

有限元课程大作业报告

白宇轩 刘锦坤 王彦梓 池易行

June 12, 2025

Contents

1 引言	2
2 算法说明	2
3 实现方案	2
3.1 C3D8 六面体单元	3
3.2 S4R 板单元	4
3.3 B31 梁单元	5
3.4 Q4 平面四边形单元	6
4 使用方法	7
5 程序验证与确认	8
5.1 新增单元的独立验证	8
5.1.1 C3D8 六面体单元	8
5.1.2 S4R 板单元	9
5.1.3 B31 梁单元	11
5.1.4 Q4 平面四边形单元	13
5.2 程序整体验证（桥梁算例求解）	13
6 任务分工与合作情况	14
7 结论	14
参考文献	15

1 引言

本次有限元法基础课程编程训练大作业我们提交的是一份基于 xzhang66/STAPpp 修改开发得到的有限元分析程序。Github 仓库链接：<https://github.com/BaicaiBear/STAPpp>

该程序实现了 Bar 杆单元、C3D8 六面体单元、S4R 板单元、B31 梁单元和 Q4 平面四边形单元的有限元分析功能，且相互之间可以联合使用，能够通过 Python 脚本读取 Abaqus 输入文件并转换为 STAPpp 输入文件进行求解。求解环节我们额外实现了两个求解器：基于 Eigen 的 LDLT 求解器和迭代求解器，分别适用于小规模和大规模问题。后处理环节我们实现了将计算结果写入 ParaView 可读的.vtk 文件，并通过 ParaView 进行可视化。

本程序每个新增单元都完成了独立的验证，包括基于算例求解的单元收敛性分析与分片试验。本程序多个单元联合使用也通过求解给定桥梁考题得到验证。

2 算法说明

STAPpp 程序从输入文件中读取有限元模型的节点坐标、单元划分、材料属性、边界条件和载荷等数据，并根据这些数据组装刚度阵与节点力进行求解。

程序实现了 Bar 杆单元、C3D8 六面体单元、S4R 板单元和 B31 梁单元、Q4 平面四边形单元的有限元分析功能，提供了将 Abaqus 输入.inp 格式文件转换为 STAPpp 输入格式的脚本，利用 ParaView 进行后处理与位移、应力云图绘制。

3 实现方案

我们通过修改 xzhang66/STAPpp 程序实现上述功能。为了使其能够兼容 S4R 板单元与 B31 梁单元，我们将节点位移自由度统一扩大到 6（3 个平移自由度和 3 个转动自由度），并匹配修改了原有的 Bar 杆单元的刚度矩阵计算方法。

前处理环节，我们编写了 inp2dat.py 的 python 脚本，识别 Abaqus 输入文件中的关键字并读取信息。在不同单元间的连接方案上，我们相较 Abaqus 原有的主从自由度法略有不同。通过扩大节点自由度，合并坐标相同的节点（节点 bcode 取“或”运算），我们直接使不同种类的单元共用节点，从而实现共有自由度位移的连续性，并将非共有自由度取为自由边界。在载荷施加，我们读取 Abaqus 输入文件中的重力加速度，通过分别计算每个单元的重力体力并将其分配到节点上的方式完成显式提取，一并写入.dat

文件作为程序输入。

求解环节，我们利用 C++ 的 Eigen 线性代数库，新增设计了基于 Eigen 的 LDLT 求解器和迭代求解器，同时保留了原有的 Skyline LDLT 求解器。LDLT 求解器适用于小规模问题，迭代求解器则适用于大规模问题。我们在求解过程中，使用了 Eigen 库的矩阵与向量操作功能，确保了计算的高效性和准确性。

后处理环节，我们在 STAPpp 程序原有的架构上实现了新增单元的节点应力计算，并在输出部分加入了将计算结果写入 ParaView 可读的.vtk 文件的功能，通过 ParaView 实现位移、应力云图的可视化。

下面将详细介绍每个新增单元的编程思路与实现方式。

3.1 C3D8 六面体单元

新建了 CC3D8 单元类，继承自 CElement 抽象基类，实现了其中的抽象方法 CC3D8::Read、CC3D8::Write、CC3D8::ElementStiffness 和 CC3D8::ElementStress 等函数。CC3D8 类中完成了标准 8 节点六面体单元（C3D8）的刚度矩阵计算、节点力后处理等功能。

CC3D8::Read 方法实现了从输入文件中读取单元编号、8 个节点编号以及对应的材料集编号，并将其与全局节点表、材料表进行关联。CC3D8::Write 方法负责将单元的编号信息写入输出流，供程序后处理模块调用。

在 CC3D8::ElementStiffness 方法中，程序采用 8 个高斯积分点（ $2 \times 2 \times 2$ 完全积分）计算刚度矩阵。首先在每个积分点处构造八节点六面体单元的三维形函数及其在自然坐标系下的梯度，然后通过雅可比矩阵将梯度映射到物理坐标系。接着基于线弹性理论构造材料刚度矩阵 \mathbf{D} ，结合应变-位移矩阵 \mathbf{B} ，使用如下公式：

$$\mathbf{K}_e = \sum_{i=1}^8 \mathbf{B}_i^T \mathbf{D} \mathbf{B}_i \cdot \det(\mathbf{J}_i) \cdot w_i$$

在所有高斯点上累加得到单元刚度矩阵 \mathbf{K}_e 。由于 C3D8 单元采用完全积分，不存在剪切闭锁问题，因此无需减缩积分或额外的稳定化项。

CC3D8::ElementStress 方法用于后处理阶段，计算单元中的应力。首先将全局位移向量映射至单元局部自由度向量，再通过 \mathbf{B} 矩阵计算应变，最后使用本构关系 $\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\epsilon}$ 得到应力张量。

3.2 S4R 板单元

新建了 CS4R 单元类，继承自 CElement 类，实现了 CElement 中的抽象方法 CS4R::Read, CS4R::Write, CS4R::ElementStiffness, CS4R::ElementStress 等方法。CS4R 单元类中实现了 S4R 单元的刚度矩阵计算、节点力计算等功能。

CS4R::Read 方法实现了单元编号、4 个节点编号与截面/材料编号的读取；CS4R::Write 实现了单元信息的输出；CS4R::ElementStiffness 实现了 S4R 单元的刚度矩阵计算；CS4R::ElementStress 实现了后处理节点应力的计算。

计算 S4R 单元的刚度矩阵时，参考 Mindlin-Reissner 理论，使用了四边形单元的形函数独立插值挠度 w 和转角 θ_x, θ_y ，分别计算弯曲刚度矩阵 \mathbf{K}_b 与剪切刚度矩阵 \mathbf{K}_s 后相加，为了避免剪切闭锁，我们使用选择性减缩积分——采用 2x2 完全积分计算弯曲刚度矩阵，采用 1x1 减缩积分计算剪切刚度矩阵——使剪切刚度阵奇异，解除剪切闭锁。

刚度阵具体的计算方式如下：

$$\mathbf{B}_{bI}^e = \nabla_s \mathbf{N}_{\theta I}^e = \begin{bmatrix} \partial N_I^e / \partial x & 0 & 0 \\ 0 & \partial N_I^e / \partial y & 0 \\ \partial N_I^e / \partial y & \partial N_I^e / \partial x & 0 \end{bmatrix} \quad (1)$$

$$\mathbf{B}_{sI}^e = \nabla \mathbf{N}_{\dot{w}I}^e - \mathbf{N}_{\theta I}^e = \begin{bmatrix} -N_I & 0 & \partial N_I / \partial x \\ 0 & -N_I & \partial N_I / \partial y \end{bmatrix} \quad (2)$$

$$\mathbf{K}_b = \int_{\Omega_e} \mathbf{B}_b^{eT} \mathbf{D}_b \mathbf{B}_b^e d\Omega_e, \quad \mathbf{K}_s = \int_{\Omega_e} \mathbf{B}_s^{eT} \mathbf{D}_s \mathbf{B}_s^e d\Omega_e \quad (3)$$

$$\mathbf{K} = \mathbf{K}_b + \mathbf{K}_s \quad (4)$$

其中，

$$\mathbf{D}_b = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}, \quad \mathbf{D}_s = kGh \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5)$$

由于这里对应的自由度排序为 $\hat{\mathbf{d}}^e = [w, \hat{\theta}_x, \hat{\theta}_y]$ ，其中 $\hat{\theta}_x = -\theta_y, \hat{\theta}_y = \theta_x$ ，所以需要

刚度阵进行一次变换，记 \mathbf{A} 为变换矩阵满足 $\hat{\mathbf{d}}^e = \mathbf{A}\mathbf{d}_e$ ，则 $\mathbf{K}_{\text{real}} = \mathbf{A}\mathbf{K}\mathbf{A}^T$ 。变换之后，按照节点自由度组装成 24×24 的刚度矩阵，并按照上三角格式储存。

应力计算时采用同样的办法构造刚度阵，从节点位移列表中取出单元节点 w, θ_x, θ_y 三个自由度，矩阵相乘得到剪力与绕 x, y 轴的弯矩。

3.3 B31 梁单元

B31 单元为两节点空间线性梁单元，基于 Timoshenko 梁理论，能够考虑弯曲、剪切、轴向和扭转变形。

在 B31.h 中定义了 B31 单元类，继承自 CElement 类，保留了原有成员函数，添加了一个 GetTransformationMatrix 实现坐标变换矩阵的求解。在 ElementGroup.h 和 Outputter.h 中添加 B31 单元，在 Material.h 中定义 B31 单元的材料参数为弹性模量、剪切模量、截面面积、惯性矩 I_y 和 I_z 以及抗扭常数 J ，同时和相关 cpp 文件中添加 B31 单元部分。在后面求解桥梁算例时，修改材料参数为弹性模量、泊松比、密度、梁的宽高以及壁厚。

在 B31.cpp 中实现单元刚度矩阵。B31 单元的刚度矩阵由轴向、扭转、弯曲与剪切刚度项组成：

$$\mathbf{K} = \mathbf{K}_{\text{axial}} + \mathbf{K}_{\text{torsion}} + \mathbf{K}_{\text{bending}} + \mathbf{K}_{\text{shear}}$$

轴向和扭转刚度矩阵采用解析积分形式，分别为

$$\mathbf{K}_{\text{axial}} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad \mathbf{K}_{\text{torsion}} = \frac{GJ}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

在计算弯曲刚度矩阵和剪切刚度矩阵时构造形函数为：

$$N_1(\xi) = \frac{1-\xi}{2}, \quad N_2(\xi) = \frac{1+\xi}{2}$$

弯曲刚度矩阵使用 2 点 Gauss 积分。对绕 y 轴和 z 轴的弯曲刚度矩阵，构造以下形式的 B 矩阵：

$$\mathbf{B}_b = \frac{dN}{dx} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{D}_b = EI_y \text{ 或 } EI_z$$

弯曲刚度矩阵

$$\mathbf{K}_{\text{bending}} = \int_{-1}^1 \mathbf{B}_b^T \mathbf{D}_b \mathbf{B}_b \frac{L}{2} d\xi$$

剪切应变的 B 矩阵形式为（以 xz 平面为例）：

$$B_{s,y} = \begin{bmatrix} \frac{dN_1}{dx} & -N_1 & \frac{dN_2}{dx} & -N_2 \end{bmatrix}, \quad D_{s,y} = kGA$$

积分采用 1 点减缩积分（ $\xi = 0$, $w = 2$ ）以避免剪切闭锁，剪切刚度矩阵为

$$K_{\text{shear}} = B_s^T D_s B_s \cdot \frac{L}{2} \cdot 2$$

由于单元为空间单元，局部刚度矩阵需通过变换矩阵 T 转回全局坐标系：

$$K_{\text{global}} = T^T K_{\text{local}} T$$

变换矩阵 T 由单元方向向量 e_x 与辅助参考向量构造，保证坐标正交性。

在应力计算部分，单元的应力分量通过从全局位移向量中提取该单元的 12 个自由度，并经过坐标变换后在局部系中计算得到。包含：

- 轴力 $N = \frac{EA}{L}(u_2 - u_1)$
- 扭矩 $T = \frac{GJ}{L}(\theta_{x,2} - \theta_{x,1})$
- 剪力 V_y, V_z 与弯矩 M_y, M_z 根据梁理论导出。

3.4 Q4 平面四边形单元

新建了 CQ4 单元类，继承自 CElement 抽象基类，实现了其中的抽象方法 CQ4::Read、CQ4::Write、CQ4::ElementStiffness 和 CQ4::ElementStress 等函数。该类实现了标准四节点四边形平面应力单元（Q4）的刚度矩阵计算、节点力后处理等功能。

CQ4::Read 方法实现了从输入文件中读取单元编号、4 个节点编号以及对应的材料集编号，并将其与全局节点表、材料表进行关联。CQ4::Write 方法负责将单元的编号信息写入输出流，供程序后处理模块调用。

在 CQ4::ElementStiffness 方法中，程序采用 2×2 Gauss 积分点（完全积分）进行刚度矩阵计算。首先在每个积分点处构造四节点四边形单元的双线性形函数及其在自然坐标系下的梯度，然后通过雅可比矩阵 \mathbf{J} 映射至物理坐标系。进而构造应变-位移矩阵 \mathbf{B} ，并基于平面应力条件下的本构矩阵 \mathbf{D} ，使用如下公式对刚度矩阵积分：

$$\mathbf{K}_e = \sum_{i=1}^4 \mathbf{B}_i^T \mathbf{D} \mathbf{B}_i \cdot \det(\mathbf{J}_i) \cdot w_i$$

其中 \mathbf{D} 表达式为：

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

采用完全积分时，Q4 单元不存在剪切闭锁问题，因此无需减缩积分或额外稳定化处理。最终将单元刚度矩阵 \mathbf{K}_e 按照节点自由度组装成 8×8 矩阵，并按上三角格式进行储存。

CQ4::ElementStress 方法用于后处理阶段，计算单元应力。首先从全局位移向量中提取出单元节点的局部自由度向量 \mathbf{d}_e （共 8 个分量），之后在指定的高斯积分点处，通过 \mathbf{B} 矩阵计算应变矢量： $\boldsymbol{\varepsilon} = \mathbf{B} \mathbf{d}_e$ ，最后使用平面应力状态下的本构关系 $\boldsymbol{\sigma} = \mathbf{D} \boldsymbol{\varepsilon}$ 得到应力矢量 $\boldsymbol{\sigma} = [\sigma_x, \sigma_y, \tau_{xy}]^T$ 。该应力矢量可用于结果可视化和后处理输出。

4 使用方法

按如下格式编写 STAPpp 输入文件 (.dat)，编译 STAPpp 程序，带参数格式运行 stap++（以 macOS 为例：./stap++ [输入文件地址] [求解器选项]）。其中求解器选项数字含义：1-Skyline LDLT 求解器，2-Eigen LDLT 求解器，3-Eigen 迭代求解器。

STAPpp 输入文件格式：

标题行

NUMNP NUMEG NLCASE MODEX

*节点数据

节点编号 bcode (6位，分别表示 $x, y, z, \theta_x, \theta_y, \theta_z$) x y z

*载荷数据（格式未修改，略）

*单元组数据（单元类型：1-Bar杆单元，2-C3D8R六面体单元，3-S4R板单元，4-B31梁单元，5-Q4平面四边形单元；单元附加参数：Area/thickness/b,d,t1,t2,t3,t4）

单元类型 本单元组的单元总数 截面/材料属性组数

截面/材料属性组号 E nu density 单元附加参数

*单元数据（格式未修改，略）

程序运行完成生成.out 输出文件包含详细节点位移和应力信息, .vtk 文件可用 ParaView 打开绘制可视化位移与应力云图。

5 程序验证与确认

每个单元都进行了收敛性分析、分片试验与算例验证；程序验证通过数值求解给定考题并与 Abaqus 对比实现。

5.1 新增单元的独立验证

5.1.1 C3D8 六面体单元

分片实验：

对于 C3D8 单元，我们事实上可以从理论上严格证明其能够通过分片实验 (Patch Test)，从而验证其一致性 (Consistency)：

假设一个给定的线性位移场如下：

$$u = a_0 + a_1x + a_2y + a_3z$$

单元上第 i 个节点处的位移为

$$u_i^{(e)} = a_0 + a_1x_i^{(e)} + a_2y_i^{(e)} + a_3z_i^{(e)}, \quad i = 1, \dots, 8$$

则在单元内部，有限元插值解为：

$$\begin{aligned} u^h &= \sum_{i=1}^8 N_i^{(e)} u_i^{(e)} \\ &= a_0 \sum_{i=1}^8 N_i^{(e)} + a_1 \sum_{i=1}^8 N_i^{(e)} x_i^{(e)} + a_2 \sum_{i=1}^8 N_i^{(e)} y_i^{(e)} + a_3 \sum_{i=1}^8 N_i^{(e)} z_i^{(e)} \\ &= a_0 + a_1x + a_2y + a_3z \end{aligned}$$

这里我们利用了形函数的几何插值性质：

$$x = \sum_{i=1}^8 N_i^{(e)} x_i^{(e)}, \quad y = \sum_{i=1}^8 N_i^{(e)} y_i^{(e)}, \quad z = \sum_{i=1}^8 N_i^{(e)} z_i^{(e)}$$

因此，C3D8 单元可以精确再现任意线性位移场从而说明它满足一致性条件，理论上能够通过分片实验。

在实际计算中，我们设计了一个简单的分片测试算例：给定最小数量的本质边界条件（消除刚体模态），施加符合线性场的自然边界条件。在这种情况下，理论应力应为常量，且单元节点的等效结点力应与解析体力完全一致。

计算结果表明，计算得到的单元应力分布为常量，残余节点力为零，数值结果与理论解完全一致。这验证了我们的实现通过了分片实验，具备良好的单元一致性与数值稳定性。

收敛性分析：

在收敛性分析部分，对于 C3D8 单元，其应该可以精确重构线性场，对于非线性场的重构，理论上应有收敛误差 $e \propto h^2$ ，为了验证这一收敛性，我们选取典型的自重场实体进行分析，经过实验，对不同密度的网格，也即不同 h 大小的实体单元进行算例验证，最后得到双对数曲线如图1所示：

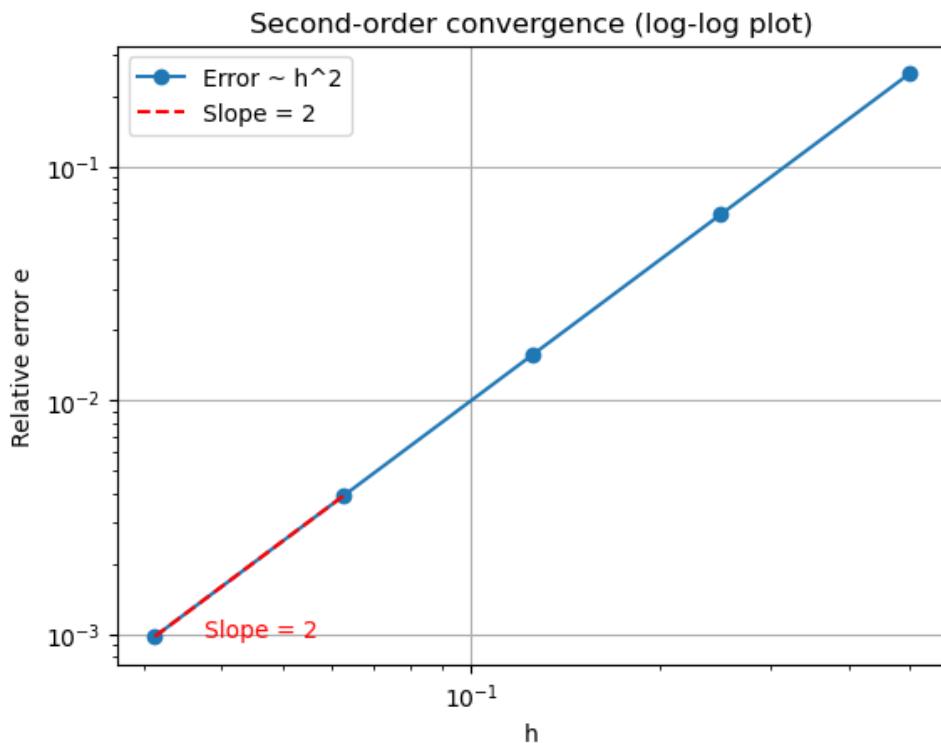


Figure 1: C3D8 收敛性分析

5.1.2 S4R 板单元

收敛性分析：

板单元的收敛性分析通过求解《有限元法基础》例题 6-4 实现：均布载荷 $q =$

$-1N/m^2$ 的四边固定正方形薄板，变长 $L = 8m$ ，厚度 $h = 0.01m$ ，弹性模量 $E = 200GPa$ ，泊松比 $\nu = 0.3$ 。该问题精确解参考《有限元法基础》第 333-334 页。

通过加密网格计算得到的位移 L2 相对误差—网格变长的双对数曲线如图2所示。S4R 单元位移 L2 误差具有 2 阶收敛率，与理论预测一致。

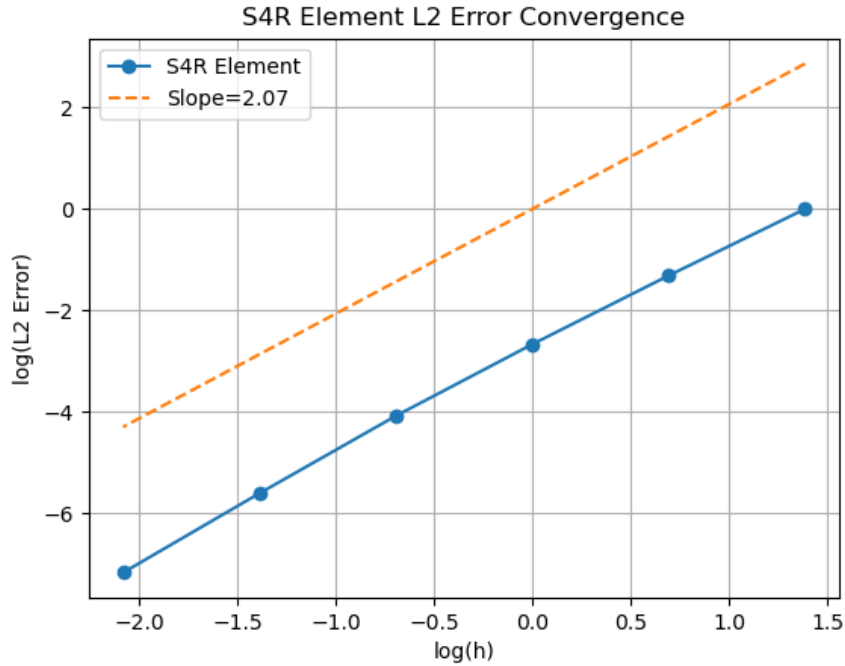


Figure 2: S4R 单元收敛性分析

分片试验：

由于 S4R 单元采用了双线性插值函数，其最高可以精准重构线性位移场。而挠度线性位移时薄板的一种刚体模态。因此在这里只试验了该单元能否重构这一线性挠度场对应的刚体模态。

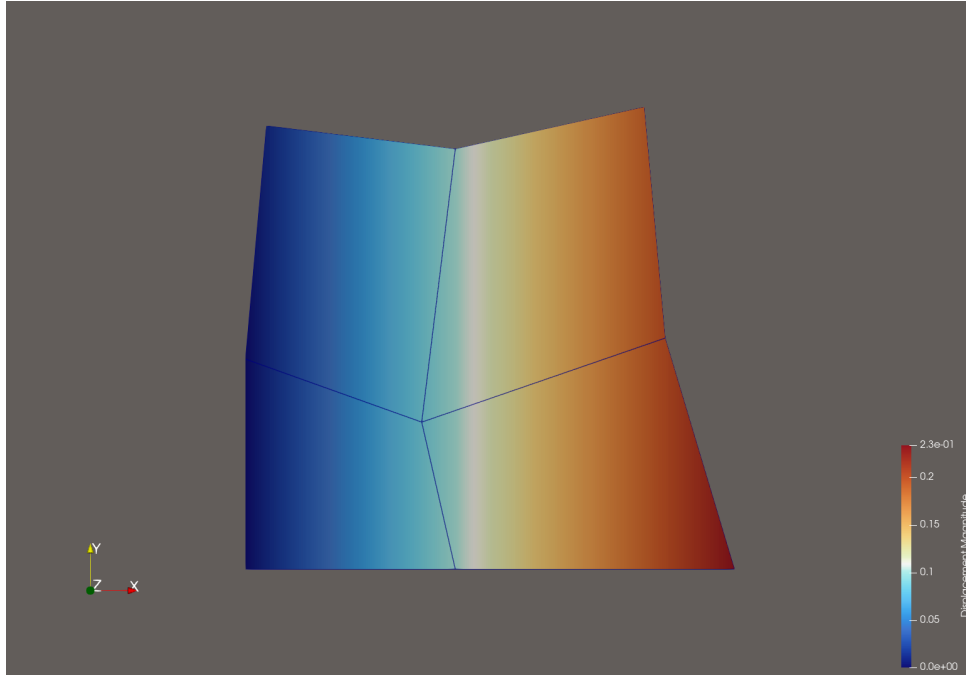
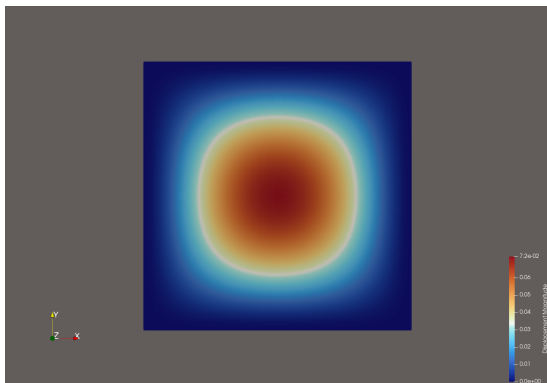


Figure 3: S4R 单元分片试验 TestC

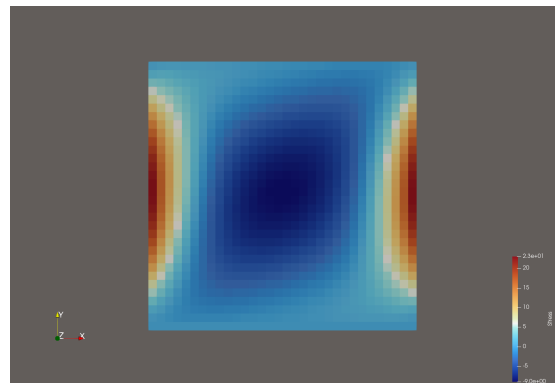
如图3所示，在左下角给定本质边界条件 $w = 0, \theta_x = 0, \theta_y = 0.1$ ，在左侧中点给定本质边界条件 $w = 0, \theta_x = 0$ ，其余节点自由（数值为 0 的自然边界条件）。计算发现有限元解可以精确重构线性挠度场。通过分片试验。

验证算例：

用 8x8 均匀网格求解《有限元法基础》例题 6-4，得到挠度场如图4a，x 方向弯矩如图4b，与实际情况符合良好。



(a) 挠度场



(b) M_x 场

Figure 4: S4R 单元验证算例

5.1.3 B31 梁单元

收敛性分析：

为了验证 B31 单元在实际问题中的数值性能，对其进行收敛性分析，通过 B31-convergence.py 文件实现。以经典的悬臂梁弯曲问题为测试模型，比较有限元解与理论精确解之间的 L^2 误差，考察误差随网格尺寸 h 变化的趋势，进而评估该单元的收敛阶。误差 $\|e\|_{L^2}$ 相对于单元尺寸 $h = L/N$ 的双对数图如下：

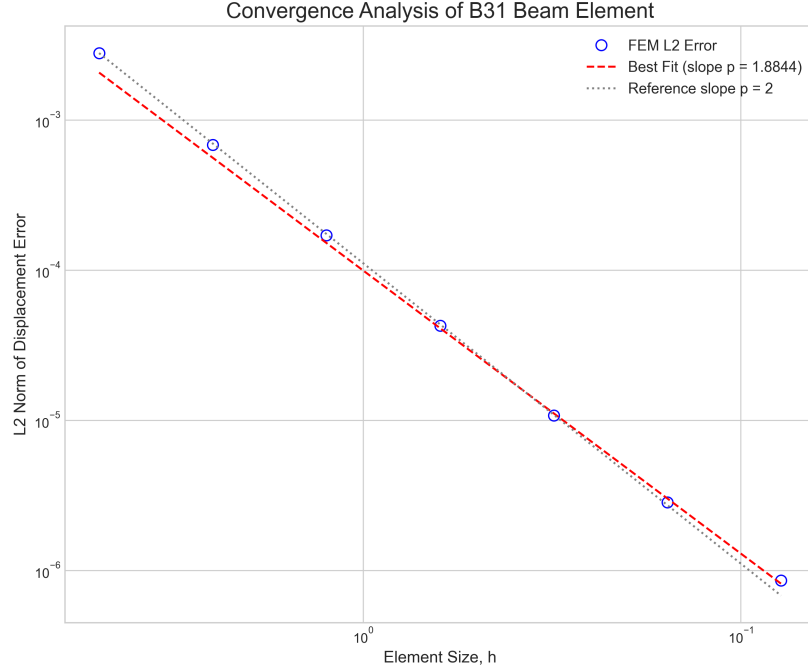


Figure 5: B31 单元收敛性分析

随着 h 减小，误差迅速降低，验证了 B31 单元良好的收敛性。通过对 $\log h$ 与 $\log \|e\|$ 拟合，得到收敛阶约为 $p \approx 1.97$ 。拟合曲线几乎与理论参考线 $\mathcal{O}(h^2)$ 重合，表明 B31 单元在弯曲主导问题中具有二阶收敛性。

分片试验：

为了进一步验证 B31 单元的一致性，采用制造解的方法进行分片实验。设计一个常曲率的理论解进行比对。考虑一根水平梁，其竖向位移为预设的二次函数形式：

$$u_y(x) = ax^2, \quad \text{其中 } a = 0.005$$

其余自由度全部为零。此位移场对应边界条件为在梁的端节点施加集中弯矩。对于每个离散单元数量 N ，从有限元结果中提取节点的 u_y 和 θ_z ，并与理论解比较，结果一致。

B31 单元的收敛性分析和分片实验见 B31-convergence branch 部分。

5.1.4 Q4 平面四边形单元

新建了 CQ4 单元类，继承自 CElement 基类，实现了标准 4 节点四边形平面单元 (Q4) 的刚度矩阵计算和应力恢复功能。本单元采用双线性形函数和完全高斯积分方案，适用于平面应力/应变问题。

5.2 程序整体验证（桥梁算例求解）

通过 inp2dat.py 将桥梁 Abaqus 输入文件 Bridge-1.inp 转换为 Bridge-1.dat 后求解，经过 ParaView 后处理可视化后如图6。与 Abaqus 求解结果7一致性良好。

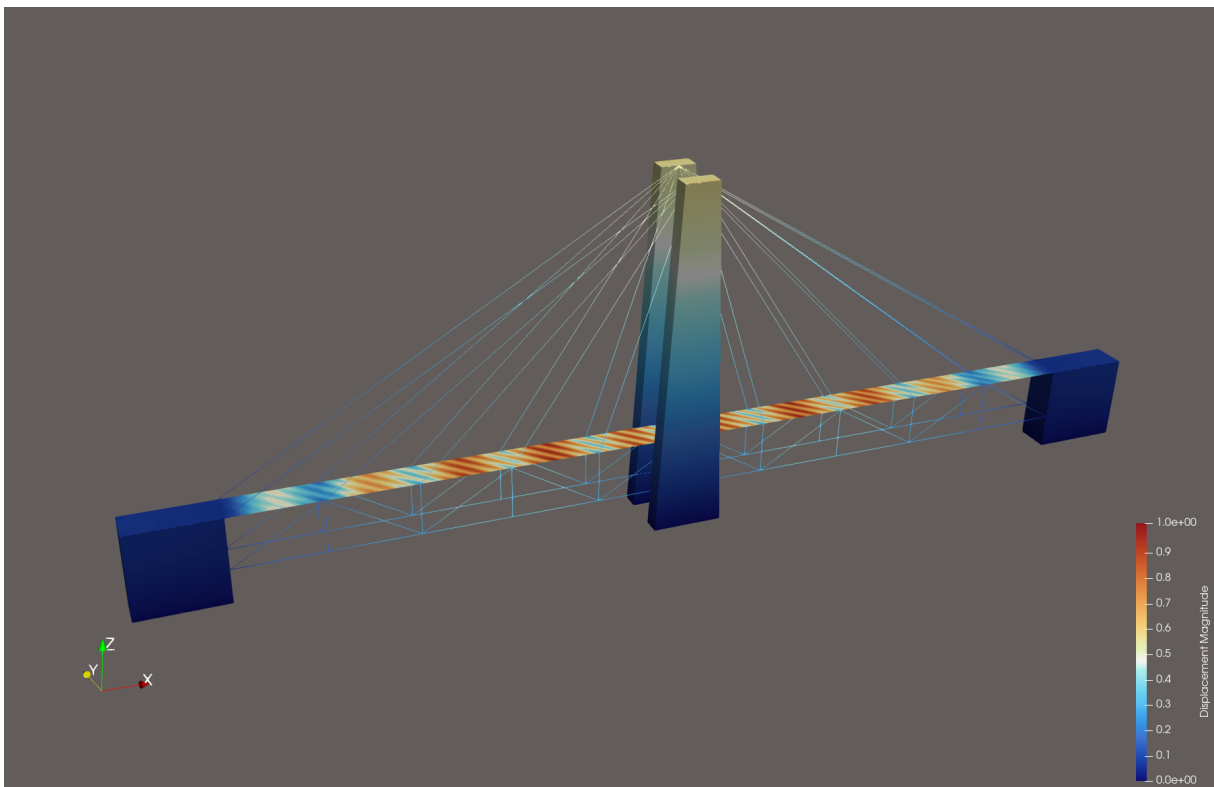


Figure 6: 考题 Bridge-1 的 y 方向位移

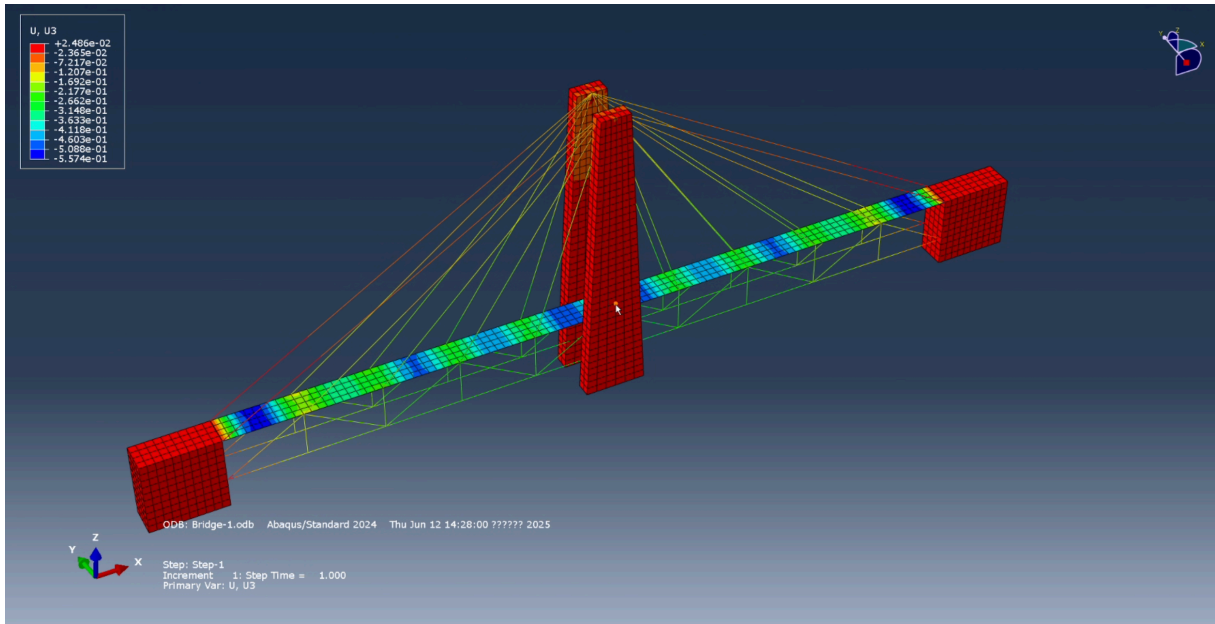


Figure 7: Abaqus 求解得到 Bridge-1 的 y 方向位移

6 任务分工与合作情况

本次小组合作编程由白宇轩、刘锦坤、王彦梓、池易行合作完成。其中白宇轩主要负责 S4R 单元、输入文件转换脚本、后处理 ParaView 绘图、求解器编写与项目代码管理；刘锦坤主要负责 C3D8R 单元、Abaqus 计算与项目代码管理；王彦梓主要负责 B31 单元；池易行主要负责 Q4 单元。单元验证每个人负责自己的单元，考题求解由白宇轩与刘锦坤共同完成。所有人均参与了本报告撰写。

7 结论

本次有限元法基础课程大作业成功开发并验证了一个基于 xzhang66/STAPpp 修改而来的有限元分析程序。该程序在原有 Bar 杆单元的基础上，增加了 C3D8 六面体单元、S4R 板单元、B31 梁单元和 Q4 平面四边形单元的有限元分析功能，还实现了不同单元间的联合使用，并提供了 Abaqus 输入文件转换工具和 ParaView 可视化后处理功能。

通过对每个新增单元进行独立的收敛性分析、分片试验与算例验证，以及对整体程序通过桥梁考题的求解验证，结果表明本程序具备良好的数值稳定性和准确性，能够满足求解的需求。特别是在求解器方面，新增的基于 Eigen 的 LDLT 求解器和迭代求解器为不同规模问题提供了可行的高效计算方案。本次合作项目充分展示了团队在有限元理

论和编程实践方面的能力。

参考文献

- Zhang, X. (2023). 有限元法基础. 北京: 高等教育出版社.