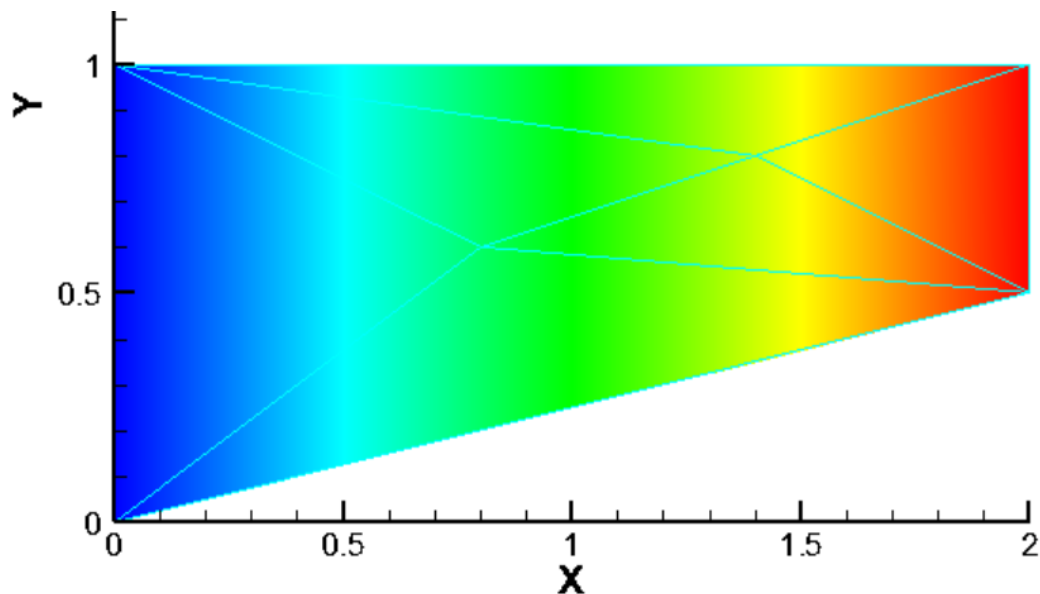
2.3.2.3 轴对称问题

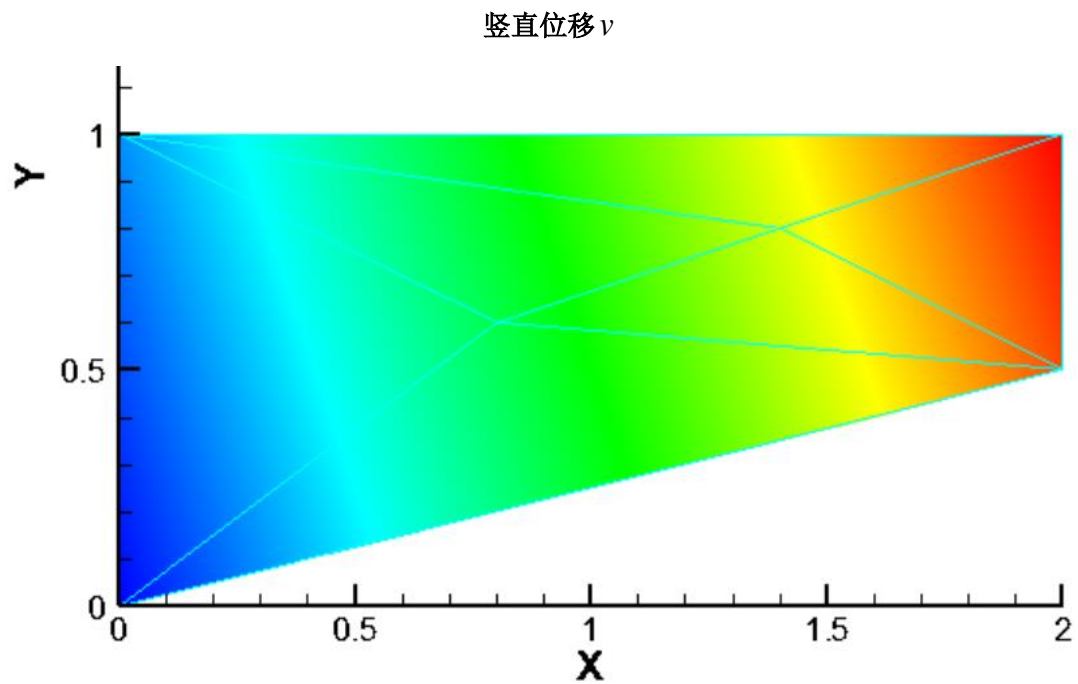
我们使用 PT3T_0.in 作为输入文件，可以求解出如下所示位移信息

DISPLACEMENTS

NODE	X-DISPLACEMENT	Y-DISPLACEMENT	Z-DISPLACEMENT
1	0.000000E+00	0.000000E+00	0.000000E+00
2	0.000000E+00	-0.100001E-06	0.000000E+00
3	0.399999E-06	0.549994E-06	0.000000E+00
4	0.400001E-06	0.599993E-06	0.000000E+00
5	0.160000E-06	0.199997E-06	0.000000E+00
6	0.280000E-06	0.399995E-06	0.000000E+00

利用此计算结果，比对位移向量 d ，可以看到结果非常的精确。同时利用 Tecplot 画图可以得到如下的位移云图

水平位移 u 



可以看出线性性有了很好的满足。同时我们可以得到应力的信息如下

S T R E S S C A L C U L A T I O N S F O R E L E M E N T G R O U P 1

ELEMENT NUMBER	GAUSS POINT		StressXX	StressYY	StressXY	
1	0.267	0.533	0.132692E+02	0.109616E+02	0.346150E+01	0.132692E+02
2	0.933	0.367	0.132692E+02	0.109615E+02	0.346151E+01	0.132692E+02
3	1.400	0.633	0.132692E+02	0.109615E+02	0.346154E+01	0.132692E+02
4	1.800	0.767	0.132692E+02	0.109615E+02	0.346155E+01	0.132692E+02
5	1.133	0.933	0.132693E+02	0.109615E+02	0.346153E+01	0.132693E+02
6	0.733	0.800	0.132693E+02	0.109616E+02	0.346151E+01	0.132692E+02

可以看出的确是常应变单元，而且与理论值很吻合

3. 八节点六面体单元（8H）

3.1 理论基础

8 节点六面体三维单元（简称 8H 单元）如图所示，各节点有 3 个自由度，则单元共有 24 个节点位移，即 8 个节点在 3 个方向上的分量。设节点的自然坐标为 ξ 、 η 、 ζ ，则单元节点的形函数如下：

$$N_1 = \frac{1}{8}(1-\xi)(1-\eta)(1-\zeta)$$

$$N_2 = \frac{1}{8}(1+\xi)(1-\eta)(1-\zeta)$$

$$N_3 = \frac{1}{8}(1+\xi)(1+\eta)(1-\zeta)$$

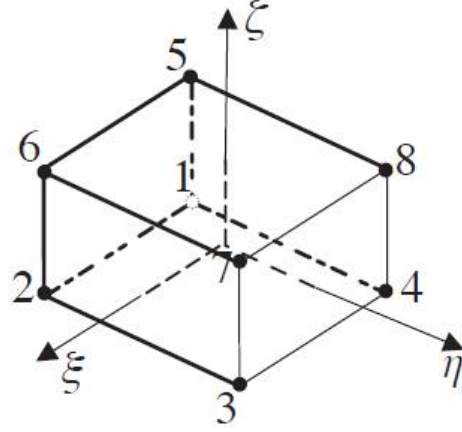
$$N_4 = \frac{1}{8}(1-\xi)(1+\eta)(1-\zeta)$$

$$N_5 = \frac{1}{8}(1-\xi)(1-\eta)(1+\zeta)$$

$$N_6 = \frac{1}{8}(1+\xi)(1-\eta)(1+\zeta)$$

$$N_7 = \frac{1}{8}(1+\xi)(1+\eta)(1+\zeta)$$

$$N_8 = \frac{1}{8}(1-\xi)(1+\eta)(1+\zeta)$$



由形函数可以求出应变矩阵 $\mathbf{B} = [\mathbf{B}_1 \quad \mathbf{B}_2 \quad \mathbf{B}_3 \quad \mathbf{B}_4 \quad \mathbf{B}_5 \quad \mathbf{B}_6 \quad \mathbf{B}_7 \quad \mathbf{B}_8]$ ，其中各子块为

$$\mathbf{B}_i = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial z} \\ 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \end{bmatrix}$$

在物理坐标系下求导时，需利用关系：

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = \mathbf{J}^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{Bmatrix}, \text{ 其中 } \mathbf{J} \text{ 为 Jacobi 矩阵。}$$

之后利用 Gauss 积分即可求出单元刚度阵：

$$\mathbf{K}^e = \int \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega^e = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \mathbf{B}^T \mathbf{D} \mathbf{B} \det \mathbf{J} d\xi d\eta d\zeta$$

其中三维空间下的弹性矩阵为：

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & & & \\ \nu & 1-\nu & \nu & & & \\ \nu & \nu & 1-\nu & & & \\ & & & \frac{1-2\nu}{2} & & \\ & & & & \frac{1-2\nu}{2} & \\ & & & & & \frac{1-2\nu}{2} \end{bmatrix}$$

之后将单元刚度阵 \mathbf{K}^e 组装即得到总刚度矩阵 \mathbf{K} ，由此通过方程 $\mathbf{K}d = f$ 即可解出整体位移矩阵 d 。

再由应力应变关系得到：

$$\begin{aligned} \varepsilon^e &= \mathbf{B}d^e \\ \sigma^e &= \mathbf{D}\mathbf{B}d^e \end{aligned}$$

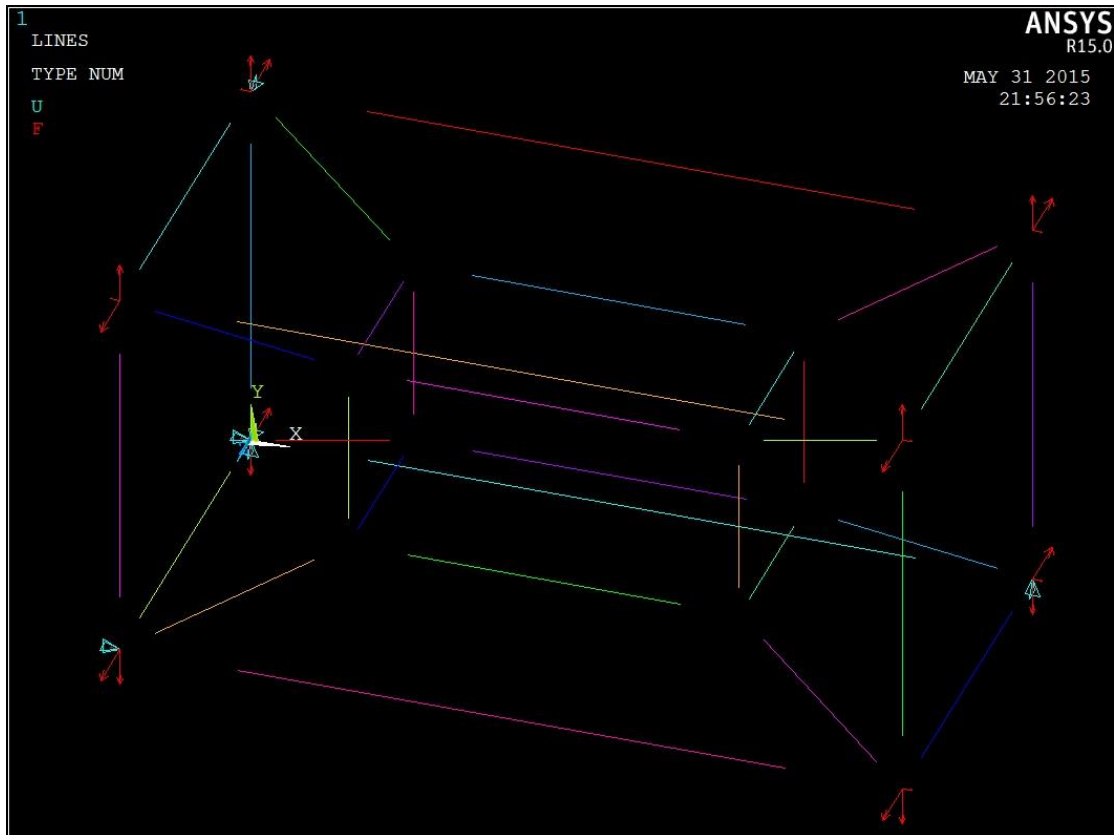
可以解出整体应变和应力场，但由有限元法计算的结果只有在高斯点上应力结果准确。

3.2 程序实现

8H 单元的实现程序具体见 hexahedral8.f90, 与其他单元不同之处主要在于形函数矩阵 \mathbf{N} 和应变矩阵 \mathbf{B} 的组成，通过调用函数 `Bmatr(xi,eta,zeta,XY,B,detJ)` 和 `Nmatr(xi,eta,zeta,Ne)` 可以构建出 8H 单元的相应矩阵，其中坐标采用二阶高斯积分的高斯点进行计算。由此可以得到各单元的刚度阵 \mathbf{S} ，再调用 `ADDBAN` 函数即可组装出刚度阵，由前所述，即可解出位移矩阵 d 以及应力、应变场。

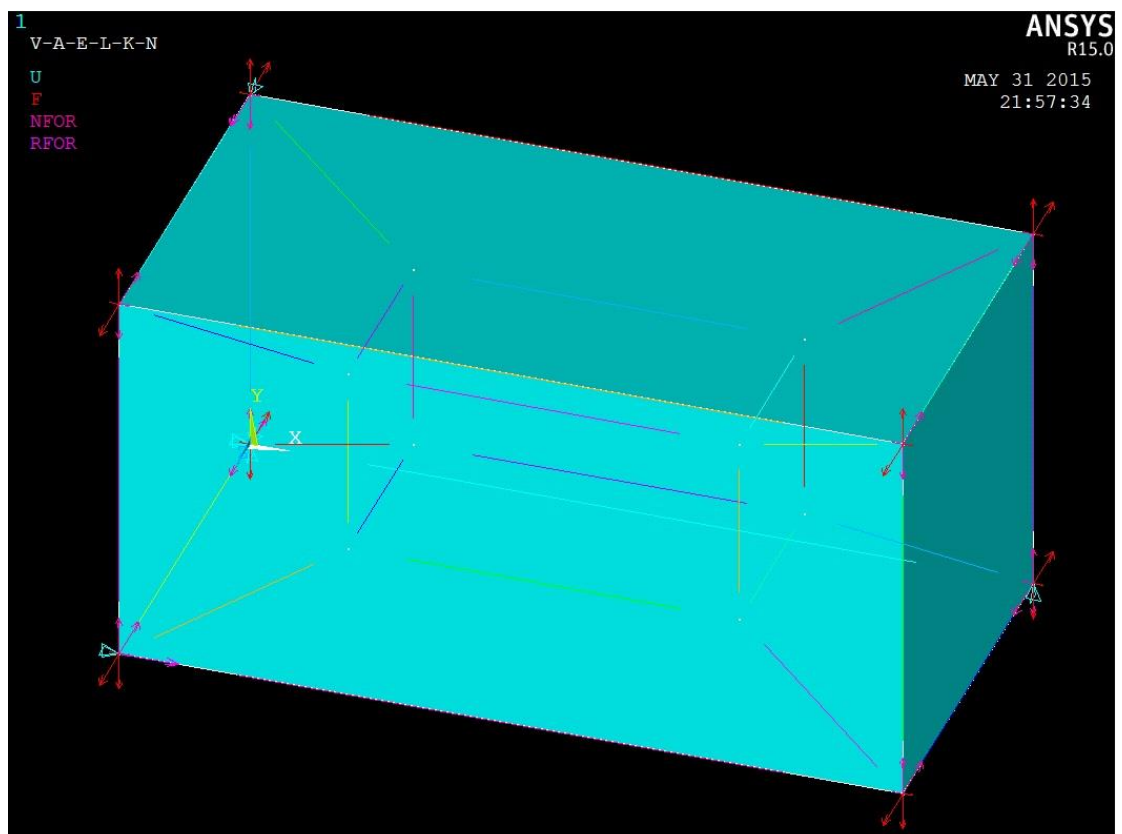
3.3 验证

对 8H 单元进行分片试验，做划分如下图：



即两个六面体相互嵌套，构成 7 个 8H 单元，周围的 6 个单元存在畸变，可以进行分片试验。

构造线性的位移场如下：



$$u = -5 \times 10^{-4} x$$

$$v = 8 \times 10^{-4} y$$

$$w = 2.1 \times 10^{-3} z$$

根据该位移场可以计算出各节点 3 个自由度上的位移 d ，再利用正确的 MATLAB 程序构建出该分片试验结构的刚度矩阵 K ，由

$$Kd = f$$

计算出各节点自由度的载荷 f ，具体结果如下：

$d =$	$f =$
0	-0.25
0	-1
0	-1.5
-1×10^{-3}	0.25
0	-1
0	-1.5
-1×10^{-3}	0.25
8×10^{-4}	1
0	-1.5
0	-0.25
8×10^{-4}	1
0	-1.5
0	-0.25
0	-1
2.1×10^{-3}	1.5
-1×10^{-3}	0.25
0	-1
2.1×10^{-3}	1.5
-1×10^{-3}	0.25
8×10^{-4}	1
2.1×10^{-3}	1.5
0	-0.25
8×10^{-4}	1
2.1×10^{-3}	1.5
-2.5×10^{-4}	0
2×10^{-4}	0
5.25×10^{-4}	0
-7.5×10^{-4}	0
2×10^{-4}	0
5.25×10^{-4}	0
-7.5×10^{-4}	0
6×10^{-4}	0
5.25×10^{-4}	0
-2.5×10^{-4}	0

6×10^{-4}	0
5.25×10^{-4}	0
-2.5×10^{-4}	0
2×10^{-4}	0
1.6×10^{-3}	0
-7.5×10^{-4}	0
2×10^{-4}	0
1.6×10^{-3}	0
-7.5×10^{-4}	0
6×10^{-4}	0
1.6×10^{-3}	0
-2.5×10^{-4}	0
6×10^{-4}	0
1.6×10^{-3}	0

将此载荷信息 f 写入输入文件，利用编写的 STAP90 程序运行，即可在输出文件 STAP90.OUT 中检验分片试验的结果如下：

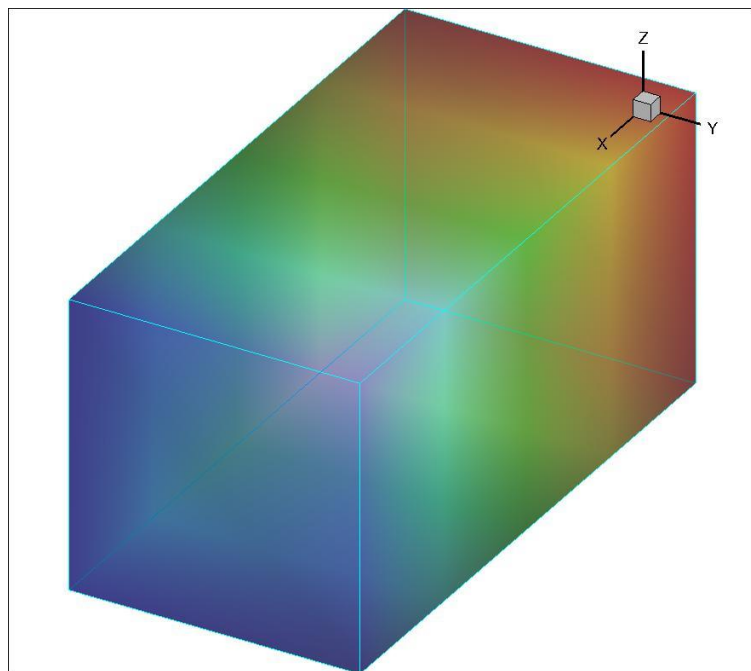
DISPLACEMENTS

NODE	X-DISPLACEMENT	Y-DISPLACEMENT	Z-DISPLACEMENT
1	0.000000E+00	0.000000E+00	0.000000E+00
2	-0.100000E-02	0.000000E+00	-0.510930E-16
3	-0.100000E-02	0.800000E-03	0.445643E-16
4	0.155041E-16	0.800000E-03	0.000000E+00
5	-0.750000E-03	0.200000E-03	0.525000E-03
6	-0.250000E-03	0.200000E-03	0.525000E-03
7	-0.750000E-03	0.600000E-03	0.525000E-03
8	-0.250000E-03	0.600000E-03	0.525000E-03
9	-0.250000E-03	0.200000E-03	0.157500E-02
10	-0.750000E-03	0.200000E-03	0.157500E-02
11	-0.750000E-03	0.600000E-03	0.157500E-02
12	-0.250000E-03	0.600000E-03	0.157500E-02
13	0.120346E-16	0.800000E-03	0.210000E-02
14	0.000000E+00	-0.208167E-16	0.210000E-02
15	-0.100000E-02	0.800000E-03	0.210000E-02
16	-0.100000E-02	-0.915067E-16	0.210000E-02

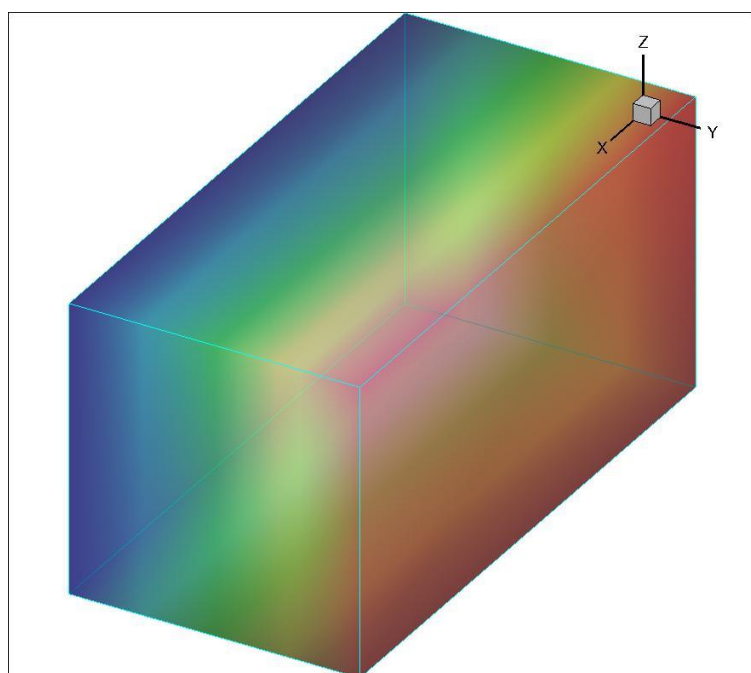
将结果与位移向量 d 进行对比，可以看出结果精度很高。

同时可以利用 Tecplot 进行后处理绘制位移云图如：

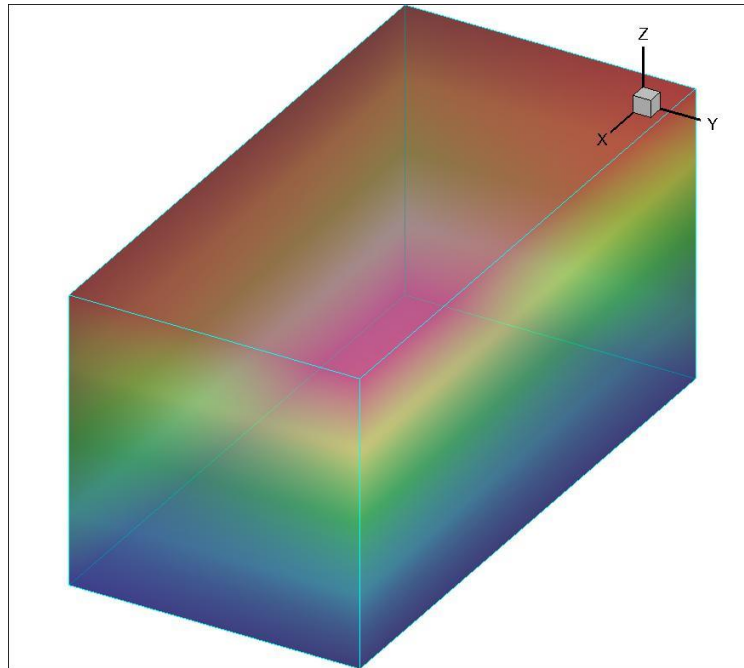
x 方向位移 u



y 方向位移 v



z 方向位移 w



可见结果确为线性位移场，结果的可信性比较高。

另一方面，我们可以通过 STAP90 程序计算出单元中的应力信息如下：

STRESS CALCULATIONS FOR ELEMENT GROUP 1

ELEMENT NUMBER	GAUSS POINT			StressXX	StressYY	StressZZ
1	-0.577	-0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
1	-0.577	-0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
1	-0.577	0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
1	-0.577	0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
1	0.577	-0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
1	0.577	-0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
1	0.577	0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
1	0.577	0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
2	-0.577	-0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
2	-0.577	-0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
2	-0.577	0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
2	-0.577	0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
2	0.577	-0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
2	0.577	-0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
2	0.577	0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
2	0.577	0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
3	-0.577	-0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
3	-0.577	-0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
3	-0.577	0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
3	-0.577	0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02

3	0.577	-0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
3	0.577	-0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
3	0.577	0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
3	0.577	0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
4	-0.577	-0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
4	-0.577	-0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
4	-0.577	0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
4	-0.577	0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
4	0.577	-0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
4	0.577	-0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
4	0.577	0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
4	0.577	0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
5	-0.577	-0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
5	-0.577	-0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
5	-0.577	0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
5	-0.577	0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
5	0.577	-0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
5	0.577	-0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
5	0.577	0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
5	0.577	0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
6	-0.577	-0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
6	-0.577	-0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
6	-0.577	0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
6	-0.577	0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
6	0.577	-0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
6	0.577	-0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
6	0.577	0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
6	0.577	0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
7	-0.577	-0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
7	-0.577	-0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
7	-0.577	0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
7	-0.577	0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
7	0.577	-0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
7	0.577	-0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02
7	0.577	0.577	-0.577	0.400000E+01	0.800000E+01	0.120000E+02
7	0.577	0.577	0.577	0.400000E+01	0.800000E+01	0.120000E+02

可见各单元应力均为常数，确为常应力单元，与理论结果相符。

由上可见，分片试验结果较好。

4. 梁单元 (beam)

4.1 理论基础

在梁模型中，我们可以采用 8H 结构进行计算，不过，对于截面很小的梁问题，建立相应的梁单元可以更好地逼近结果。这里我们采用 2 节点普通弯曲梁进行分析。

对于如右图所示的模型，每个节点取杆向位移 u_x ，挠度 u_y 以及转角 θ ，则每个单元的位移向量为：

$$d^e = [u_{x1} \ u_{y1} \ \theta_1 \ u_{x2} \ u_{y2} \ \theta_2]^T$$

其中 u_x 为横向位移，计算方式与 TRUSS 单元相同。对于挠度 u_y 以及转角 θ ，有如下形函数：

$$N_{u1} = \frac{1}{4}(1-\xi)^2(2+\xi)$$

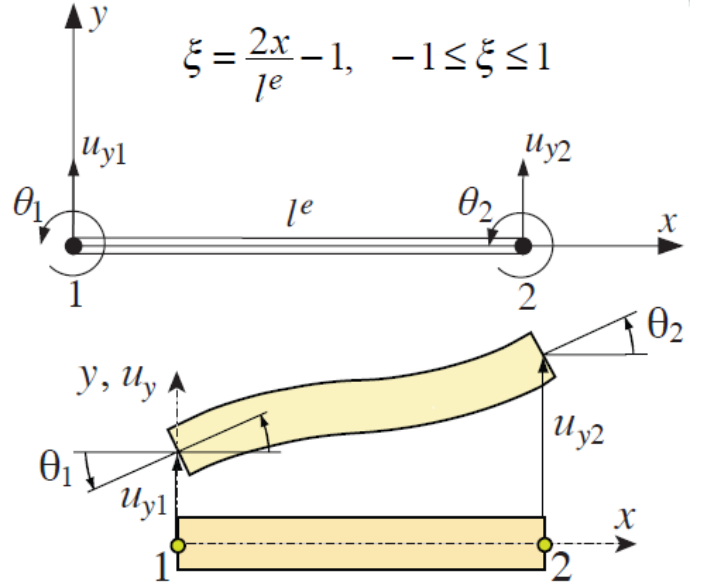
$$N_{\theta1} = \frac{l^e}{8}(1-\xi)^2(1+\xi)$$

$$N_{u2} = \frac{1}{4}(1+\xi)^2(2-\xi)$$

$$N_{\theta2} = \frac{l^e}{8}(1+\xi)^2(\xi-1)$$

$$N^e = [N_{u1} \ N_{\theta1} \ N_{u2} \ N_{\theta2}]$$

$$\frac{d^2 N^e}{dx^2} = \frac{1}{l^e} \underbrace{\begin{bmatrix} \frac{6\xi}{l^e} & 3\xi-1 & -\frac{6\xi}{l^e} & 3\xi+1 \end{bmatrix}}_{B^e}$$



则刚度矩阵为： $K^e = \int_{\Omega} EIB^{eT} B^e dx$ ，即：

$$K^e = \begin{bmatrix} \frac{EA}{l^e} & 0 & 0 & -\frac{EA}{l^e} & 0 & 0 \\ 0 & \frac{12EI}{l^{e3}} & \frac{6EI}{l^{e2}} & 0 & -\frac{12EI}{l^{e3}} & \frac{6EI}{l^{e2}} \\ 0 & \frac{6EI}{l^{e2}} & \frac{4EI}{l^e} & 0 & -\frac{6EI}{l^{e2}} & \frac{2EI}{l^e} \\ -\frac{EA}{l^e} & 0 & 0 & \frac{EA}{l^e} & 0 & 0 \\ 0 & -\frac{12EI}{l^{e3}} & -\frac{6EI}{l^{e2}} & 0 & \frac{12EI}{l^{e3}} & -\frac{6EI}{l^{e2}} \\ 0 & \frac{6EI}{l^{e2}} & \frac{2EI}{l^e} & 0 & -\frac{6EI}{l^{e2}} & \frac{4EI}{l^e} \end{bmatrix}$$

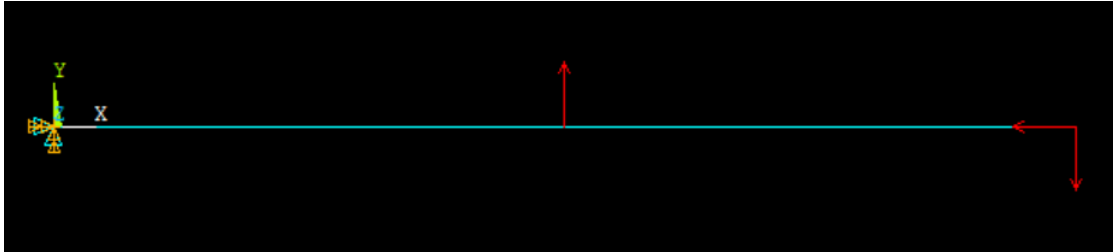
4.2 程序实现

参见 beam.f90

具体实现过程与 TRUSS 单元非常类似，只需要按照理论要求读取单元节点信息并组装刚度阵位移阵进行计算即可。

4.3 验证

我们采用如下算例：长度为 10 的梁，其中在(0,0)出固定所有位移，(5,0)处施加 Y 方向大小为 100N 的外力，在(10,0)处施加 Y 方向大小为-100N 的外力和 X 方向-100N 的外力。



然后分成 2*100 个单元进行计算，比较(10,0)点的位移。弹性模量为 $E=1E-6$ ，截面积为 $1E-6$ 。

STAP90 计算结果：

102 -0.100000E+00 -0.229167E+01 -0.375000E+00

对于解析解，(10,0)点 Y 方向位移为：

$$w = \frac{F\left(\frac{L}{2}\right)^3}{3EI} + \frac{F\left(\frac{L}{2}\right)^2 * \frac{L}{2}}{2EI} - \frac{FL^3}{3EI} = -2.2916667$$

可以看到结果完全吻合，验证了程序的正确性。

5. 板、壳单元（plate&shell）

5.1 理论基础

由于壳单元，可以利用板单元再叠加 4Q 单元的形式得到，因此两种单元在原理上大致相同。而我们组的板壳单元，采用了 Mindlin 板壳理论。即构造了 4 结点 Mindlin 板单元。

该理论基于的假设即为，在变形前后，垂直于中心面的平面仍然保持平面，但并不一定始终垂直于中心面。

而这一理论的主要方法，便是对位移和转动各自独立插值。由此，可以得到系统的总势能可以表示为

$$\bar{\Pi}_p = \Pi_p + \iint_{\Omega} \alpha_1 \left(\frac{\partial w}{\partial y} - \theta_x \right)^2 dx dy + \iint_{\Omega} \alpha_2 \left(\frac{\partial w}{\partial x} + \theta_y \right)^2 dx dy$$

而

$$\Pi_p = \iint_{\Omega} \left(\frac{1}{2} \kappa^T D_b \kappa - q w \right) dx dy - \int_{s_3} \bar{Q}_n w ds + \int_{s_2 + s_3} (M_s \theta_s + M_n \theta_n) ds$$

而板的曲率和扭率为

$$\kappa = \begin{pmatrix} \kappa_x \\ \kappa_y \\ \kappa_z \end{pmatrix} = \begin{pmatrix} \frac{\partial \theta_x}{\partial x} \\ -\frac{\partial \theta_y}{\partial y} \\ -\frac{\partial \theta_x}{\partial x} + \frac{\partial \theta_y}{\partial y} \end{pmatrix} \quad C = \begin{bmatrix} \frac{\partial w}{\partial y} - \theta_x \\ \frac{\partial w}{\partial x} + \theta_y \end{bmatrix}$$

由于还需要考虑由于剪应变而产生的应变，所以厚板的应力应变关系如下所示

$$M = D_b \kappa$$

$$Q = D_s C$$

其中

$$D_b = \frac{Et^3}{12(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad D_s = \frac{Et}{2(1+\nu)\alpha} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

这里的 E 为杨氏模量， ν 为泊松比， t 为板厚， $\alpha = \frac{Gt}{2k}$ 。由剪切应变能等效的原则， k 应

取 $\frac{5}{6}$ 。

5.2 程序实现

参见 plate.f90 以及 shell.f90

与之前的单元大致相同,通过给定结点的坐标以及自由度的信息来,构造单元的刚度阵。需要注意的是,壳单元的刚度阵由三个部分组成,即平面拉伸的 $4Q$ 单元分量,以及转动单元的弯曲刚度分量 K_b , 以及剪切应变分量 K_s

又因为我们的为四结点单元,对位移及转角的插值仍可以使用 4 结点的线性插值函数从而得到

$$\begin{bmatrix} w \\ \theta_x \\ \theta_y \end{bmatrix} = [N_1 I \quad N_2 I \quad N_3 I \quad N_4 I] \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}, \text{其中 } u_i = \begin{bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \end{bmatrix}$$

再带入之前 κ 与 C 的表达式

$$\begin{aligned} \kappa^e &= [B_{b1} \ B_{b2} \ B_{b3} \ B_{b4}] u^e \\ C^e &= [B_{s1} \ B_{s2} \ B_{s3} \ B_{s4}] u^e \\ B_{bi} &= \begin{bmatrix} 0 & 0 & \frac{\partial N_i}{\partial x} \\ 0 & -\frac{\partial N_i}{\partial y} & 0 \\ 0 & -\frac{\partial N_i}{\partial x} & \frac{\partial N_i}{\partial y} \end{bmatrix} \quad B_{si} = \begin{bmatrix} -N_i & 0 & \frac{\partial N_i}{\partial x} \\ 0 & -N_i & \frac{\partial N_i}{\partial y} \end{bmatrix} \end{aligned}$$

将以上的式子带入之前的势能泛函,再取变分为 0,可以得到

$$\begin{aligned} Ku &= (K_b + \alpha K_s) u = P \\ K_b &= \sum_e K_b^e \quad K_s = \sum_e K_s^e \quad P = \sum_e P^e \\ K_b^e &= \iint_{\Omega_e} B_b^T D_b B_b dx dy \quad K_s^e = \iint_{\Omega_e} B_s^T D_s B_s dx dy \\ P^e &= \iint_{\Omega_e} N^T \begin{pmatrix} q \\ 0 \\ 0 \end{pmatrix} dx dy + \int_{S_{2e}+S_{3e}} N^T \begin{pmatrix} 0 \\ M_n \\ M_n \end{pmatrix} ds + \int_{S_{3e}} N^T \begin{pmatrix} 0 \\ 0 \\ Q_n \end{pmatrix} ds \end{aligned}$$

再通过求解 $Ku = P$ 便可以得到节点位移的坐标。

在实现的过程中,有一个需要注意的问题

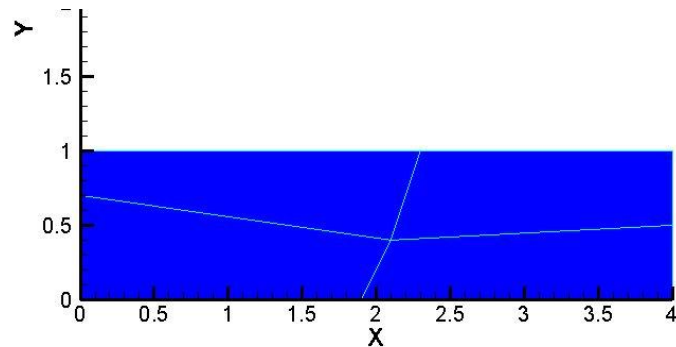
正如 Timoshenko 梁单元的情况一样,在板很薄即 ($t/l \ll 1$) 的情况下会发生剪切自锁,必须保证有限元求解方程的刚度矩阵中与罚函数相关的部分的 K_s 的奇异性,为保证 K_s 的奇异性,需要对单元的刚度阵采用选择性减缩积分,从而使问题的解答中包含了除刚体运动以外的且对变形能无贡献的变形模式,即零能模式。

在该单元中,对 K_b 矩阵采用 2×2 的高斯积分,而对 K_s 矩阵则采用 1×1 的高斯积分。

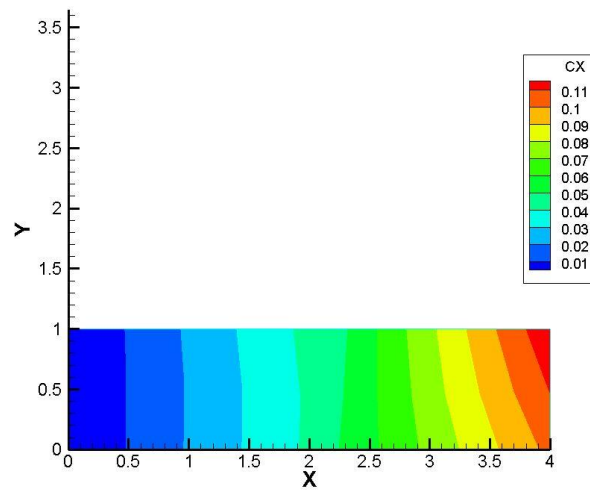
5.3 验证

5.3.1 分片试验

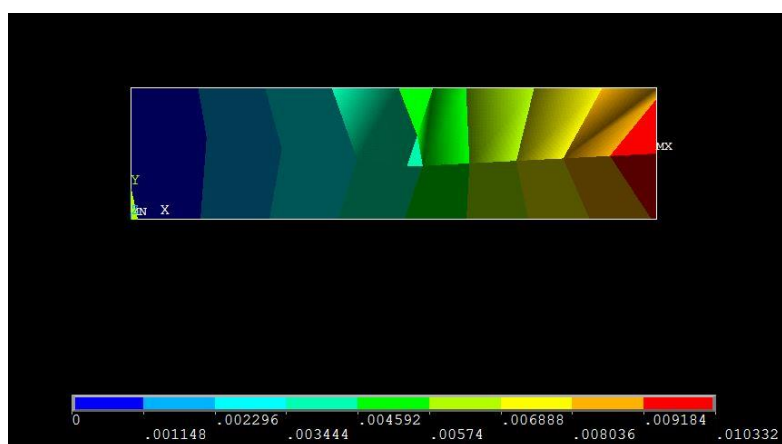
查阅文献可知，4 结点板壳单元不能通过分片试验，故在此仅利用一个算例说明一下。划分一个不规则的网格如图所示



左端固支，在右端加一个大小为 10 的 x 方向的弯矩，可以得到 x 方向转角的分布如图所示



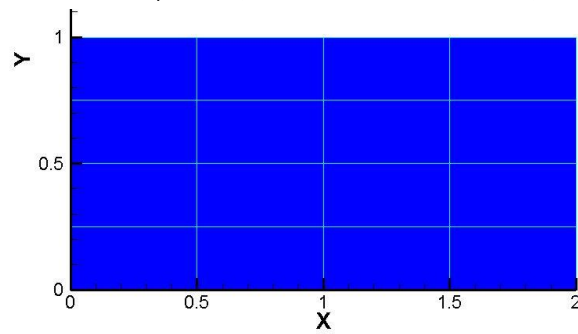
而根据 ANSYS 算出的结果



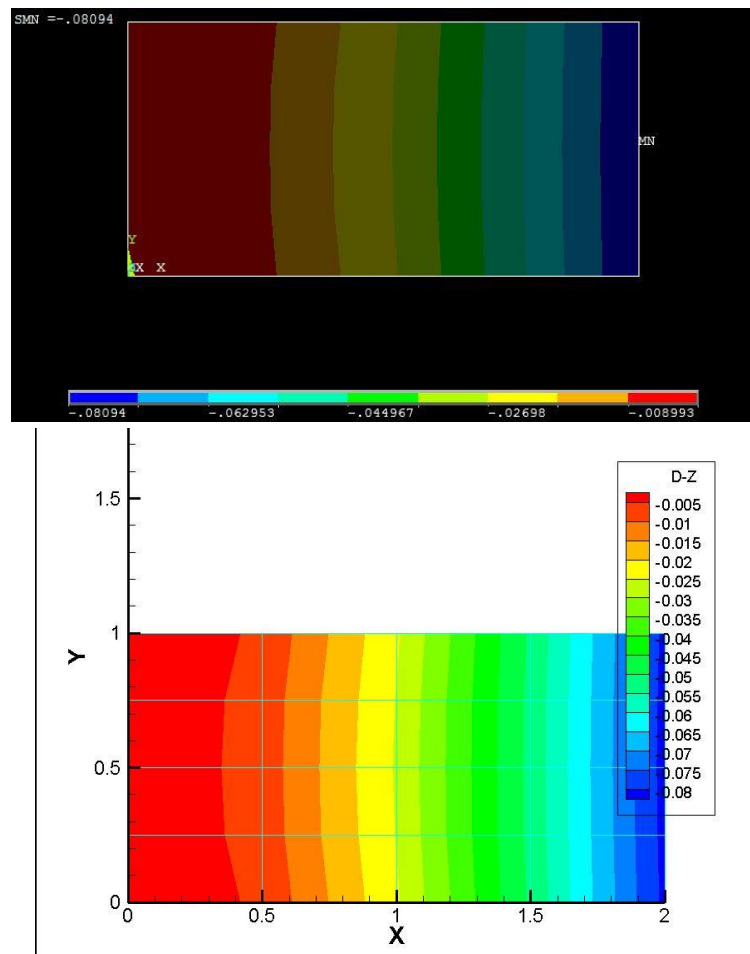
可以发现，stap90 的转角结果与 ANSYS 还是大致相同

5.3.2 收敛性分析

当网格划分如图所示时 4×4 , 在右端加上沿线均匀分布的载荷时



将 ANSYS 结果与 shell 模块算得的结果比较



二者结果相同。

不妨考察单元中特点的节点，再加密网格进行比较

于是我们考察 $(2, 0.5)$ 这一点的挠度

	4X4	8X8	16X16
Ansys	-0.80940E-01	-0.82089E-01	-0.82422E-01
Stap90	-0.818358E-01	-0.830173E-01	-0.823216E-01
error	1.107%	1.131%	0.122%

结果与 ANSYS 吻合的相当好，并且随着网格数的增大，也与 ANSYS 一起逐渐收敛。

(二) 高级单元

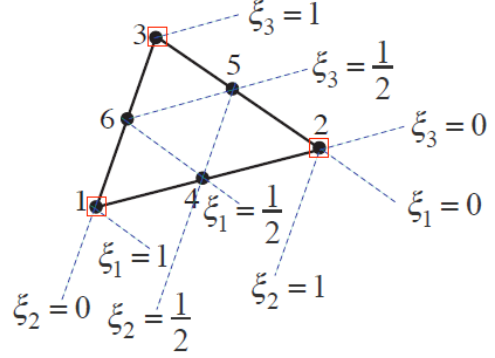
1. 六节点三角形亚单元

为了让程序功能更强大，我们在原有 3T 单元的基础上添加了 6 节点的三角形亚单元。

1.1 理论基础

为建立三角形亚单元，我们需要求出形函数矩阵以及对应的应变矩阵。考虑如右图所示 6 节点三角形单元，由图可以利用巧凑边点元的方法得到各节点处的形函数如下：

$$\begin{aligned} N_1^{6T} &= \xi_1(2\xi_1 - 1) & N_4^{6T} &= 4\xi_1\xi_2 \\ N_2^{6T} &= \xi_2(2\xi_2 - 1) & N_5^{6T} &= 4\xi_2\xi_3 \\ N_3^{6T} &= \xi_3(2\xi_3 - 1) & N_6^{6T} &= 4\xi_1\xi_3 \end{aligned}$$



由于是亚单元，所以我们有

$$x = \sum_{I=1}^3 x_I \xi_I, \quad y = \sum_{I=1}^3 y_I \xi_I \quad \theta^e(x, y) = N^{6T}(\xi_1, \xi_2, \xi_3) d^e$$

于是可以得到如下所示方程

$$\begin{bmatrix} 1 & 1 & 1 \\ x_1^e & x_2^e & x_3^e \\ y_1^e & y_2^e & y_3^e \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}$$

于是可以得到

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1^e & x_2^e & x_3^e \\ y_1^e & y_2^e & y_3^e \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \frac{1}{\det J} \begin{bmatrix} x_2^e y_3^e - x_3^e y_2^e + (y_2^e - y_3^e)x + (x_3^e - x_2^e)y \\ x_3^e y_1^e - x_1^e y_3^e + (y_3^e - y_1^e)x + (x_1^e - x_3^e)y \\ x_1^e y_2^e - x_2^e y_1^e + (y_1^e - y_2^e)x + (x_2^e - x_1^e)y \end{bmatrix}$$

其中

$$\det J = (x_2^e y_3^e - x_3^e y_2^e) - (x_1^e y_3^e - x_3^e y_1^e) + (x_1^e y_2^e - x_2^e y_1^e)$$

于是有

$$\frac{\partial}{\partial x} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} = \frac{1}{\det J} \begin{bmatrix} y_{23}^e \\ y_{31}^e \\ y_{12}^e \end{bmatrix} \quad \frac{\partial}{\partial y} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} = -\frac{1}{\det J} \begin{bmatrix} x_{23}^e \\ x_{31}^e \\ x_{12}^e \end{bmatrix}$$

又

$$\frac{\partial N_I^{6T}}{\partial x} = \begin{bmatrix} \frac{\partial N_I^{6T}}{\partial \xi_1} & \frac{\partial N_I^{6T}}{\partial \xi_2} & \frac{\partial N_I^{6T}}{\partial \xi_3} \end{bmatrix} \begin{bmatrix} \frac{\partial \xi_1}{\partial x} \\ \frac{\partial \xi_2}{\partial x} \\ \frac{\partial \xi_3}{\partial x} \end{bmatrix} = \frac{1}{\det J} \begin{bmatrix} \frac{\partial N_I^{6T}}{\partial \xi_1} & \frac{\partial N_I^{6T}}{\partial \xi_2} & \frac{\partial N_I^{6T}}{\partial \xi_3} \end{bmatrix} \begin{bmatrix} y_{23}^e \\ y_{31}^e \\ y_{12}^e \end{bmatrix}$$

$$\frac{\partial N_I^{6T}}{\partial y} = \begin{bmatrix} \frac{\partial N_I^{6T}}{\partial \xi_1} & \frac{\partial N_I^{6T}}{\partial \xi_2} & \frac{\partial N_I^{6T}}{\partial \xi_3} \end{bmatrix} \begin{bmatrix} \frac{\partial \xi_1}{\partial y} \\ \frac{\partial \xi_2}{\partial y} \\ \frac{\partial \xi_3}{\partial y} \end{bmatrix} = -\frac{1}{\det J} \begin{bmatrix} \frac{\partial N_I^{6T}}{\partial \xi_1} & \frac{\partial N_I^{6T}}{\partial \xi_2} & \frac{\partial N_I^{6T}}{\partial \xi_3} \end{bmatrix} \begin{bmatrix} x_{23}^e \\ x_{31}^e \\ x_{12}^e \end{bmatrix}$$

将

$$\begin{aligned} N_1^{6T} &= \xi_1(2\xi_1 - 1) & N_4^{6T} &= 4\xi_1\xi_2 \\ N_2^{6T} &= \xi_2(2\xi_2 - 1) & N_5^{6T} &= 4\xi_2\xi_3 \\ N_3^{6T} &= \xi_3(2\xi_3 - 1) & N_6^{6T} &= 4\xi_1\xi_3 \end{aligned}$$

带入上式，即可以得到单元的应变矩阵

$$B_I^e = \begin{bmatrix} N_{I,x}^{6T} & 0 \\ 0 & N_{I,y}^{6T} \\ N_{I,y}^{6T} & N_{I,x}^{6T} \\ \frac{N_I^{6T}}{r} & 0 \end{bmatrix}$$

其中

$$r = \xi_1 x_1^e + \xi_2 x_2^e + \xi_3 x_3^e$$

1.2 程序实现

参见 trielm.f90

在了解了原理之后，我们主要基于原有的 3T 程序 trielmt.f90 来进行修改。我们主要修改了 BNmat 函数，改为 BNmat6 函数，在里面计算了 N 矩阵和 B 矩阵。

值得注意的是，我们算出的 N 矩阵和 B 矩阵有 12 列，然而其实对比 3T 我们发现，6T 多出来的 3 个点的位移我们可以不用关心，因此我们利用 N 矩阵和 B 矩阵算出来你的单元刚度阵 S 我们在组装的时候后只需组装前 6 行前 6 列即可。因此我们在 ADDBAN 函数里传入的是 S(1:6)。因此我们可以用 3T 的算例来验证 6T 的结果。

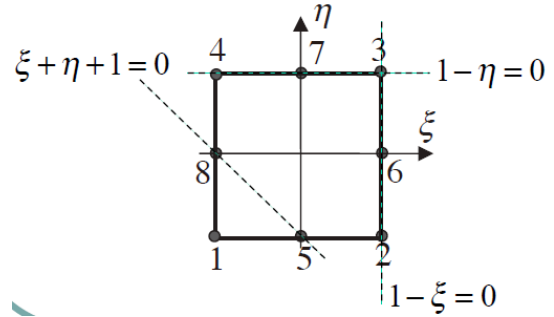
1.3 验证

这里的所有验证都和 3T 的验证相同，因此可以见 3T 的验证。同时结果方面和 3T 的结果也相同，在这里我们不再赘述

2. 八节点四边形亚参元

2.1 理论基础

为建立 8 节点四边形亚参元，我们需要求出形函数矩阵以及对应的应变矩阵。考虑如右图所示 8 节点四边形单元，由图可以利用巧凑边点元的方法得到各节点处的形函数如下：



$$N_1^{8Q} = -\frac{1}{4}(1-\xi)(1-\eta)(\xi+\eta+1)$$

$$N_2^{8Q} = \frac{1}{4}(1+\xi)(1-\eta)(\xi-\eta-1)$$

$$N_3^{8Q} = \frac{1}{4}(1+\xi)(1+\eta)(\xi+\eta-1)$$

$$N_4^{8Q} = -\frac{1}{4}(1-\xi)(1+\eta)(\xi-\eta+1)$$

$$N_5^{8Q} = \frac{1}{2}(1+\xi)(1-\xi)(1-\eta)$$

$$N_6^{8Q} = \frac{1}{2}(1+\eta)(1-\eta)(1+\xi)$$

$$N_7^{8Q} = \frac{1}{2}(1+\xi)(1-\xi)(1+\eta)$$

$$N_8^{8Q} = \frac{1}{2}(1+\eta)(1-\eta)(1-\xi)$$

而利用形函数矩阵，求导得到应力矩阵 **B** 以及组装刚度矩阵的过程与 4Q 单元一致，此处就不再赘述。

而不同之处在于是亚参元，所以对位移相关参数应有关系：

$$x = \sum_{I=1}^4 x_I \xi_I, \quad y = \sum_{I=1}^4 y_I \xi_I, \quad \theta^e(x, y) = N^{4Q}(\xi, \eta) d^e$$

即相关参数仅使用 4 节点单元的形函数插值得到，在计算矩阵 **B** 以及单元刚度阵时应加以注意。

2.2 程序实现

在了解了原理之后，我们主要基于原有的 4Q 程序 `quadrElem.f90` 来进行修改。我们主要修改了计算 **B** 和 **N** 的函数，具体实现程序见 `quad8.f90`

值得注意的是，和 6T 不同，这里的 8Q 没有采取 6T 那种只求端点处位移的方法，这里是把所有节点的位移都读入了，因此输入文件和 4Q 有区别，要读入所有点的坐标，而最后算出来的位移是所有节点的位移。因此我们的算例在 4Q 单元的基础上加以改动，新的算例为 `quad8-4e.in` 用于计算相同形状下 4 个 8 节点亚参元的位移和应力应变解。

(三) 其他功能

1. 模态的计算

1.1 理论和程序实现

为了让程序功能更强大，我们添加了计算模态的功能。为了计算模态，首要的任务是计算质量矩阵。对于弹性力学问题，我们可以得知

$$M^e = \int_V \rho N^T N dV$$

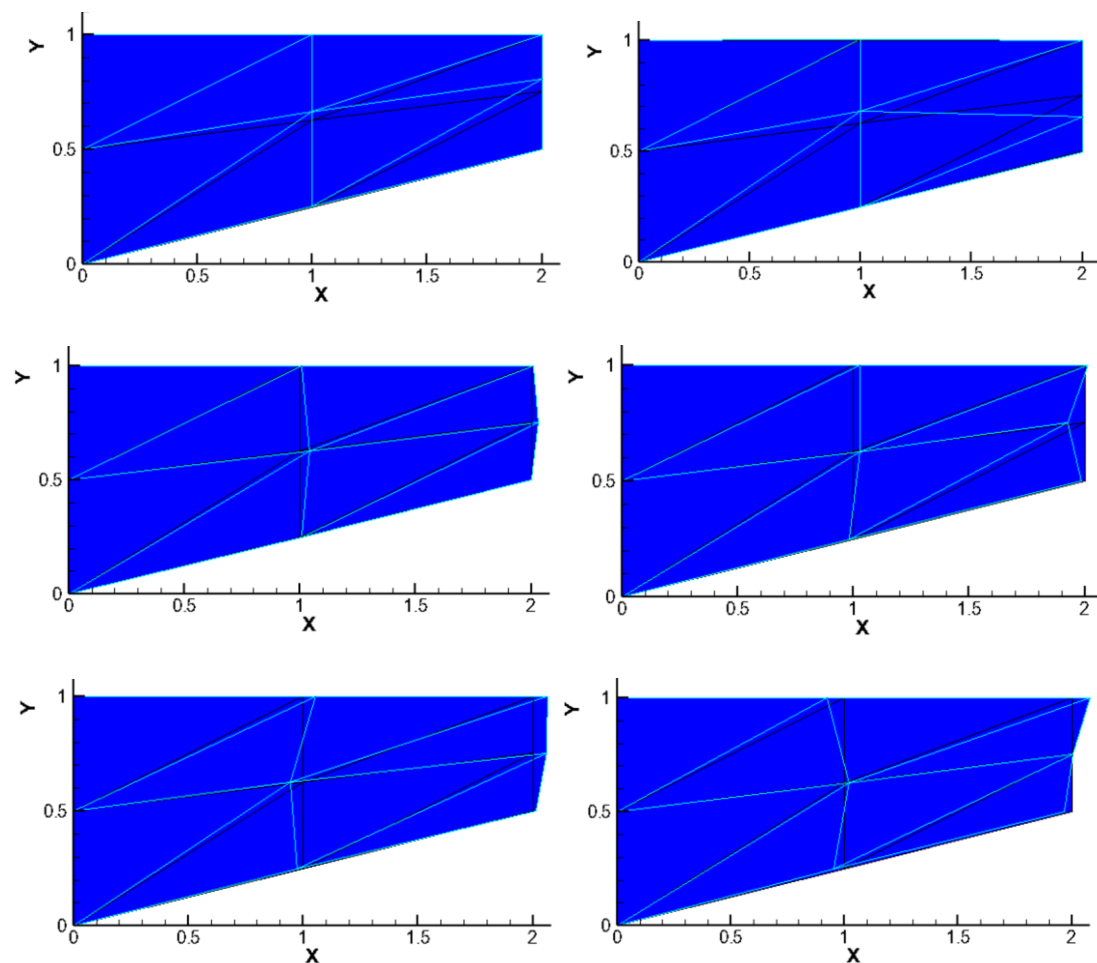
于是，我们只要求出了单元形函数矩阵，就可以得到质量矩阵。

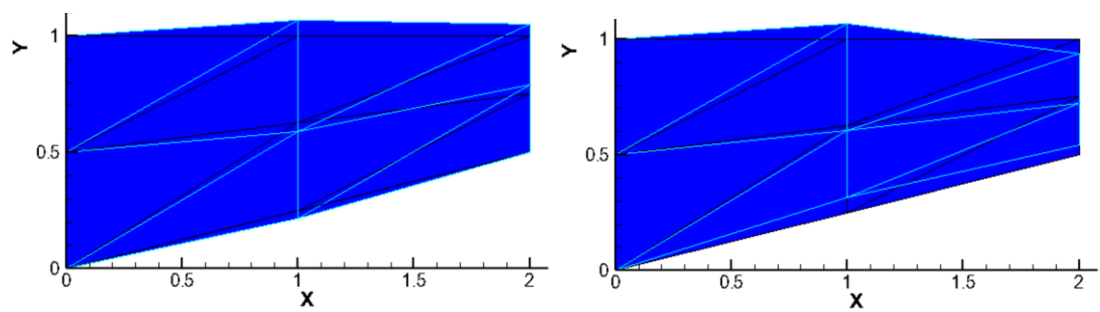
因此我们对于 3T, 4Q, 8H 单元都加上了计算质量矩阵的部分。在 `stap.f90` 中我们为质量矩阵分配了一个序号为 13 的数组。同时我们在 `elcal.f90` 中添加了 EIGEN 子程序，用于计算模态和特征值。

在这个 EIGEN 子程序中。我们主要利用了计算动力学网站上的 SSPACE90 和 LANCZOS 代码。我们用了这两个子程序来计算模态。我们在输入文件的 NPAR 数组中也给了用户一些选择。NPAR(5)用于选择方法，如果 NPAR(5)==1，即选择子空间迭代法，如果 NPAR(6)==2，即选择 Lanczos 法。而 NPAR(6)用于告诉程序计算模态的阶数。

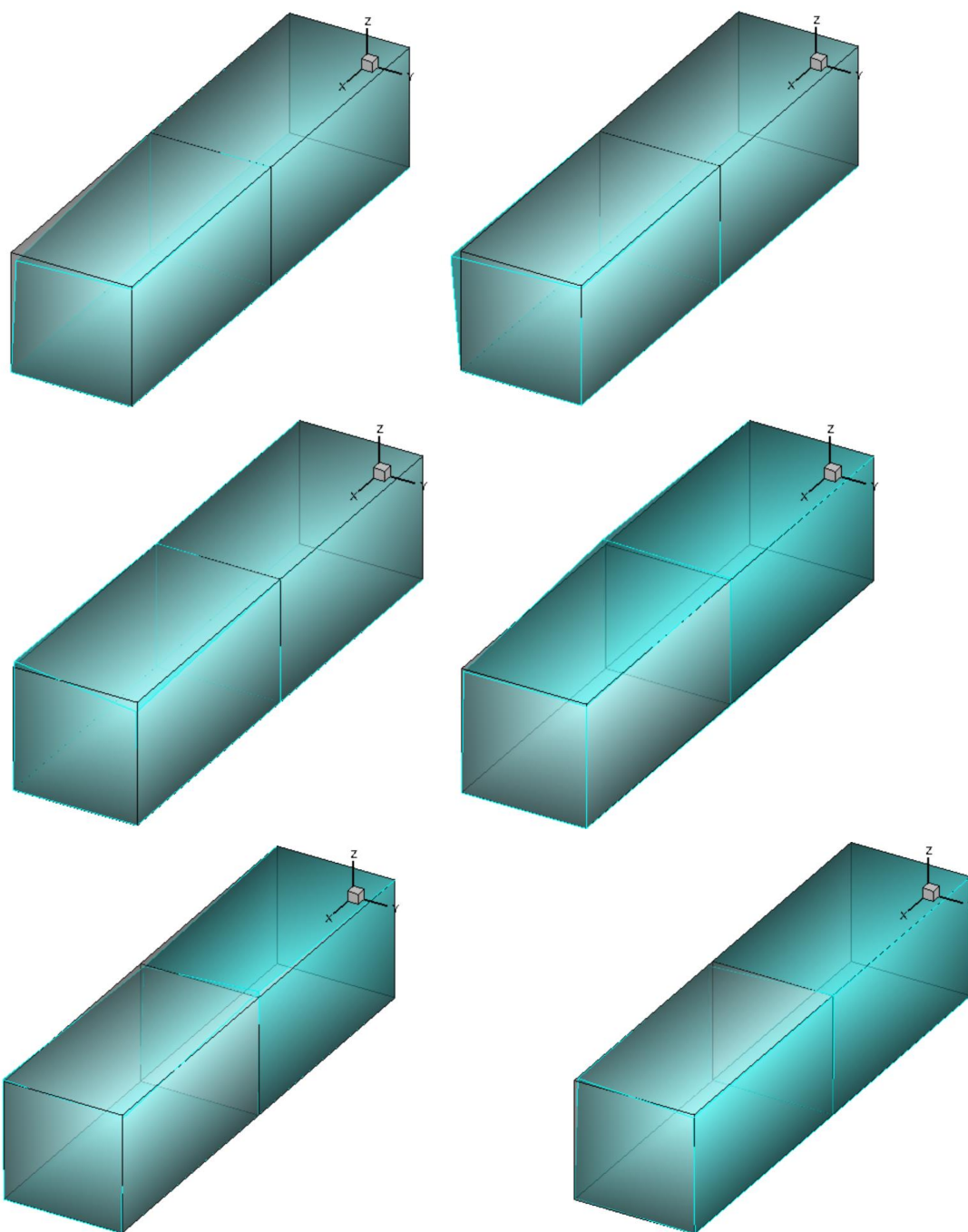
1.2 结果展示

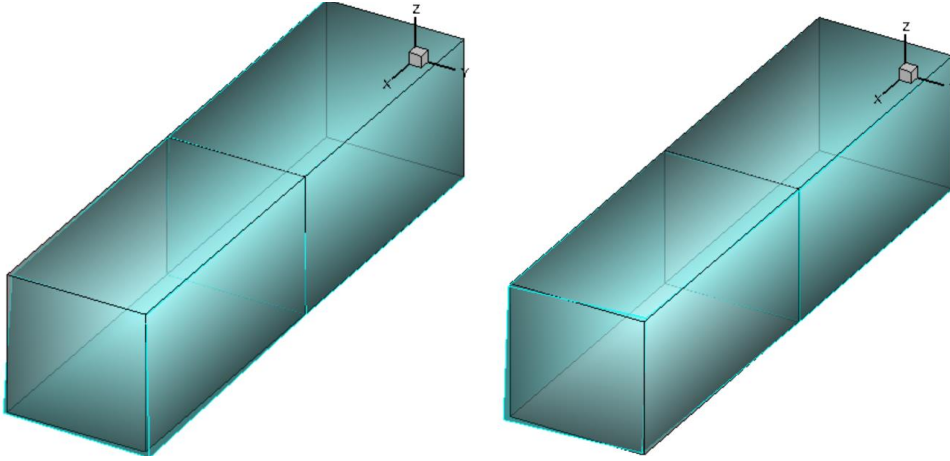
下面是 3T 单元的模态分析，我们利用输入文件是 `trias8.in`，下列分别是 1 到 8 阶模态：





下面是 8H 单元的模态情况，输入文件是 hex2.in，以下分别是 1 到 8 阶模态：





2. 超收敛分片应力恢复 (SPR)

参考 spr.f90

为了利用单元内部存在最佳应力点应力具有高一阶精度的特性, 我们可以利用 SPR 的方法, 又称分片应力磨平。

假设分片内应力的改进解

$$\sigma^* = Pa$$

其中, P 包含适当的多项式的项次, a 是一组待定的参数

对二维问题而言, 线性单元可用

$$P = [1 \quad x \quad y]$$

例如四结点单元, 三节点单元

二次单元可用

$$P = \begin{bmatrix} 1 & x & y & x^2 & xy & y^2 \end{bmatrix},$$

如八节点四边形单元, 六节点三角形单元

分片磨平应力改进解 σ^* , 和有限元计算得到的应力结果 $\tilde{\sigma}$ 按最小二乘法原理构造泛函

$$A(\tilde{\sigma}, \sigma^*) = \sum_{e=1}^m \int_{V_e} (\tilde{\sigma} - \sigma^*)^2 dV = \sum_{e=1}^m \int_{V_e} (\tilde{\sigma} - Pa)^2 dV$$

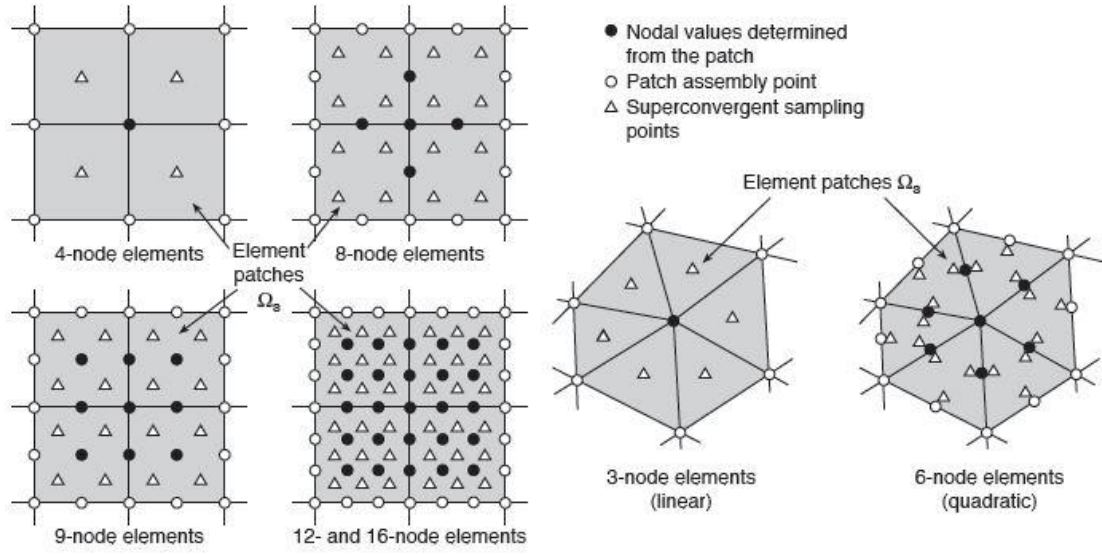
对于未知参数求导, 求泛函极值可得

$$\sum_{e=1}^m \int_{V_e} P^T P dV \cdot a = \sum_{e=1}^m \int_{V_e} P^T \tilde{\sigma} dV$$

或表示

$$Ca = d$$

为了得到某个结点的应力, 可以利用该结点周围的单元的高阶精度应力点, 来构造解 $\tilde{\sigma}$



由此再行求解方程

$$a = C^{-1}d$$

其中

$$C = \sum_{i=1}^n P^T(x_i, y_i) P(x_i, y_i) w_i$$

$$d = \sum_{i=1}^n P^T(x_i, y_i) \tilde{\sigma}(x_i, y_i) w_i$$

求得 a 之后，再带入结点的 $P(x_n, y_n)$ ，便可以得到结点的应力 σ^*

在 4Q 单元中，采用 SPR，在后处理得到的 tec.dat 文件中，可以勾选应力选项，来观察经 SPR 处理后得到的应力结果。

（四） 前处理

1. 简介

在计算仅由几个或几十个单元构成的有限元问题的时候，我们还可以手动输入各单元信息以及载荷信息等参数，但是如果是分成了几百个甚至几千个单元的有限元问题，手动输入无疑是很难实现的，这时候我们可以借助其他软件进行单元划分，同时将划分结果以某种形式输出为 STAP90 能够处理的文件格式，这个过程称为前处理。在本次程序实现中，我们采用 ANSYS 进行前处理过程。

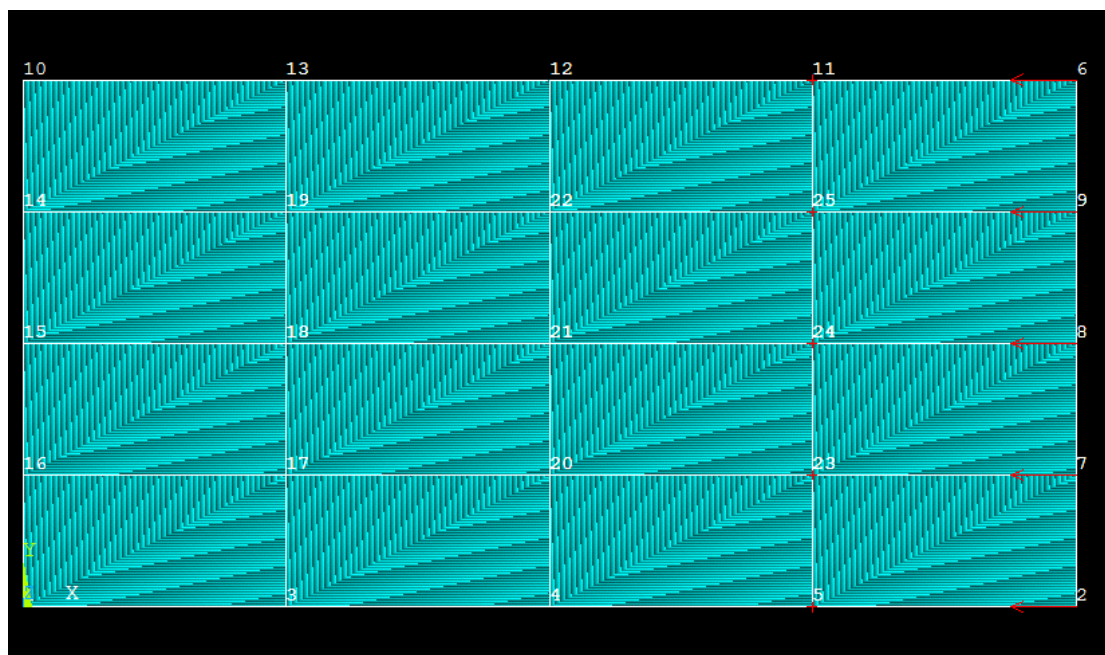
2. 实现过程

在通过 ANSYS 建模并 mesh 之后，通过 CDWRITE,ALL,ANS,IN 指令可以将模型信息全部输出至 ANS.IN 文件中，然后通过 STAP90 中的 PREPROCESSOR.F90 文件读取文件内容并储存关键信息，然后输出至 STAP90.IN 文件进行接下来的计算分析。

3. 具体示例

我们采用一个 SHELL 算例进行演示。

ANSYS 中模型如下，为一个厚度为 0.1，长为 2，宽为 1 的壳模型，划分为十六个单元，



左端固定所有自由度，右端 6、9、8、7、2 节点施加 X 方向-100 的外力，11、25、24、23、5 节点施加 Z 方向 100 的外力，之后用 CDWRITE,ALL,ANS,IN 输出文件 ANS.IN。

3.1 读入节点数和单元数

ANS.IN 文件中存在如下格式内容：

```
*ELSE                                !offset database for the following FE model
NUMOFF,NODE,                          25
```

```

NUMOFF,ELEM,      16
NUMOFF,MAT ,      1
NUMOFF,SECN,      1
NUMOFF,TYPE,      1

```

通过字符串判断出*ELSE，之后读入节点数 NUMP 和单元数 NUME。

3.2 读入单元种类

ANS.IN 文件中存在如下格式内容：

```

DOF,DELETE
ET,      1,181
NBLOCK,6,SOLID,      25,      25

```

通过字符串判断出 DOF，之后读入单元种类 ELETY，下表为不同单元种类对应 ELETY 数值。其中 4Q/3T 均为 PLANE182，shell 和 plate 均为 shell181。

ELETY	LINK180	PLANE182	SOLID185	BEAM188	SHELL181
单元类型	TRUSS	4Q/3T	8H	BEAM	PLATE/SHELL

3.3 读入节点的全局编号

ANS.IN 文件中存在如下格式内容：

```

(3i9,6e20.13)
      1      11      0 0.00000000000000E+00
      2      21      0 2.00000000000000E+00
      3      12     25000 5.00000000000000E-01
      4      12     50000 1.00000000000000E+00
      5      12     75000 1.50000000000000E+00
      6      31      0 2.00000000000000E+00 1.00000000000000E+00
      7      22     25000 2.00000000000000E+00 2.50000000000000E-01
      8      22     50000 2.00000000000000E+00 5.00000000000000E-01
      .....

```

通过字符串判断出 3i9,6，之后读入各节点坐标，用数组 POS(3,NUMP)储存。

3.4 读入节点的全局编号

ANS.IN 文件中存在如下格式内容：

```

EBLOCK,19,SOLID,      16,      16
(19i9)
      1      1      1      1      0      0      13
0      4      0      1      1      3      17      16
      1      1      1      1      0      0      13
0      4      0      2      3      4      20      17
      1      1      1      1      0      0      13

```

```

0      4      0      3      4      5      23      20
      1      1      1      1      0      0      13
0      4      0      4      5      2      7      23
      1      1      1      1      0      0      13
0      4      0      5      16     17     18     15

```

.....

通过字符串判断出 (19i9)，之后读入每个单元对应的节点编号，用 IEN(DNS,NUME) 储存。下表为不同单元对应的 DNS 值：

ELETY	LINK180	PLANE182	SOLID185	BEAM188	SHELL181
DNS	2	4	8	2	4

3.5 读入集中载荷

ANS.IN 文件中存在如下格式内容：

ERESX,DEFA

```

D,      1,UX  ,  0.00000000  ,  0.00000000
D,      1,UY  ,  0.00000000  ,  0.00000000
D,      1,UZ  ,  0.00000000  ,  0.00000000
D,      1,ROTX,  0.00000000  ,  0.00000000
D,      1,ROTY,  0.00000000  ,  0.00000000
D,      1,ROTZ,  0.00000000  ,  0.00000000
D,     10,UX  ,  0.00000000  ,  0.00000000
D,     10,UY  ,  0.00000000  ,  0.00000000
D,     10,UZ  ,  0.00000000  ,  0.00000000
D,     10,ROTX,  0.00000000  ,  0.00000000

```

.....

```

F,      2,FX  , -100.000000  ,  0.00000000
F,      5,FZ  ,  100.000000  ,  0.00000000
F,      6,FX  , -100.000000  ,  0.00000000
F,      7,FX  , -100.000000  ,  0.00000000
F,      8,FX  , -100.000000  ,  0.00000000
F,      9,FX  , -100.000000  ,  0.00000000

```

.....

通过字符串判断出 ERESX，之后读入每个集中载荷的信息，D 代表位移约束，F 代表外力约束，UX/UY/UZ 代表 X/Y/Z 三个方向的位移约束，ROTX/ROTY/ROTZ 代表三个方向的转角约束。

3.6 STAP.IN 文件输出

将以上读取的信息按照 STAP90 处理的文件格式输出。

（五） 后处理

1、简介

对于计算后得到的结果均是以数字方式表示,尤其是复杂问题的解非常的繁琐,通过 TECPLOT 可以对数据进行可视化处理,生成云图,使结果一目了然,省却了逐项分析比较的许多麻烦。

2、实现过程

通过 POSTPROCESSOR 子程序实现了对各单元的后处理,生成统一格式的可被 TECPLOT 读入的数据文档 TEC.DAT,可用 TECPLOT 画出各单元各方向位移云图,可以直观的观察各部分变形大小分布。此外,因 4Q 单元进行了 SPR 处理,因此可以画出应力云图,各点应力也可以直观了解。PLATE 和 SHELL 单元因 TECPLOT 中无对应单元,所以采用 4Q 单元进行近似构造,可画各方向位移云图。

POSTEIGEN 子程序可以对 3T、4Q、8H 计算得到的模态进行后处理,每个 ZONE 对应的编号 N 即为第 N 阶模态。

3、具体示例

见各单元中示例,在此不在赘述。

三、分工与合作

在本次作业中，大家充分发挥了合作的精神，每个人各尽其能，共同完成这个系统性的 STAP90 程序以及前后处理的相关工作。具体任务分工如下：

- 黄维啸（组长）：3T、6T、计算动力学模块
- 潘哲鑫：4Q、SHELL、PLATE、SPR。
- 祝凌霄：BEAM、ANS 前处理及部分 ANS 建模。
- 舒炫博：8H、8Q 亚参元
- 袁博：TECPLOT 后处理。

在本次作业中，我们小组保证了每周 1~2 次的小组讨论以确保每名成员的工作进程，并讨论彼此在这一周内得到的收获以及遇到的问题，很好的保证了组内同学的效率。同时，我们小组坚持在 SVN 源中修改文件以及上传相关资料，一开始感觉很不习惯，但是很快的我们发现了 SVN 的便利之处，SVN 可以查看每名同学的上传内容以及版本，有几次我们出现了自己修改的内容出现 BUG，我们则可以从 SVN 中之间调取之前的版本。

四、总结与反思

这次大作业主要是对 STAP90 程序的扩充与完善，既是对编程能力的一种考验，也是对之前学过的有限元知识的一种考察。通过这次大作业，我们对有限元方法的理解更深了一层，认识到要写一个程序远不仅仅是将结果算出来就可以，也要通过各种手段来证明解的正确性（分片试验与收敛性分析等）。

在单元的添加中，我们也认识到了分工与合作的重要性。事实上，各种单元的添加如果了解了单元的性质和原理之后，还是相对比较容易的。不过比较难做的是后面各种功能的添加，例如高级单元、SPR 还有计算动力学模块等等，有时候实现起来，比较困难，但是最后大家一起分工合作之后，效率还是很高的。

而对于前处理方面，需要对 ANSYS 比较好的理解。涉及到很多文件读写的操作，也使得大家对于这方面以及 ANSYS 输出文件有了更深入的理解。

后处理主要需要对文档读写格式的运用，文件的读写在平时应用可能不多，使用并不是那么熟练，以后要加强这方面的训练。

除开技术层面，团队合作也是极为重要的一环，只凭一个人是不太可能在这么短的时间里完成一个功能相对全面的大程序的。SVN 为我们提供了合作的便捷渠道，使得团队工作更加高效。每周两次的小组讨论也让我们交流中收获了很多，一起讨论一起 debug，彼此都学习到了很多，也增进了彼此的感情。当看到程序最终正确运行的时候，每个人的心情都很激动。

感谢各位同学的努力，感谢老师的指导，我们在将来会做的更好。