Nonlinear Finite Element Method (Spring 2024)

Mid-term Project

Student Name: Puhan Zhang
Student ID: 2301213175

一、项目说明

本项目是北京大学 2024 春非线性有限元方法（Nonlinear Finite Element Methods）课程的作业。通过完善 matlab 程序，学习小变形问题下的采用有限元方法对线弹性材料的求解过程。

本案例研究梯形板的平面应力问题，如图 1。梯形板左侧固定在墙面上，顶侧被施加均匀分布的外应力。通过本程序能求解梯形板在此外力作用下，各点的位移及内应力。

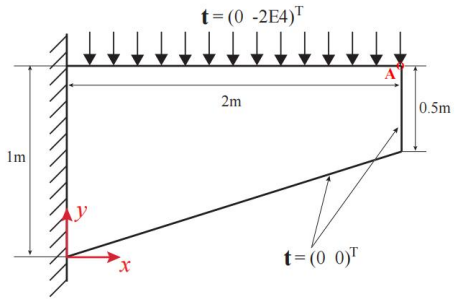


图 1 梯形板的平面应力问题

二、程序结构

(1) main.m

程序流程图如图 2。

首先通过 generate_mesh 编号结点与网格。然后按两条线分别计算 K 矩阵和 F 矢量。

第一条线计算 K 矩阵：先通过 g_center 计算网格的重心坐标和网格面积，然后通过 B_matrix 计算 B 矩阵，再通过 K_matrix 计算 K 矩阵。

第二条线计算 F 矢量：先通过 Boundary_conditions 解析边界条件，再通过 F_vector 计算各结点上的外力即 F 矢量。

通过 Enforce_BC，采用划 0 置 1 法，按照位移边界条件处理 K 矩阵和 F 矢量。然后通过$u = K/F$计算各点应变。然后通过 constitutive 计算各点应力。

为了使结果直观，需要绘图。通过 generate_mesh 生成网格后，使用 triplot 和 quadplot 分别绘制三角形与四边形网格。程序计算出结果后，通过 plot_results 绘制梯形板的形变和应力分布。

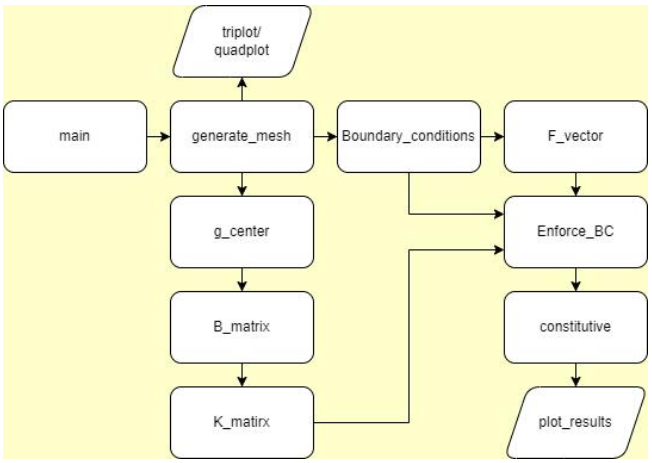


图 2 程序流程图

(2) generate_mesh.m

第一步是给梯形板划分网格。网格分为三角形网格和四边形网格两种。

划分网格，首先在梯形板中划分结点，结点相连形成网格。网格可以被表示为结点的集合。三角形网格包含 3 个结点，四边形网格包含 4 个结点，NNE 边形网格包含 NNE 个结点。

如图 3，结点按行列有规律地排布与编号。结点的编号记作 id_nodes，编号规定如下：编号从 1 开始，以梯形板左上方的点为 1 号结点，按先从左到右再从上到下的顺序编号，最后编号的结点是梯形板右下方的点。结点一共有 j 行 k 列，共有 nodes=j*k 个结点，即 id_nodes 的取值范围为从 1 到 nodes 的正整数。

对于四边形网格。将结点按行列方向连线，共获得 elements=(j-1)*(k-1)个网格。四边形网格的编号方式，遵循先从左到右再从上到下的顺序。对于三角形网格，可以先划分出四边形网格，然后每个四边形用一条对角线将其划分为两个三角形，共获得 elements=2*(j-1)*(k-1)个网格。三角形网格的编号方式和四边形网格类似，也遵循先从左到右再从上到下的顺序。

函数 generate_mesh 的作用是计算各结点坐标，计算网格与结点的关系。会生成两个数组 x_a 和 elem。x_a 是[nodes×dim]的数组，存储所有 nodes 个结点每个结点的坐标。dim 是坐标的维度，本案例是平面问题所以 dim=2（空间问题 dim=3）。elem 是[elements×NNE]的数组，存储所有 elements 个网格每个网格包含的结点的编号。三角形网格包含 3 个结点所以 NNE=3，四边形网格包含 4 个结点所以 NNE=4。

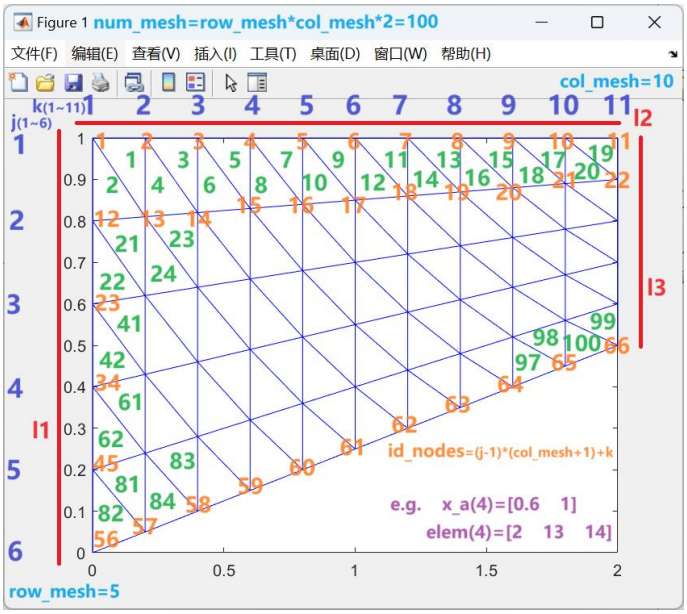


图 3 结点与网格的编号方式

(3) g_center.m

函数 g_center 的作用是计算每个网格的重心坐标和网格面积。会生成两个数组 xg 和 Area。xg 是[elements×dim]的数组，存储所有 elements 个网格每个网格的重心坐标。Area 是[elements×1]的数组，存储所有 elements 个网格每个网格的面积。

多边形的重心坐标，就是各顶点坐标的均值。

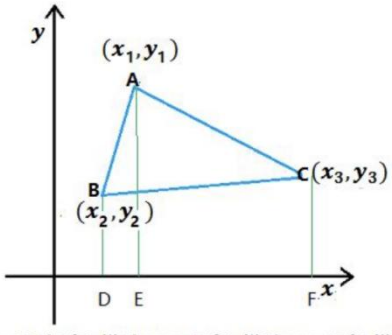
已知三角形三个顶点的坐标，就能计算三角形的面积，如图 4。

若三角形 3 个顶点坐标为：$(x_1, y_1), (x_2, y_2), (x_3, y_3)$，

则三角形的面积为：$\text{Area} = \left|\frac{1}{2}(x_1y_2 + x_2y_3 + x_3y_1 - x_1y_3 - x_3y_2 - x_2y_1)\right|$

四边形的面积可以拆分成两个三角形相加，因此若四边形 4 个顶点坐标为：$(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$，

则四边形的面积为：$\text{Area} = \left|\frac{1}{2}(x_1y_2 + x_2y_3 + x_3y_1 - x_1y_3 - x_3y_2 - x_2y_1)\right| + \left|\frac{1}{2}(x_1y_4 + x_4y_3 + x_3y_1 - x_1y_3 - x_3y_4 - x_4y_1)\right|$



三角形面积=梯形DECA 面积+梯形BFEC 面积-梯形DFBA面积

$=(y_1+y_2)(x_1-x_2)/2 + (y_3+y_2)(x_3-x_2)/2 - (y_3+y_1)(x_3-x_1)/2$

$$A = \frac{1}{2}\left[x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)\right]$$

图 4 三角形顶点坐标计算三角形面积

(4) Boundary_conditions.m

函数 B_conditions 负责解析边界条件。

边界条件分为位移边界条件和应力边界条件，分别存储在 A 矩阵和 B 矩阵中。

A 矩阵是[a×4]的，表示有 a 个位移边界条件。B 矩阵是[b×2]的，表示有 b 个应力边界条件。本案例中，两种边界条件各只有一个，因此 a=1 且 b=1。

函数 displa 负责解析位移边界条件即 A 矩阵，返回两个均为[dim*nodes×1]的矩阵 boundary 和 disp。boundary 存储各结点的各坐标维度是否受限制，disp 存储各结点的各个坐标维度是否有固定的位移。

函数 dist 负责解析应力边界条件即 B 矩阵，返回一个[2*nodes×1]的矩阵 l_area。l_area 存储各结点各轴向占据的边界长度。

(5) B_matrix.m

函数 B_matrix 负责计算 B 矩阵。

B 矩阵的计算公式：

$$\mathbf{B}(q)=\begin{pmatrix} \frac{\partial N_I^{(q)}}{\partial x} & 0 & \frac{\partial N_J^{(q)}}{\partial x} & 0 & \dots & \frac{\partial N_K^{(q)}}{\partial x} & 0 \\ 0 & \frac{\partial N_I^{(q)}}{\partial y} & 0 & \frac{\partial N_J^{(q)}}{\partial y} & \dots & 0 & \frac{\partial N_K^{(q)}}{\partial y} \\ \frac{\partial N_I^{(q)}}{\partial y} & \frac{\partial N_I^{(q)}}{\partial x} & \frac{\partial N_J^{(q)}}{\partial y} & \frac{\partial N_J^{(q)}}{\partial x} & \dots & \frac{\partial N_K^{(q)}}{\partial y} & \frac{\partial N_K^{(q)}}{\partial x} \end{pmatrix} \quad \text{for } q \in 1,2,\dots,M$$

对于平面问题，B 矩阵是[3×2*NNE]的。对于空间问题，B 矩阵是是[6×3*NNE]的。本案例是平面问题，因此三角形网格的 B 矩阵是[3×6]的，四边形是[3×8]的。

(6) K_matrix.m

函数 K_matrix 负责计算 K 矩阵。

K 矩阵是[dim*nodes×dim*nodes]的。

首先要计算出单个网格的K_e矩阵，编号为 i 的网格的$K_e(i)$矩阵为：

$$K_e(i)=\int B^T C B dV = B^T C B * Area(i)$$

其中，B 矩阵已由函数 B_matrix 计算出，其对于每个网格都是一样的。Area(i)已由函数 g_center 计算出。C 矩阵是平面应力下的刚度矩阵，已知杨氏模量 E 和泊松比υ，各向同性材料的刚度矩阵可以很容易地求出：

$$\mathbf{C}=\frac{E}{1-\nu^2}\begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{pmatrix}$$

单个网格的K_e矩阵是[2*NNE×2*NNE]的，即三角形网格是[6×6]的，四边形是[8×8]的。将[2*NNE×2*NNE]的K_e矩阵映射到[dim*nodes×dim*nodes]的$K_e^{'}$矩阵。$K_e^{'}$矩阵与K_e矩阵对应的结点的值是一样的，其它值置零。

加和各网格的$K_e^{'}$矩阵得到 K 矩阵：

$$K=\sum_{i=1}^{elements} K_e^{'}(i)$$

计算得到的 K 矩阵的非零值呈“三条平行等距的带状”分布。图 5 是网格单元为 200 的三角形网格的 K 矩阵（已由下述划 0 置 1 法调整过），可以明显地观察到 K 矩阵的非零值呈“三条带状”分布。

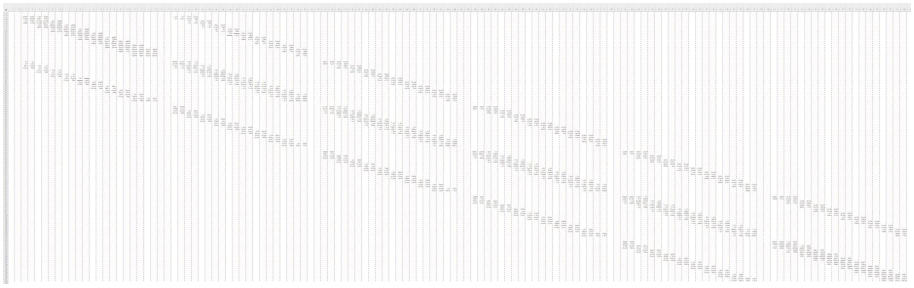


图 5　网格单元为 200 的三角形网格的 K 矩阵

(7) F_vector

函数 F_vector 负责计算各结点上的外力。返回一个[dim*nodes×1]的矩阵 F，储存各结点各轴向受到的外力。

该结点该方向上的外力，等于该结点该方向上的外应力乘该结点该方向占据的边界长度：

$$F(i) = \sigma \cdot l\_area(i)$$

本案例施加的应力为：

$$\sigma = -Load = -2 \times 10^4\ N/m$$

(8) Enforce_BC

函数 Enforce_BC 负责处理位移边界条件，对 K 矩阵和 F 矩阵进行调整。

一种常用的方法是划 0 置 1 法。对于固定位置的位移边界条件，将固定的结点坐标对应的 K 矩阵的行列的元素置 0，然后将对应的对角线元素置 1，同时，将对应的 F 矩阵的元素置 0。

划 0 置 1 法保证了固定的结点坐标的位移始终为零，因此可以成功利用位移边界条件。

(9) 计算各点应变

在完成 K 矩阵和 F 矩阵的计算后，就能获得各结点各坐标的位移了。

位移矩阵是[dim*nodes×1]的矩阵 u：

$$u = K/F$$

(10) 计算各点应力

函数 constitutive.m 负责利用计算出来的各点应变，计算各点应力。

三、结果分析

(1) 两类网格

(a) 三角形网格下的应力分布

运行程序，分别计算网格单元数为 2，100 和 2000 的应力分布。

网格变形和应力分布如图 6 和图 7。

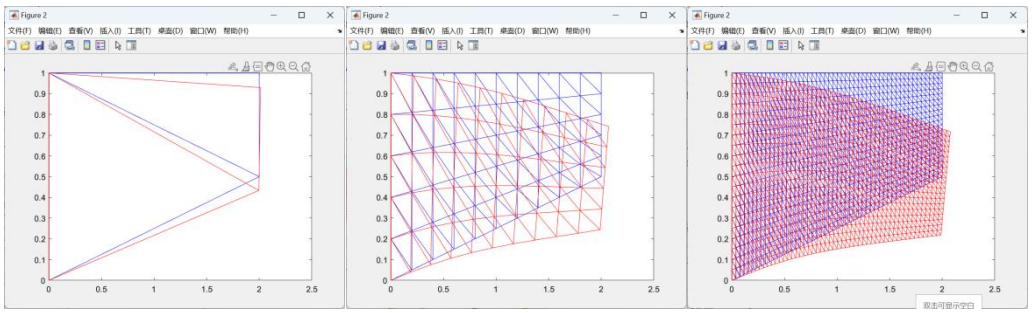


图 6 三角形网格单元数为 2，100 和 2000 的网格变形

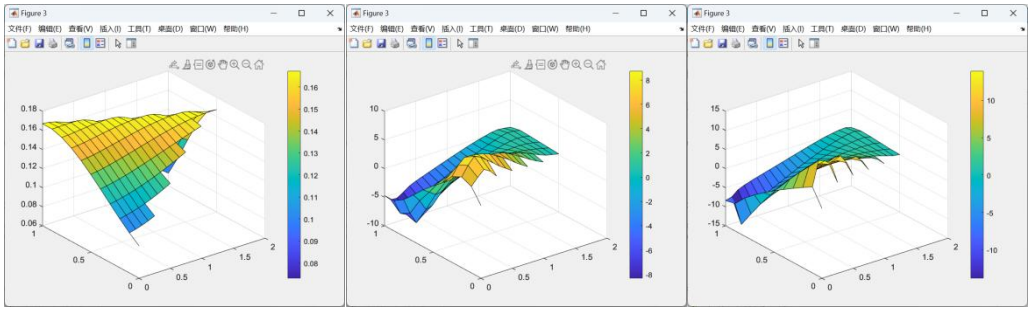


图 7 三角形网格单元数为 2，100 和 2000 的应力分布

可以看出，三种网格数计算出的应力结果差别非常大。网格单元数 2 太少，计算的精度太低，应力分布并不符合实际情况。单元数 100 的应力分布的曲面趋势是正确的，但是精确度依然不足。单元数 2000 的网格密集，能很直观细致地反映梯形板的变形，应力分布贴合实际情况。

(b) 四边形网格下的应力分布

运行程序，分别计算网格单元数为 1，50 和 1000 的应力分布。

网格变形和应力分布如图 8 和图 9。

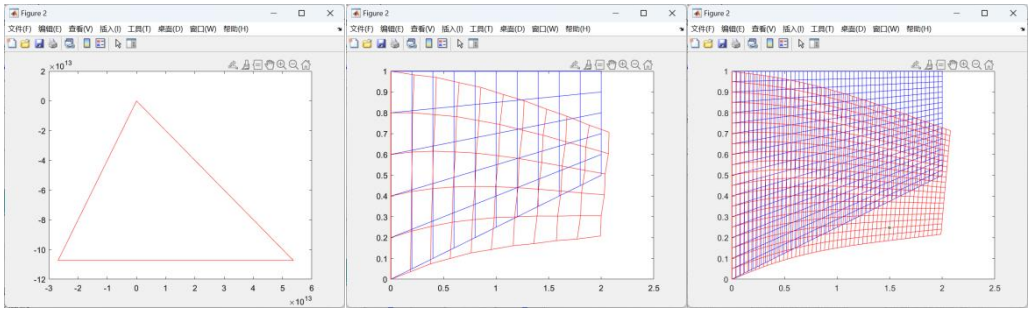


图 8 四边形网格单元数为 2，100 和 2000 的网格变形

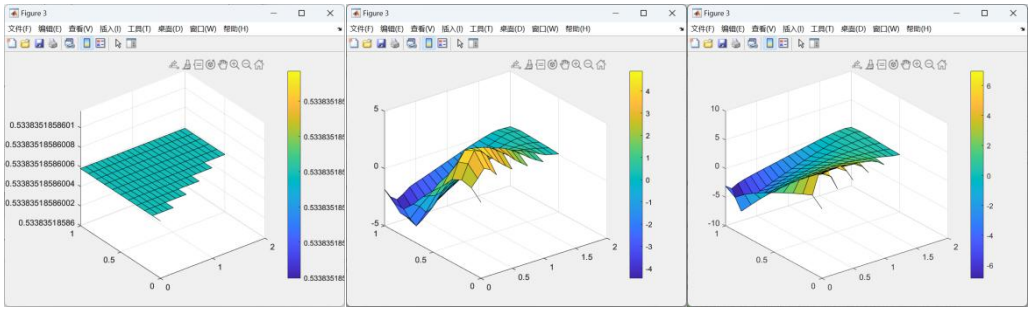


图 9 四边形网格单元数为 2，100 和 2000 的网格变形

同三角形网格的分析结果，三种四边形网格数计算出的应力结果差别非常大。网格单元数 1 得出的应力分布完全不符合实际情况。甚至计算得到的梯形板右上角的位移完全错误，以至无法正确画出变形后的网格。单元数 1000 的网格密集，直观细致，贴合实际情况。

(c) 两类网格的收敛率

(c1) 三角形网格

针对梯形板的 A 点进行考察，如图 1，A 点是梯形板右上角的点。

如表 1，划分不同密集程度的网格，计算得到的 A 点沿 y 轴的应变值。

表 1 三角形网格 A 点沿 y 轴的应变值

$elements$	row_mesh	col_mesh	$h = l_2/col\_mesh$	id_nodes_A	$u_y(x_A, y_A)$
2	1	1	2	4	-0.007041
20	2	5	0.4	12	-0.018962
100	5	10	0.2	22	-0.025767
400	10	20	0.1	42	-0.027731
1024	16	32	0.0625	66	-0.028190
2000	20	50	0.04	102	-0.028329
4096	32	64	0.03125	130	-0.028425
10000	50	100	0.02	202	-0.028474
25600	80	160	0.0125	322	-0.028496
40000	100	200	0.01	402	-0.028501

随着网格的密集，即随着网格大小指标h的减小，A 点沿 y 轴的应变值u_h逐渐收敛于固定值$u = -0.0285m$。

收敛率：$\|e_h\|_1 = \|u - u_h\|_1 \le ch^k$，满足$\lim_{h\to 0} \|u - u_h\|_1 = 0$

可得：$\log \|u - u_h\|_1 = \log c + k \log h$

表 2 是收敛率的表格。

表 2 三角形网格应变值的收敛

h	$\log h$	u_h	$\|e_h\|_1 = \|u - u_h\|_1$	$\log \|e_h\|_1$
2	0.693147	-0.007041	0.021459	-3.841596
0.4	-0.916291	-0.018962	0.009538	-4.652470
0.2	-1.609438	-0.025767	0.002733	-5.902361
0.1	-2.302585	-0.027731	0.000769	-7.170190
0.0625	-2.772589	-0.028190	0.000310	-8.078003
0.04	-3.218876	-0.028329	0.000171	-8.675706
0.03125	-3.465736	-0.028425	0.000075	-9.497026
0.02	-3.912023	-0.028474	0.000026	-10.564901
0.0125	-4.382027	-0.028496	0.000004	-12.456780
0.01	-4.605170	-0.028501	0.000001	-13.470315

如图 10，拟合函数，得到公式：$\log \|u - u_h\|_1 = -3.5772 + 1.8322 \log h$

k取整数，得到：$k = 2$

因此：$\|u - u_h\|_1 \le ch^2$

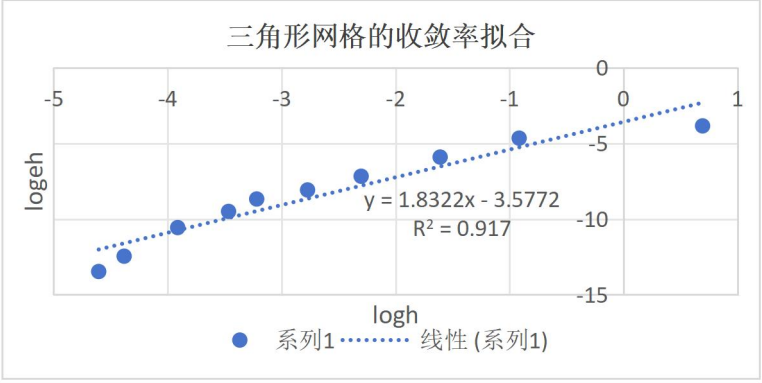


图 10 三角形网格的收敛率拟合

(c2) 四边形网格

如表 3，划分不同密集程度的网格，计算得到的 A 点沿 y 轴的应变值。

表 3 四边形网格 A 点沿 y 轴的应变值

$elements$	row_mesh	col_mesh	$h = l_2/col\_mesh$	id_nodes_A	$u_y(x_A, y_A)$
1	1	1	2	4	-1.07e+13
10	2	5	0.4	12	-0.037642
50	5	10	0.2	22	-0.029344
200	10	20	0.1	42	-0.028703
512	16	32	0.0625	66	-0.028580
1000	20	50	0.04	102	-0.028553
2048	32	64	0.03125	130	-0.028526
5000	50	100	0.02	202	-0.028517
12800	80	160	0.0125	322	-0.028513
20000	100	200	0.01	402	-0.028513

表 4 是收敛率的表格。

表 4 四边形网格应变值的收敛

h	$\log h$	u_h	$\|e_h\|_1 = \|u - u_h\|_1$	$\log \|e_h\|_1$
2	0.693147	-1.07e+13	1.07e+13	30.004756
0.4	-0.916291	-0.037642	0.009142	-4.694885
0.2	-1.609438	-0.029344	0.000844	-7.077751
0.1	-2.302585	-0.028703	0.000203	-8.503844
0.0625	-2.772589	-0.028580	0.000080	-9.429903
0.04	-3.218876	-0.028553	0.000053	-9.838587
0.03125	-3.465736	-0.028526	0.000026	-10.551512
0.02	-3.912023	-0.028517	0.000017	-10.996402
0.0125	-4.382027	-0.028513	0.000013	-11.227892
0.01	-4.605170	-0.028513	0.000013	-11.280608

如图 11，拟合函数，得到公式：$\log \|u - u_h\|_1 = -4.0964 + 1.7191 \log h$

k取整数，得到：$k = 2$

因此：$\|u - u_h\|_1 \leq ch^2$

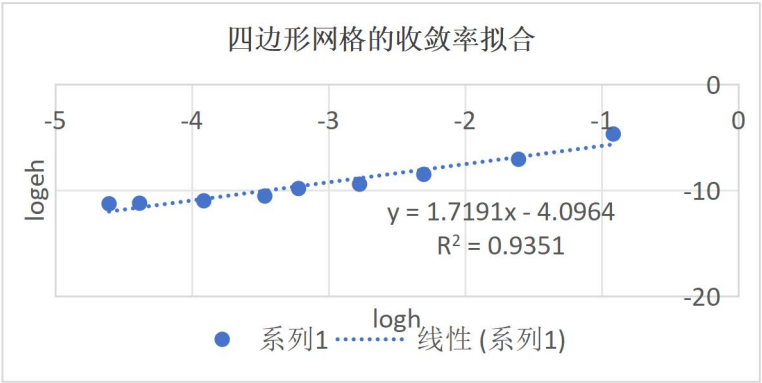


图 11 四边形网格的收敛率拟合

(c3) 对收敛率的讨论

三角形网格和四边形网格都反映了同样的规律：

①网格单元数量越多，网格单元大小越小，计算得到的应变与实际应变越接近。即：

$\lim_{h\to 0}\|u-u_h\|_1=0$。更能反映出实际变形，如图 12，分别是 40000 个网格单元的三角形网格，和 20000 个网格单元的三角形网格，能很好地反映梯形板的形变。

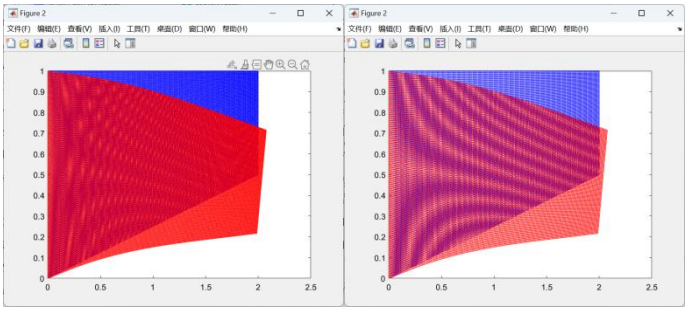


图 12 梯形板在大数量网格下绘制出的形变

②网格的收敛速度与网格单元大小的平方成正比，即：$\|u-u_h\|_1 \le ch^2$。

之所以成平方关系，是因为本案例是平面问题。平面网格是二维的，网格单元面积正比于网格单元大小的平方，即：$Area \propto h^2$。因此相当于：$\|u-u_h\|_1 \le c\cdot Area$。网格单元越小，计算得到的应变的误差越小。

③三角形网格的收敛是要快于四边形网格的。

若k不取整数，三角形网格拟合计算出的$k_t=1.8322$，四边形网格拟合计算出的$k_q=1.7191$。可见$k_t>k_q$，三角形网格比四边形网格能更快地收敛。因为同一组结点，构建的网格单元为三角形时，面积更小，计算的精确程度更高，收敛更快。

(2) 与其它仿真软件的比较

首先使用 Solidworks 进行建模，输出 step 格式文件，如图 13。

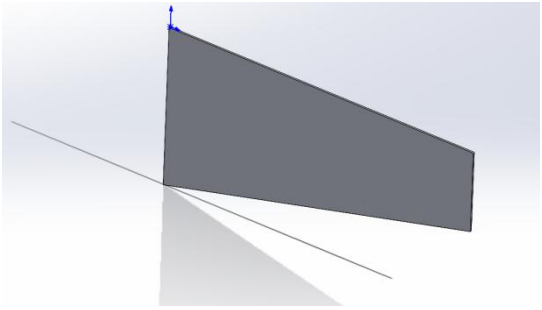


图 13 Solidworks 建模

打开 Comsol。导入 step 文件。网格选择超细化，选择“全部构建”，完整网格包含“79864”个域单元、52516 个边界单元和 952 个边单元。如图 14。

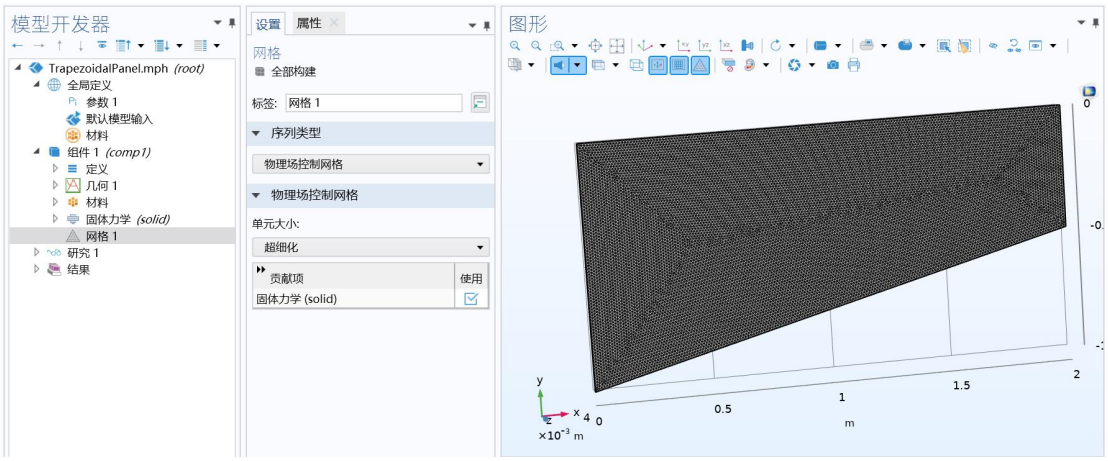


图 14 Comsol 构建网格

添加材料并定义材料属性。添加固体力学物理场的固定约束和边界载荷。添加研究，选择“稳态”。如图 15。

图 15 Comsol 定义材料和物理场

对研究右键点击，选择“计算”，生成结果。如图 16。

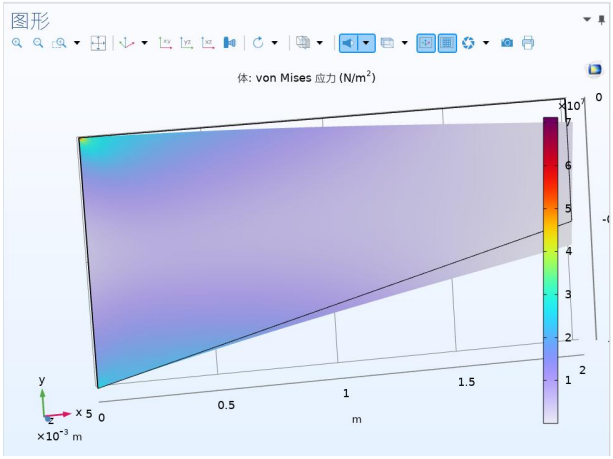


图 16　Comsol 绘制结果

可以获得如下结论：

①matlab 程序和 Comsol 仿真得到的形变结果和数值一致。

②Comsol 的仿真速度更快，可能的原因是额外进行了矩阵运算的优化。

四、总结展望

本案例充分展示了小变形非线性有限元问题的求解过程，通过 matlab 程序直接地了解了相关求解器的设计思路，颇有收获。