

基于 ABAQUS 二次开发的 腹板开孔钢板混凝土组合结构抗剪性能数值分析

朱秦昊 2024312648

1 选题背景

钢-混凝土组合结构充分发挥材料特点，是基于钢筋混凝土结构、钢结构发展而来的一种新型组合结构形式。组合结构通过合理结合钢、混凝土两种材料，充分利用不同材料的特点与性能，并在承载力、延性等方面表现出较高的性能^[1]。

钢板混凝土组合结构是在组合梁、组合柱的基础上演化发展而来，其典型构成为外侧钢板、核心混凝土、界面（抗剪）连接件以及腹间连接系^[2]。在该结构体系中，外侧钢板是主要受拉构件，并对核心混凝土施加约束；核心混凝土作为主要受压构件，并对钢板提供有效支撑；界面连接件连接不同材料，同时提高钢板的局部稳定性；腹间连接系连接外侧钢板，提升整体性，并提升抗剪、抗冲击能力。凭借优异的承载力、延性、抗震性能以及良好的抗冲击性能、防水性能^[4-5]，该结构已广泛应用于高层剪力墙、核工程、桥塔等领域，并在海洋与地下结构、防护工程和隧道结构中展现出巨大的应用前景，目前已有相关工程实例^[6-7]。

得益于钢板混凝土组合结构的优越性能，其作为新尝试运用于港珠澳大桥、深中通道等沉管隧道工程中，如图 1 所示。与传统的钢筋混凝土结构相比，新组合结构体积、重量更小，便于安装与运输，减小航道疏浚量，减少对原有水域生态环境的影响；抗冲击爆炸能力好；防水、防渗性能好，对混凝土开裂不敏感，间接增强耐久性；周围钢板可充当混凝土浇筑模板，并可采用自密实混凝土从顶部开孔处进行浇筑，施工便捷度高。

实际工程中常涉及在钢腹板上进行开孔以加快浇筑过程、提高施工效率、简化施工流程，但腹板开孔会对结构抗剪性能产生重要影响，对抗剪承载力及刚度影响巨大，但目前缺乏对腹板开孔钢板混凝土组合的相关研究，限制了工程设计，影响了性能提升，增加了潜在风险，无法满足更加普遍更广泛的需求。

本研究前期已进行的 6 个钢板混凝土组合结构大比例缩尺试验开展，试验研究了腹板开孔形状、开孔尺寸对结构抗剪承载力的影响，分析了不同试件的破坏

模式以及裂缝发展模式。外荷载与跨中挠度的荷载-位移曲线如图 1 所示，试验汇总结果如表 1 所示。

各 ScO 试件破坏模式如图 2 所示，剪切试验破坏过程如下：

加载初期底部连接件处由跨中向支座逐渐出现微小竖向裂缝，荷载逐渐增大，竖向裂缝发展到型钢连接件与核心混凝土接触面后转化为斜裂缝，并向加载位置处延伸。后续加载中三条主裂缝逐步出现，并不断斜向伸长至构件顶部，发展成贯穿全高的主斜裂缝。随着荷载增加，主斜裂缝之间会出现更多微小裂缝，导致裂缝变密，并出现混凝土剥落、型钢连接件裸露。结构抗剪承载力较弱，破坏受混凝土斜向压溃控制，属于剪压破坏模式。

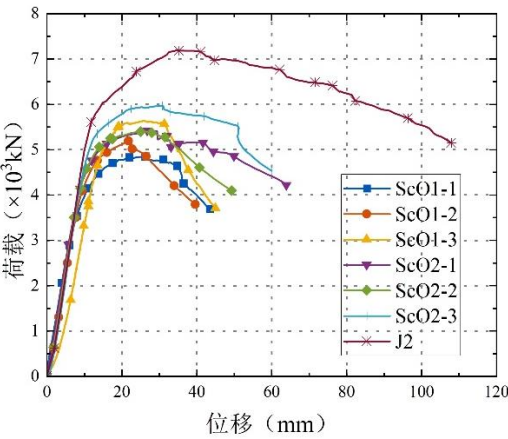


图 1 抗剪试件荷载-位移曲线

表 1. 抗剪试件试验结果汇总（单位：mm、kN）

编号	试件	破坏模式	δ_{yt}	F_{yt}	δ_{ut}	F_{ut}
1	ScO1-1	剪压破坏	11.98	4264	26.18	4852
2	ScO1-2	剪压破坏	13.03	4703	21.79	5190
3	ScO1-3	剪压破坏	18.30	5411	19.70	5641
4	ScO2-1	剪压破坏	12.85	4863	26.82	5418
5	ScO2-2	剪压破坏	12.96	4936	26.12	5399
6	ScO2-3	剪压破坏	14.84	5474	30.74	5968
7	J2	弯剪破坏	18.60	6291	34.70	7194

注： δ_{yt} 为屈服位移； F_{yt} 为屈服荷载； δ_{ut} 为峰值位移 F_{ut} 为峰值荷载。

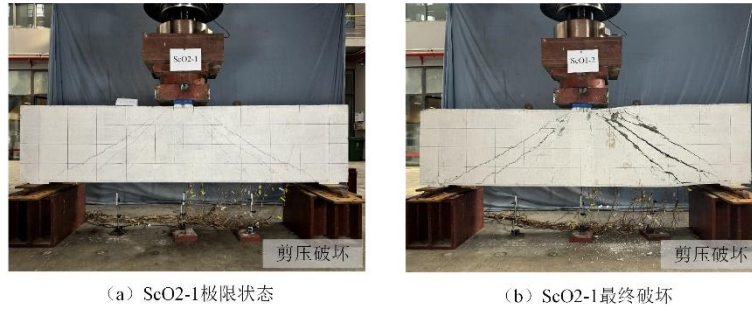


图 2 抗剪试件典型极限荷载状态与最终破坏状态

2 选题目的

本文聚焦于腹板开孔钢板混凝土抗剪性能的数值分析，基于 Abaqus 有限元分析软件，着重考虑翼缘板、剪力连接件、钢腹板、开孔等处与混凝土作用的精细界面行为，考虑可能存在的接触、摩擦、粘聚力等作用。材料模型考虑钢材的塑性、强化行为，考虑混凝土的开裂、塑性、损伤行为，并合理设置混凝土三维本构参数以考虑约束作用。对比分析不同界面设置、本构模型对分析结果的影响，合理平衡准确度与效率，开发高效模型及相关算法，基于试验结果从极限承载力、荷载-位移曲线、构件破坏模式等方面进行参数标定。此外，采用 Python 语言对 Abaqus 进行二次开发，实现 Python 建模与分析功能，为后续开展大范围参数分析提供工具支撑，系统揭示关键参数对结构抗剪性能的影响规律。

3 有限元模型

对于腹板开孔钢板混凝土组合结构的剪切受力分析，试验研究是基础，数值分析是其重要工具和补充。开发精准高效的数值模型，腹板开孔钢板混凝土组合结构的工作机理、扩充不同工况计算结果、验证相关理论，都起着重要的作用。

3.1 几何条件与网格划分

有限元模型使用三维实体-壳混合单元剪建立。构件有限元网格划分分别如图 3、图 4 所示，其中钢结构采用壳单元建立，混凝土采用实体单元建立。根据模型总体尺寸，综合考虑计算准确性与效率，经网格测试，钢结构采用 40mm 网格，混凝土采用 100mm 网格。

支座处模拟实际支撑条件，分别设置相关边界约束。加载点处设置了钢垫板加载块防止应力集中，并将其上表面耦合至加载参考点，使用位移控制方式进行

加载。此外，计算开启大变形选项以考虑几何非线性。

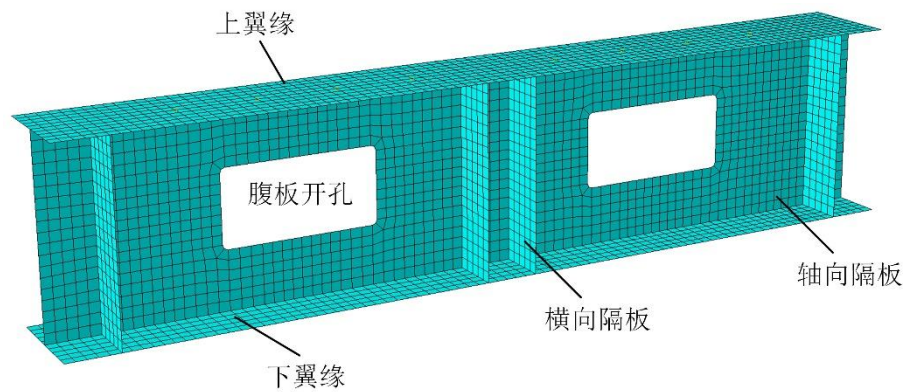


图 3 构件钢结构有限元模型网格划分

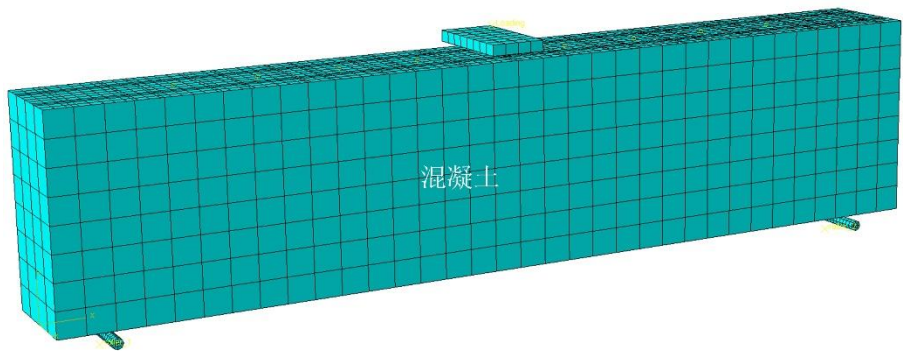


图 4 构件混凝土有限元模型网格划分

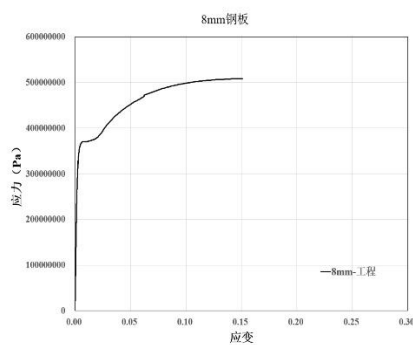
3.2 界面连接与接触

结构中钢板以焊接形式连接，为简化分析过程，将各钢板部件 Merge 成统一模块，采用共节点约束。由于轴向腹板与横向腹板完全嵌入在混凝土中，将其与混凝土通过 Embed 进行耦合，上、下翼缘与混凝土之间通过 Surface-to-surface 接触进行定义。由于翼缘处的界面摩擦、横向隔板和少量连接件已可以提供充分的剪力连接，我们在模型剪力时进行了适当简化，将连接件和加劲肋进行适当等效。根据试验结果，裂缝往往先出现于连接件位置处，我们将连接件位置处的翼缘节点与混凝土节点设置 tie 约束，提高区域刚度。

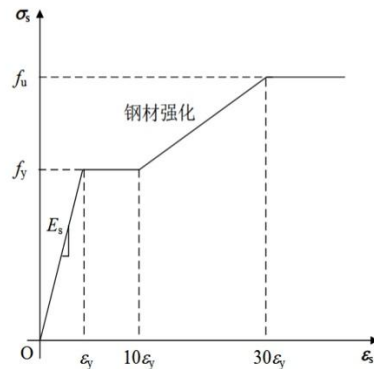
3.3 材料本构

钢材使用 Von-Mises 屈服面以及等向强化法则。本构曲线最初采用图 6（a）所示的试验测得曲线，计算中发现使用图 6（b）所示的简化曲线对结果影响很小，基本可以忽略，因此本研究中采用如 6（b）所示的简化钢材本构曲线，以便定量分析钢材屈服、极限强度的影响。钢材弹性模量采用 $2.05 \times 10^5 \text{Mpa}$ ，泊松比为 0.3。

混凝土采用基于 GB50010-2010 规范的塑性损伤模型 (CDP)，并根据材性试验所测数据计算真实混凝土受压、受拉本构曲线。



(a) 试验测得本构曲线



(b) 简化的钢材本构曲线

图 6 钢材的本构曲线

4 计算结果

4.1 荷载-位移曲线

以试件 ScO1-2 为例，有限元计算得到的荷载-位移曲线与试验的对比如图 7 所示。对荷载-位移曲线进行分析，存在 3 个典型的加载阶段。第一阶段为初始阶段，整个构件为弹性状态直到混凝土出现结构性裂缝，此时结构的整体刚度出现变化。因钢结构部分主要承受拉应力且刚度相比于混凝土较大，所以裂缝开裂后整体荷载-位移曲线上的刚度变化不明显。第二阶段混凝土裂缝持续发展，因钢结构部分未进入屈服，结构整体仍表现为弹性。随后轴向隔板进入屈服，整个构件也进入屈服，此时结构进入塑性段，荷载随位移缓慢增加。在屈服阶段，由于混凝土受力不均匀，受力较大的部分逐渐开始受压软化，此部分混凝土的应力水平开始下降。加载位置处混凝土首先压溃并退出工作，整体结构的刚度变为负值，此时有限元求解也停止。

具体数据如表 2 所示。从计算结果中可知有限元的模拟结果良好，受剪承载力的误差一般在 5% 以内，破坏模式同为剪压破坏，因此有限元计算结果对于理论设计方法有一定的参考价值，可以模拟腹板开孔钢板混凝土组合结构的受剪过程模拟。

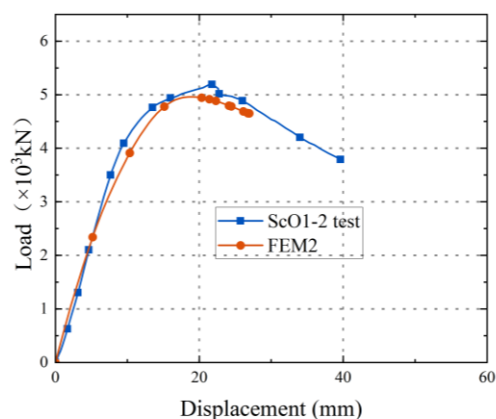


图 7 试验与有限元荷载-位移曲线对比

表 2. 有限元-试验承载力对比（单位：kN）

编号	试件	破坏模式	试验峰值荷载	有限元峰值荷载	误差
1	ScO1-1	剪压破坏	4852	4802	-1.0%
2	ScO1-2	剪压破坏	5190	4959	-4.4%
3	ScO1-3	剪压破坏	5641	5729	1.6%
4	ScO2-1	剪压破坏	5418	5636	4.0%
5	ScO2-2	剪压破坏	5399	5529	2.4%
6	ScO2-3	剪压破坏	5968	5957	-0.2%

注： δ_{yt} 为屈服位移； F_{yt} 为屈服荷载； δ_{ut} 为峰值位移 F_{ut} 为峰值荷载。

4.2 裂缝发展

以构件 ScO1-2 为例，有限元计算得到的裂缝发展如图 8 所示。在图 8 中，有限元结果中的裂缝以混凝土受拉损伤表示。在开裂点时，底部的受拉裂缝与斜向受剪裂缝已经出现；屈服点时，裂缝长度进一步延伸，宽度同时增加；屈服后不久，斜裂缝发展至构件顶部，贯穿全高；极限荷载状态时，斜裂缝更加明显，同时平行出现了更多的斜裂缝。有限元计算结果与图 2 试验中的观察结果相似。该模型已能较好模拟腹板开孔钢板混凝土组合结构受剪全过程，为后续开展大范围参数分析奠定基础。

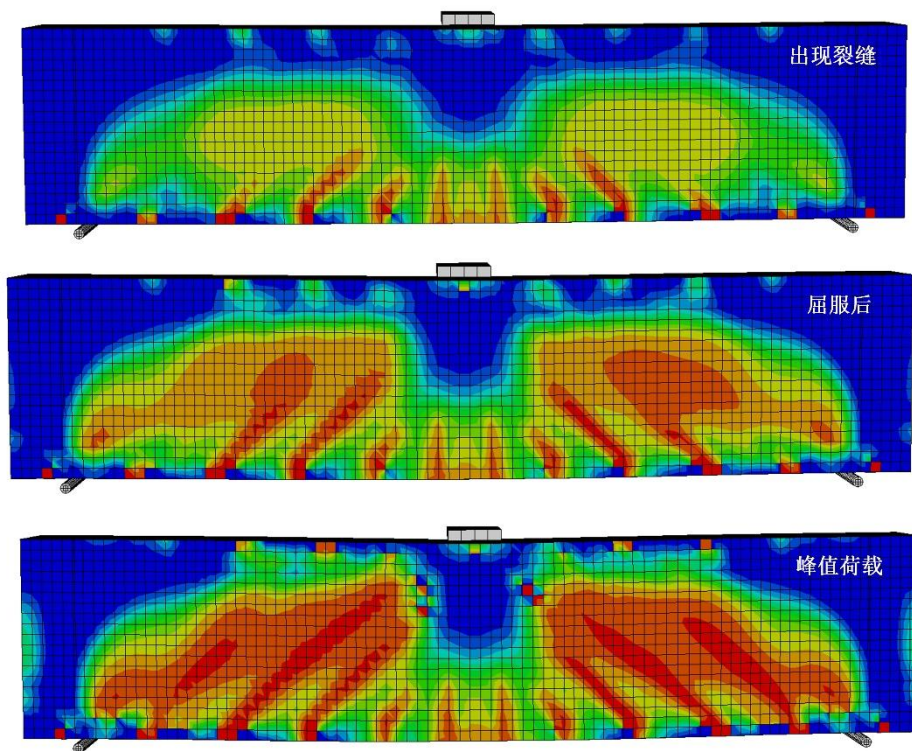


图 8 有限元剪切裂缝模拟

5 基于 Python 语言的 Abaqus 二次开发

随着有限元分析在结构工程领域中的广泛应用，对仿真过程的定制化与自动化需求日益增长。Abaqus 作为功能强大的有限元分析软件，提供了基于 Python 语言的二次开发接口，支持用户通过脚本控制仿真建模、分析提交及结果处理等全过程操作。相比于传统的图形界面操作，基于 Python 语言的建模方式在参数化设计、批量建模、模型复用和自动化分析等方面具有显著优势，成为科研与工程实践中不可或缺的开发手段。

5.1 Python 与 Abaqus/CAE 的关系

Abaqus/CAE 作为 Abaqus 软件中的前处理模块，承担了模型构建、分析设置与提交等核心功能。其底层架构本质上以 Python 语言为核心，通过 Python 解释器解析用户指令，并与 Abaqus 的命令接口、图形界面以及脚本系统进行交互。用户可以通过三种方式与 CAE 进行通信：图形用户界面(GUI)、命令行接口(CLI)和 Python 脚本。这些操作方式最终均会被统一转化为 Python 解释器可识别的命令，推动模型构建或分析作业的执行。

建模过程中，用户在 GUI 中进行的所有操作，系统会自动记录为.rpy 格式的 Replay 脚本文件，形式上为 Python 命令序列。用户可以通过修改这些脚本文件实现模型的参数化修改或批量操作。此外，用户也可以直接编写或运行自定义的 python 脚本，以更加灵活地控制建模流程或执行后处理任务。Abaqus/CAE 接收到 Python 命令后，将自动生成.inp 格式的输入文件并调用求解器进行分析，输出.odb 格式的结果数据库文件供后续读取与可视化。

因此，可以认为 Abaqus/CAE 中的核心操作实质上是对 Python 指令的封装与图形化呈现，二次开发实质上是对这些 Python 指令的直接调用与优化。通过构建高层次的 Python 脚本，用户可实现复杂建模逻辑、批量结构仿真和结果自动提取，显著提升仿真效率和可控性。

5.2 GUI 与 Python 脚本建模方法

在实际建模流程中，GUI 操作与 Python 脚本之间具有天然的对应关系，用户可基于以下两种典型方法实现混合建模：

方法一：基于 Replay 文件的参数化建模。

当用户通过 GUI 完成初始模型构建后，Abaqus 会自动生成包含建模指令的 Replay 文件（.rpy），该文件记录了每一个操作对应的 Python 命令。用户可以在此基础上进行手动编辑，引入参数化控制语句、循环结构或逻辑判断，实现模型的灵活复用和批量建模。此方法适用于已有建模模板的参数优化场景，具备操作简便、上手快速等优点。

方法二：使用 PythonReader 实现建模命令提取。

为进一步提升脚本获取的精准性与实时性，可借助专用工具如“PythonReader”软件，该工具能够实时监听用户在 GUI 中的建模操作，并将对应的 Python 命令以标准化格式输出并保存为脚本文件。这种方式能够准确捕捉每一步的建模细节，尤其适用于复杂结构或包含大量操作的模型开发。生成的脚本文件不仅可作为调试与学习的参考，更可作为高效建模的基础框架。

综上所述，无论采用 Replay 文件编辑还是借助命令提取工具，GUI 操作与 Python 脚本之间的映射关系为实现参数化、自动化建模提供了坚实基础。通过对二者的融合使用，可在可视化建模与脚本逻辑之间实现灵活切换，大幅提升有限

元建模的效率与可扩展性，为复杂结构问题的仿真研究提供强有力的技术支撑。

5.3 腹板开孔钢板混凝土组合结构 Python 脚本建模

在钢板混凝土组合结构的有限元建模中，为提高建模效率、确保模型一致性并开展参数化分析，采用基于 Python 脚本的建模方式具有重要优势。本研究以具有开孔腹板的钢板混凝土组合结构为研究对象，借助 Python 脚本实现结构模型的高效构建与参数化分析，重点关注开孔几何对力学性能的影响，并探索其在非连续域下建模与网格划分的可控性。

5.3.1 模型建构流程与 Python 命令整合

根据研究中的建模方法，复现有限元模型。本研究首先采用 Abaqus/CAE 图形界面进行模型搭建，并借助 PythonReader 软件实时提取每一建模操作对应的 Python 命令。该软件能够将用户在 GUI 中的点击操作自动转换为标准 Python 代码，并导出为结构清晰的脚本文件，从而便于用户在后续分析中进行灵活调用和批量控制。

在模型构建过程中，主要包括部件创建、几何布尔操作、材料属性定义、截面赋值、装配与交互设置、加载与边界条件设置、网格划分及作业提交等核心步骤。重点关注腹板开孔相关代码，如图 9 所示，在命令整合中，将开孔的长、宽、半径等几何参数定义为变量，并构造循环语句实现开孔参数的批量遍历。在本次建模过程中，将混凝土材料定义封装为函数模块，定义混凝土损伤塑性（CDP）模型以便后续相同属性模型使用，如图 10 所示。

通过整合上述步骤所提取的 Python 语句，构建了可完整自动生成目标模型与分析提交的脚本框架，并对关键步骤进行了模块化封装，提升了建模代码的可维护性与可扩展性。

```
# 开孔参数设置
opening_parameters = [
    (400.0, 400.0), # 参数1
    (500.0, 300.0), # 参数2
    (600.0, 200.0) # 参数3
]

for i, (opening_length, opening_width) in enumerate(opening_parameters):
    model_name = f'Model-{i+1}'
    mdb.Model(name=model_name)

s = mdb.models['Model-1'].ConstrainedSketch(name='__profile__',
    sheetSize=5000.0)
g, v, d, c = s.geometry, s.vertices, s.dimensions, s.constraints
s.setPrimaryObject(option=STANDALONE)
s.rectangle(point1=(-1700.0, 375.0), point2=(-1700.0, -375.0)) #web整体参数
s.rectangle(point1=(-750+0.5*opening_length, 0.5*opening_width), point2=(-750-0.5*opening_length, -0.5*opening_width)) #左开孔参数
s.rectangle(point1=(750-0.5*opening_length, 0.5*opening_width), point2=(750+0.5*opening_length, -0.5*opening_width)) #右开孔参数
```

图 9 开孔参数设置

```
def create_concrete_material(model_name):
    """
    Function to create and define the concrete material in the specified model.

    Parameters:
        model_name (str): The name of the model in which the material will be created.
    """
    mdb.models[model_name].Material(name='concrete')
    mdb.models[model_name].materials['concrete'].Elastic(table=((29463.0, 0.2), ))
    mdb.models[model_name].materials['concrete'].ConcreteDamagedPlasticity(table=((45.0, 0.1, 1.16, 0.667, 0.005), ))

    # Compressive behavior
    mdb.models[model_name].materials['concrete'].concreteDamagedPlasticity.ConcreteCompressionHardening(
        table=((13.092452, 0.0), (18.252399, 2.9e-05), (19.449714, 3.67e-05),
            (21.349061, 5.23e-05), (24.751495, 9.5e-05), (31.485932, 0.000323591),
            (33.5568, 0.000722952), (32.384849, 0.00092612), (29.618965, 0.001175386),
            (26.292766, 0.00144884), (23.044769, 0.001704802), (20.15112, 0.001956881),
            (17.679411, 0.00219752), (15.60496, 0.002426634), (13.872484, 0.002645822),
            (12.423161, 0.002856788), (11.204533, 0.003061048), (10.172981, 0.003259866),
            (9.293331, 0.00345426), (8.53758, 0.003645039), (7.883512, 0.003832847),
            (7.313478, 0.004018194), (6.813393, 0.004201489), (6.371955, 0.004383058),
            (5.980032, 0.004563166), (5.630201, 0.004742027), (5.316379, 0.004919817),
            (4.777555, 0.005272742), (4.332701, 0.005622834), (3.960056, 0.005970722),
            (3.643919, 0.006316861), (3.37272, 0.006661586), (3.137776, 0.007005149),
            (2.932457, 0.007347741), (2.751628, 0.007689511), (2.591251, 0.008030576),
            (2.448116, 0.008371028)))

    # Tensile behavior
    mdb.models[model_name].materials['concrete'].concreteDamagedPlasticity.ConcreteTensionStiffening(
        table=((2.67528, 0.0), (2.87024, 0.000802581), (2.94774, 0.000899951),
            (2.66681, 0.00110949), (2.27912, 0.00132264), (1.95032, 0.00153381),
            (1.6928, 0.00174255), (1.49252, 0.00194934), (1.33477, 0.0021547),
            (1.20829, 0.00235899), (1.10506, 0.00256249), (1.0194, 0.0027654),
            (0.947259, 0.00296785), (0.885709, 0.00316994), (0.832582, 0.00337174),
            (0.786256, 0.00357331), (0.745495, 0.0037747), (0.709342, 0.00397592),
            (0.677045, 0.00417702), (0.648008, 0.00437801), (0.62175, 0.0045789),
            (0.597882, 0.00477971), (0.576082, 0.00498045), (0.556086, 0.00518113),
            (0.537673, 0.00538175), (0.520654, 0.00558233), (0.504873, 0.00578286),
            (0.490194, 0.00598336), (0.266076, 0.00799097), (0.167958, 0.0099943),
            (0.116464, 0.011996), (0.0860074, 0.0139971), (0.0664308, 0.0159977),
            (0.053057, 0.0179982), (0.043487, 0.0199985)))

    # Concrete compression damage
    mdb.models[model_name].materials['concrete'].concreteDamagedPlasticity.ConcreteCompressionDamage(
        table=((0.0, 0.0), (0.01618608, 2.897e-05), (0.01917568, 3.668e-05),
            (0.02477357, 5.232e-05), (0.03827999, 9.504e-05), (0.09628021, 0.00032359),
            (0.18256074, 0.00072295), (0.221013, 0.00090568), (0.26495848, 0.00110932),
            (0.31203709, 0.00132522), (0.36020642, 0.00154673), (0.40785202, 0.00176938),
            (0.45380356, 0.00199045), (0.49729037, 0.00220848), (0.53786823, 0.0024228),
            (0.57534017, 0.00263323), (0.60968458, 0.00283989), (0.64099624, 0.00304299),
            (0.66944121, 0.00324283), (0.69522428, 0.00343974)))
```

图 10 混凝土函数封装

5.3.1 模型建构流程与 Python 命令整合

由于腹板中设置开孔会破坏其几何连续性，使得模型在几何处理和网格划分过程中出现不规则性，导致仿真精度与计算效率面临挑战。在建模过程中发现，若采用 GUI 中自动划分网格的操作虽然能显著提升建模与计算速度，但容易在开孔边缘产生网格质量不均的情况，进而引发局部应力集中或收敛困难。此外，不同开孔尺寸下自动网格策略对模型结果影响不一致，导致模型的可重复性与可

对比性下降。

为应对此问题，本文尝试从两个方向探索改进途径：

一方面，计划开发“开孔-网格协调”算法，依据开孔尺寸自动调整网格分布密度，确保开孔区域附近具备足够细致的单元划分而其他区域仍维持高效建模要求。该算法可通过 Python 脚本集成在网格模块中，并根据开孔尺寸动态调节种子密度与网格控制策略。

另一方面，在当前研究阶段，仍采用自动划分网格策略并针对不同开孔尺寸进行独立标定。即，在每一类参数组合下通过试验结果对模型进行误差修正与对比标定，以降低因网格不一致所引入的系统误差。尽管该方法暂时无法实现跨尺寸的统一建模体系，但为开展参数影响研究提供了可操作的中间路径。

综上所述，基于 Python 的建模方式不仅提升了模型复现效率，还为参数化开孔研究提供了高度灵活的技术平台。通过对材料属性模块的独立封装、对开孔结构的参数控制以及对网格策略的系统思考，为复杂组合结构的自动化建模与性能分析奠定了坚实的技术基础。

6 心得体会

通过本次围绕钢板混凝土组合结构开展的研究工作，我深刻体会到土木工程结构数值分析在科研与工程实践中的重要意义。研究过程中，我以试验结果为基础，对组合结构进行了系统的有限元建模与模拟分析，并通过 Abaqus 平台构建了高精度的数值模型。在此基础上，我进一步借助 Python 脚本语言实现了模型的参数化建模，大幅提升了模拟效率与可复用性。通过对比分析不同参数组合下结构的力学响应，不仅加深了我对组合结构受力机理的理解，也让我在实际操作中逐步掌握了 Abaqus 建模流程与 Python 脚本编写方法，为后续的科研工作打下了坚实基础。

同时，通过学习“土木与建筑工程 CAE”课程，我对计算辅助工程技术在土木工程领域的广泛应用有了更加全面的认知。课程内容涵盖了从可视化表达、几何模型构建、有限元离散化，到智能化分析与设计优化的全过程，系统性地展示了现代 CAE 技术的核心思想与技术路径。在课程的大作业与小作业中，我主要聚焦于结构模型的构建与离散化处理，这不仅帮助我更好地理解数值模拟的基本

逻辑，也提升了我将理论知识转化为工程实践能力的水平。

更为重要的是，课程中所介绍的智能化建模与数据驱动分析等前沿内容引发了我对未来研究方向的深入思考。在当前土木工程逐步迈向智能化、自动化的大趋势下，如何将人工智能技术与传统工程建模手段深度融合，正成为研究的新热点。我希望能在后续的研究中进一步探索结构参数识别、模型自适应更新、结果智能评估等方向，使自身研究不断接轨技术前沿，拓宽学术视野，提升工程思维，最终服务于更高质量的土木工程结构设计与分析。

7 对本课程的建议

《土木与建筑工程 CAE》课程内容安排合理、体系完整，涵盖了可视化、模型化、离散化、智能化等多个层面，具有较强的理论深度与实践指导意义。胡老师讲解细致、态度亲切，对待学生十分认真负责。通过设置大作业与小作业，课程有效激发了同学们自主探索的积极性，鼓励大家结合自身兴趣选择研究方向，在实际操作中提升对建模工具与工程问题的理解能力。

希望在课程教学中能引入更多与 CAE 相关的研究热点与最新进展分享，例如智能建模、数据驱动仿真、AI 辅助设计等内容。通过讲解实际科研案例或前沿成果，不仅可以加深同学们对所学知识的理解，也有助于激发对未来研究的思考与兴趣。期待本课程在今后的教学中继续优化，不断丰富教学内容，进一步推动土木工程与信息技术的深度融合。

参考文献

- [1] 聂建国, 陶慕轩, 黄远, 等. 钢-混凝土组合结构体系研究新进展 [J]. 建筑结构学报, 2010(06):71-80. (NIE Jianguo, TAO Muxuan, HUANG Yuan, et al. Research advances of steel-concrete composite structural systems [J]. Journal of Building Structures, 2010 (06):71-80. (in Chinese))
- [2] ODUYEMI T O S, WRIGHT H D. An experimental investigation into the behaviour of double-skin sandwich beams [J]. Journal of Constructional Steel Research, 1989, 14(3): 197-220.
- [3] LIEW J Y R, SOHEL K M A, KOH C G. Impact tests on steel-concrete-steel sandwich beams with lightweight concrete core [J]. Engineering Structures, 2009, 31(9): 2045-2059.
- [4] ROBERTS T M, EDWARDS D N, NARAYANAN R. Testing and analysis of steel-concrete-steel sandwich beams [J]. Journal of Constructional Steel Research, 1996, 38(3): 257-279.
- [5] NIE J G, HU H S, FAN J S, et al. Experimental study on seismic behavior of high-strength concrete filled double-steel-plate composite walls [J]. Journal of Constructional Steel Research, 2013, 88:

206-219.

- [6] 聂建国, 陶慕轩, 樊健生, 等. 双钢板-混凝土组合剪力墙研究新进展 [J] . 建筑结构, 2011, 41(12): 52-60. (NIE Jianguo, TAO Muxuan, FAN JianSheng, et al. Research advances of composite shear walls with double steel plates and filled concrete[J]. Building Structure, 2011, 41(12): 52-60. (in Chinese))
- [7] WRIGHT H D, ODUYEMI T O S, EVANS H R. The experimental behaviour of double skin composite elements [J] . Journal of Constructional Steel Research, 1991, 19(2): 97-110.