

3T 单元编程报告

管唯宇

2017 年 12 月 8 日

1 组装刚度阵

类似 4Q 单元，进行一次 3d-2d-3d 的转换。

n, i, j 同 4Q 一样生成，但对于 3T 单元，设置 $i = \bar{p}_{21}$ 可以简化计算量。

计算刚度阵有

$$K^e = A^e B^{eT} D B^e$$

计算应力有

$$\sigma = D \varepsilon = D B^e d^e$$

由于 D, B^e, d^e 都是常矩阵，故 3T 单元中为常应力场。

2 分片试验

分片实验数据如/data/3T/patch.dat 所示，如图 2所示。

图中，节点为精确解，实体为计算解。

3T 单元通过分片实验。

3 收敛率计算

__TEST__ 被定义时，会在计算应力时额外输出高斯点的位移和积分的系数。(采用三点高斯积分)。

这个宏可以在 cmake 中通过 STAP++_TEST 选项开启。

根据

$$e^2 = \int_{\Omega} (u - u^e)^2 d\Omega = \sum W_i (u - u^e)^2 |J|$$

就可以计算误差的积分。上式中， $W_i |J|$ 合并为 weight，即

$$e^2 = \sum_{i=1}^{NOE} w_i \sum_{j=1}^3 (u_{ij} - u_{ij}^e)^2$$

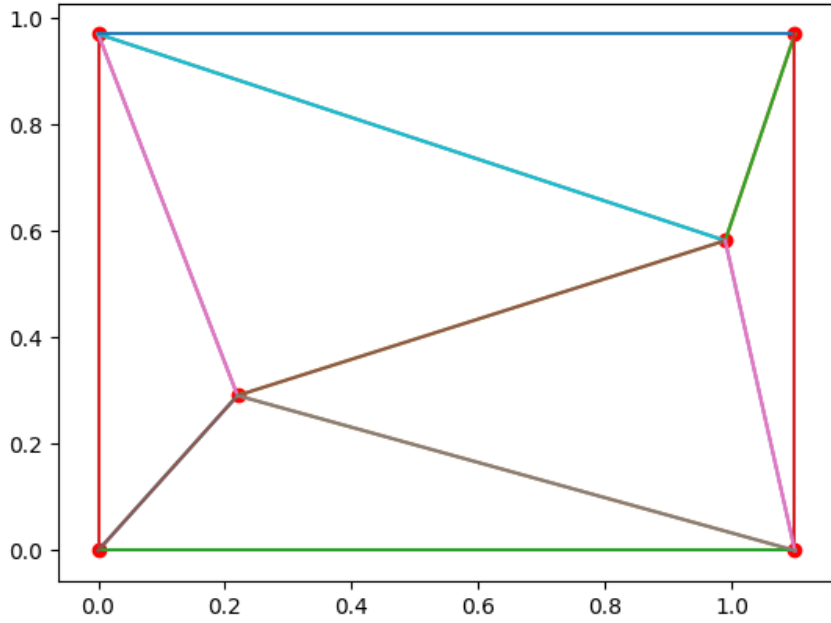


图 1: 分片试验

通过一个简单的 python 文件, 根据高斯点的坐标算出准确位移, 我们就可以计算出总的误差 e 。

考虑如图 2 这样一个简单的情况。体力随 X 线性变化, Y 方向限制自由度。体力 $\mathbf{f} = bx$ 容易计算得到应变满足

$$\begin{cases} \varepsilon_x = \frac{1-\nu^2}{E}b \left(2 - \frac{x^2}{2}\right) \\ \varepsilon_y = 0 \\ \gamma_{xy} = 0 \end{cases}$$

位移满足

$$\begin{cases} u_x = \frac{(1-\nu^2)b}{6E}x(12 - x^2) \\ u_y = 0 \end{cases}$$

生成任务文件的脚本见 `/data/3T/genDat.py`。

处理输出文件的脚本见 `/data/3T/calcErr.py`。

测试文件见 `/data/3T/run-rate.py`

网格划分为每单位长度 $1 \sim 32, 64, 128$ 个单元, 计算结果如图 3 所示。

即, 3T 单元位移二阶收敛。

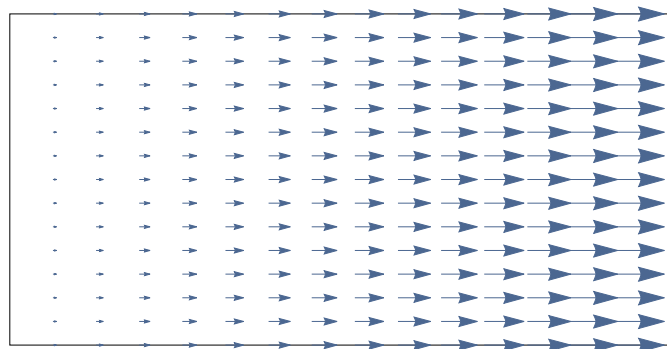


图 2: 施加载荷

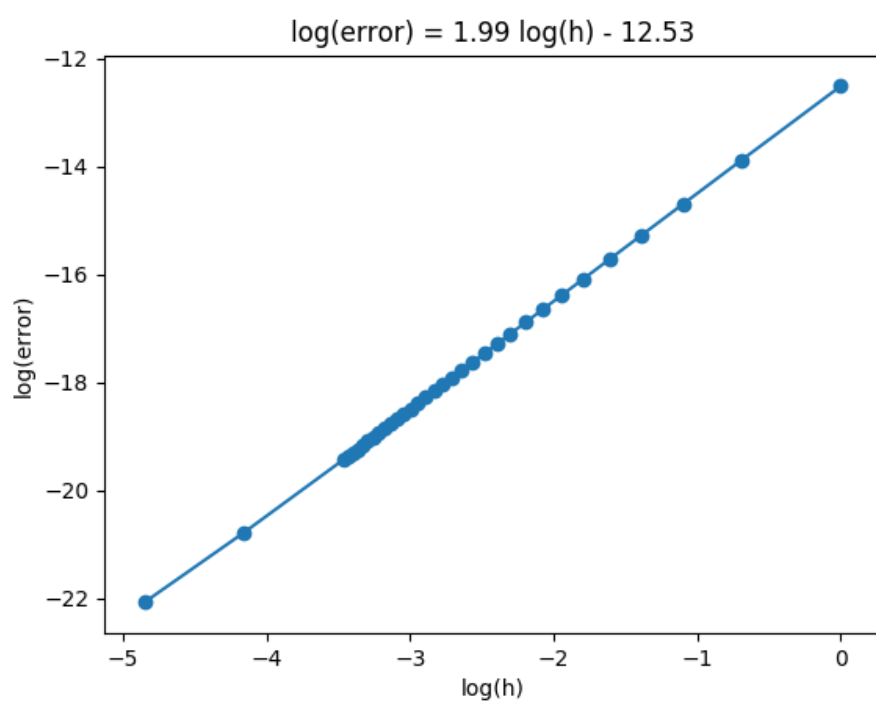


图 3: 误差分析