

基于 ABAQUS 的平面钢框架静力分析插件

李伟

2014210131

一、选题背景

1、自 20 世纪 60 年代、有限元方法被提出至今，该方法已经比较完善，并产生了众多成熟的通用有限元分析平台，行业内的常规问题已经基本可以较圆满解决，有限元方法是解决实际问题的首选方案；

2、为了更高效的利用有限元方法解决问题，除开发新的有限元算法外，应当基于这些成熟的有限元平台进行二次开发，将前人长期积累的工程经验用于解决问题；

3、结构工程师使用通用有限元软件时，更希望根据自身需要开发本专业领域内的模块、函数与界面，将软件中无关部分剔除；

4、对于前处理（如快速建模）、后处理（如创建与访问数据库、按需生成结果文件）与自定义模块（如自动后处理模块）等确实存在着相关需求；

5、通用有限元软件 ABAQUS 提供了专门的二次开发接口，包括用户子程序与 ABAQUS 脚本接口。其中脚本接口在 Python 语言基础上定制开发，为进行相关二次开发提供了可能与便利；

6、脚本编程与 INP 建模相比，可以更加方便地进行参数分析与快速建模。

二、研究目的

基于在背景中提到的相关内容，应可以在 ABAQUS 平台上进行简单的插件开发，使之更符合土建工程师的思考习惯与输入习惯，并将与结构计算无关的内容与繁琐的操作隐藏起来。为了考察这种构想的可能性，选择一个简单但是典型的平面钢框架为主体进行插件开发。

三、研究内容

1、搭建插件界面并完成核心脚本的编写，使之成为一个较为独立的结构分析插件；

2、具体可实现的内容包括：快速建立几何模型，材料、截面、质量的快速输入，荷载输入，挠度曲线与应力云图的显示等。

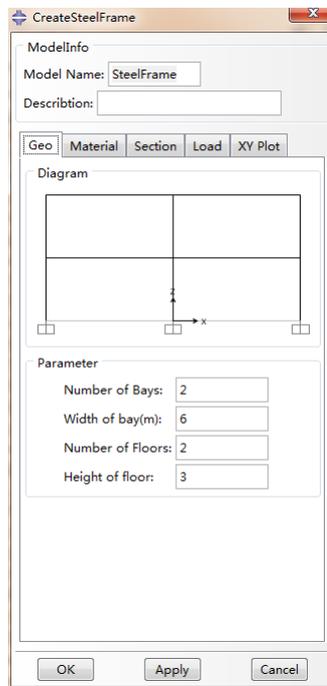
四、界面说明

1、整体说明

整体界面可分为两部分：上半部分需要对结构进行命名并对结构概况进行描述，下半部分使用标签页的方式分别对结构的几何、材料、截面、荷载、输出模式等参数进行输入或选择，是插件的主体部分。

以下对于各项输入进行进一步说明。

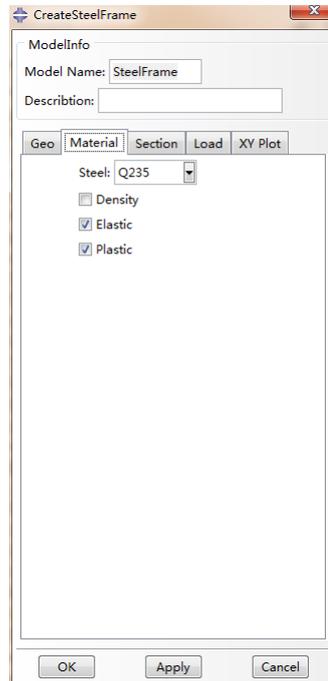
2、快速搭建几何模型



将平面框架抽象成等跨度、等层高的框架，因此分别输入跨数、跨距、层数、层高后即可迅速搭建。

当建立起框架的主体形状后，由于草图中保存了全部的约束（如竖直、水平、平等、垂直等）信息，可以方便的根据实际形状对于结构的几何参数进行修改。

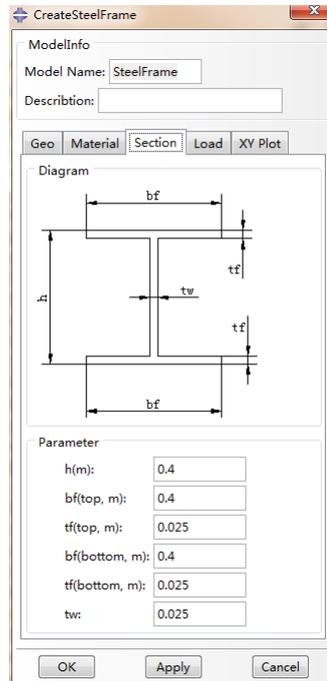
3、材料信息输入



提供了常用的钢材属性，选择后可以直接输入到 ABAQUS 模型树中的材料库中。由实际经验知，材料属性对于正确、快速获取计算结果至关重要，因此各大设计院均在此方面积累了大量的经验，可以通过此种方式充分利用。

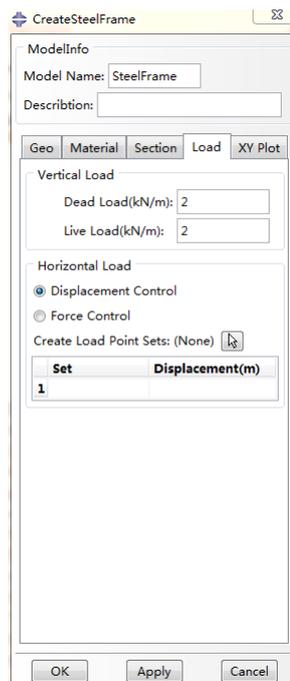
具体到材料参数，根据实际需要提供了密度、弹性性质、塑性性质等选项，方便使用者按需选择。

4、截面信息



与几何建模类似，在此插件中简单的以工字形截面来建立。可以根据需要自行修改。

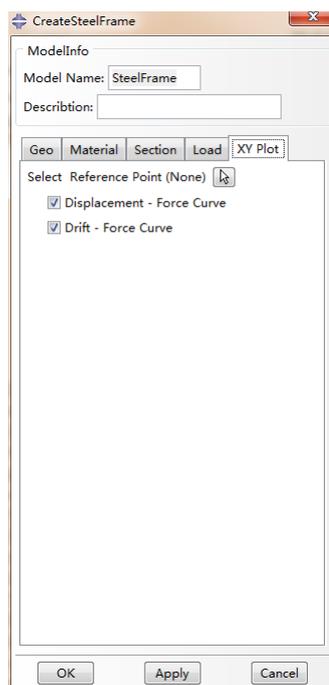
5、荷载信息



首先按照土建工程师的习惯，输入恒载、活载，由程序自动根据规范计算荷载后添加到构件上。

插件中提供了位移加载与力加载两种方式，具体操作上需要先选择一个点做为加载点，然后输入加载目标。

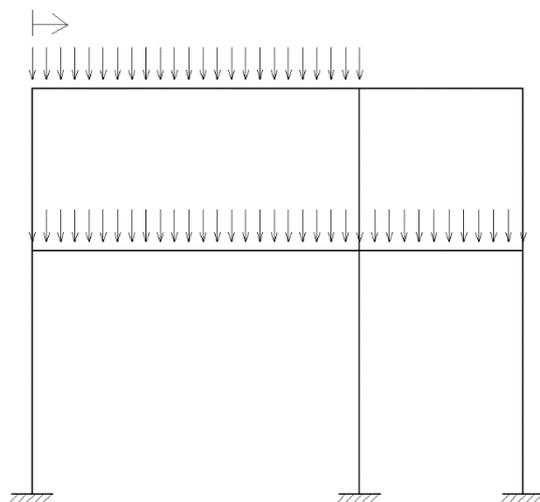
6、荷载信息



提供力一位移曲线与力一层间位移角曲线两种曲线方式，可以在 ODB 中直接生成相关数据对，在 ABAQUS 可视化界面中直接选择即可。

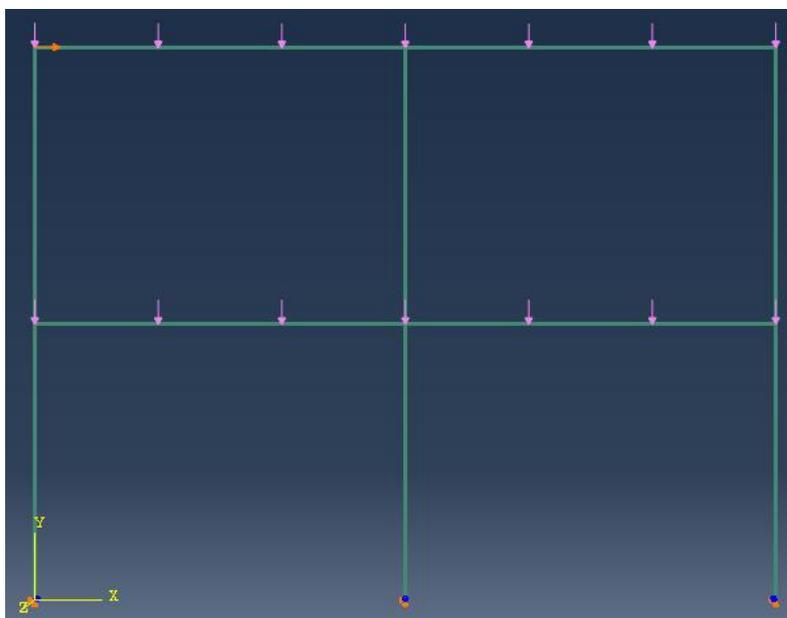
五、使用实例

以下图所示框架为例，说明如何利用该插件进行分析。



1、插件输入

首先根据插件界面提示，建立起一个等跨等层高、满跨加载的框架。

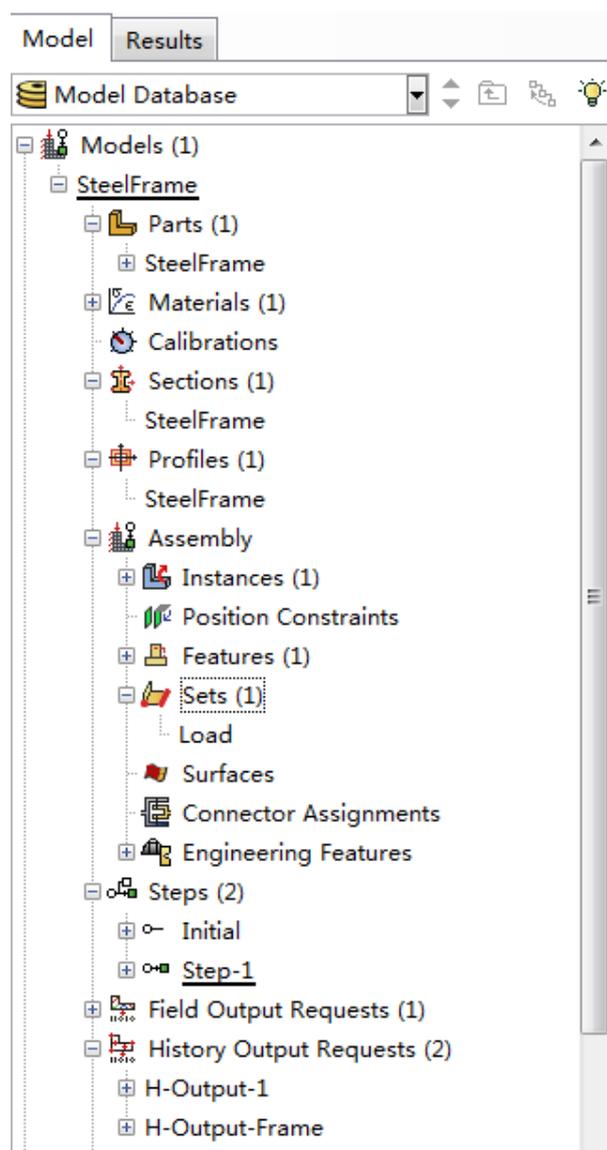


2、模型修改

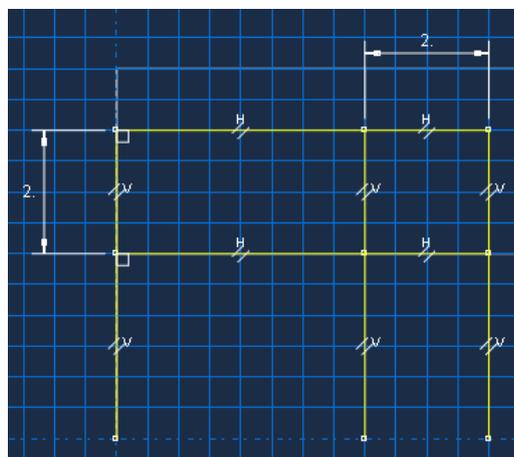
由于几何模型、荷载与实际模型不符，因此需要进一步调整。

由下图中 ABAQUS 模型树显示，在插件中输入后，全部按 ABAQUS 中的形式保

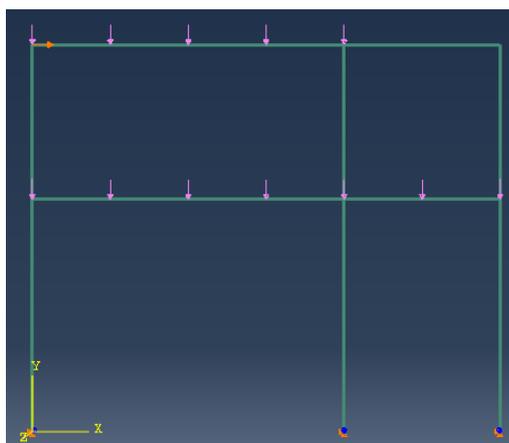
存了全部的信息，全部可以由用户按照 ABAQUS 的相关用法进行修改。



在本实例中，主要需要对几何模型等进行修改。在草图中保留了相关约束，因此可以通过修改标注的方式修改。



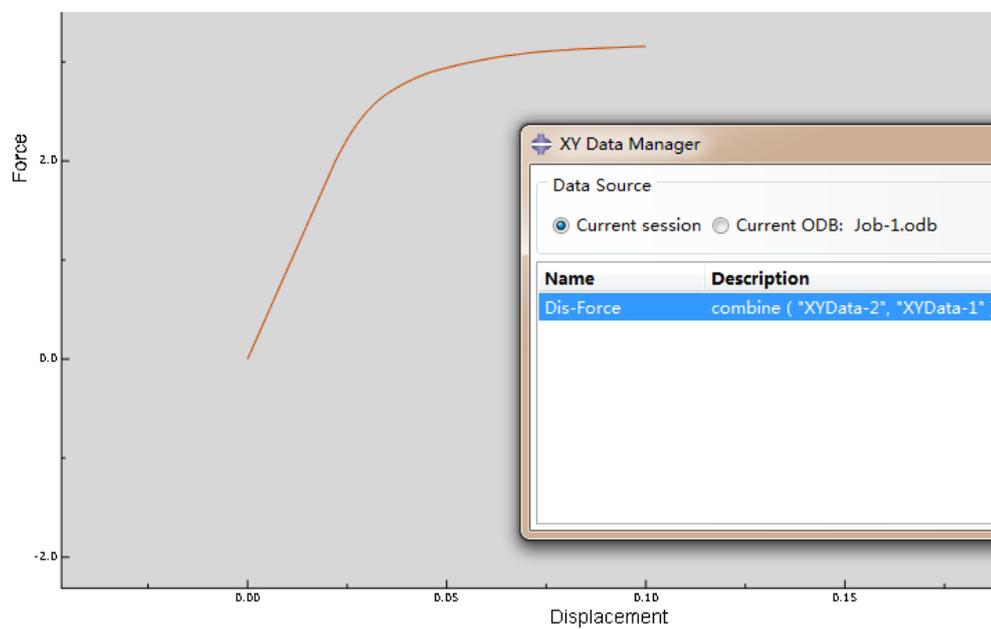
进行其他修改后可得到最终的计算模型。



3、后处理

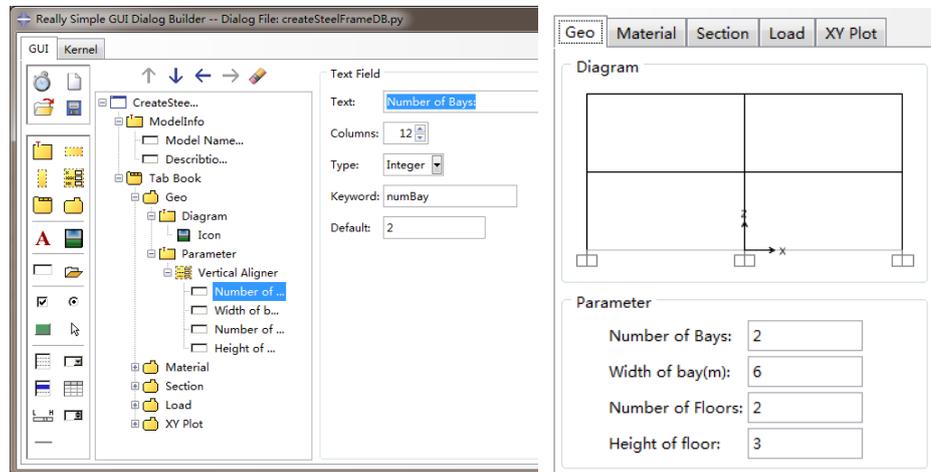
选择输出力-位移曲线。

传统的 ABAQUS 后处理需要自行选择数据对，当输出量大时会比较费时。本插件的功能之一即是自动将相关的数据对选择出来，用户可以数据库中查看。



六、技术手段

1、RSG 对话框构造器



RSG 对话框构造器是 ABAQUS 内置的 GUI 界面编辑器,提供了诸多 Abaqus GUI 工具包的简化接口。

使用 RSG 有以下优势:

- (1) 各种控件种类齐全,几乎可以涵盖全部常用操作;
- (2) 输入框会给出相关提示信息,用户根据提示即可进行编辑,无需专门学习相关知识。由于所有输入参数均需提前定义数据域类型,当输入异常时自带异常抛出;
- (3) 提供了比较简捷的布局形式,可以较好实现控件的对齐布局。

然而其也有一些不利之处。主要是当用户对于控件有高级需求时(如需要将两种基本控制复合),只能通过改写脚本文件的方式。由于相关参考资料较少,需要编写者有较高的代码编写水平,成本较高。且在改写脚本的过程中无法实时查看,也提高了制作难度。

2、内核脚本

```
def createSketch(modelName, numBay, bayWidth, numFloor, floorHeight):
    if modelName in mdb.models.keys():
        del mdb.models[modelName]

    session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE)
    #create Model
    Mdb()
    frameModel = mdb.Model(name=modelName)
    #=====
    #create Sketch
    frameSketch = frameModel.ConstrainedSketch(name=modelName, sheetSize=max(2*numBay*bayWidth, 2*numFloor*floorHeight))
    frameSketch.setPrimaryObject(option=STANDALONE)

    #Draw Columns
    for noColumn in range(numBay+1):
        xyCoordsColumnBottom=(bayWidth*noColumn, 0.0)
        xyCoordsColumnTop=(bayWidth*noColumn, floorHeight*numFloor)
        frameSketch.Line(point1=xyCoordsColumnBottom, point2=xyCoordsColumnTop)

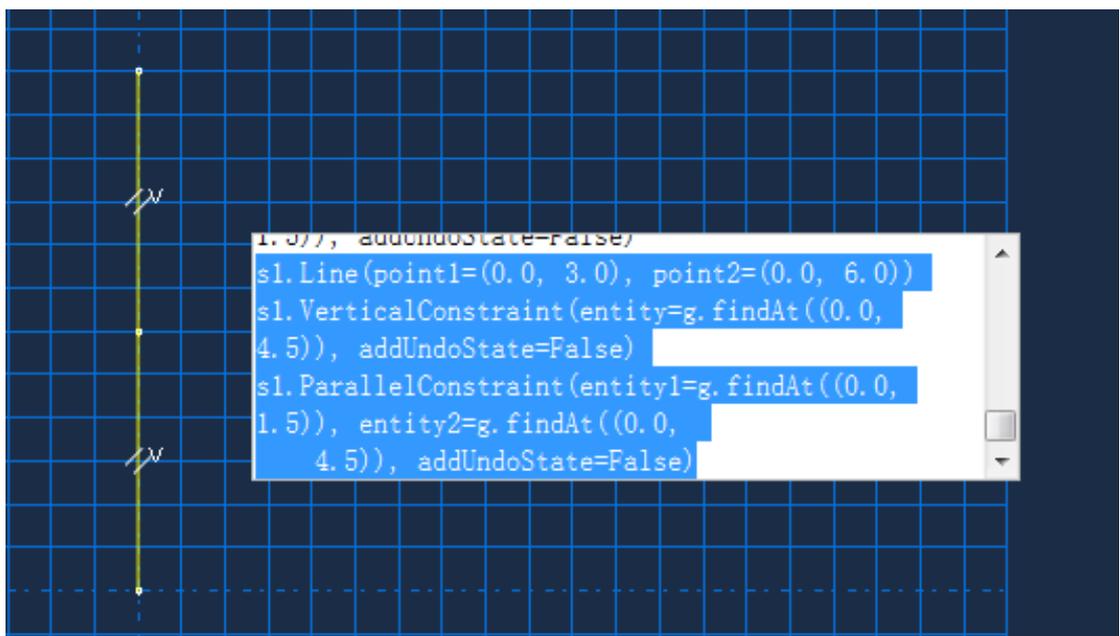
    #Draw Beams
    for noBeam in range(1, numFloor+1):
        xyCoordsBeamLeft=(0.0, floorHeight*noBeam)
        xyCoordsBeamRight=(bayWidth*numBay, floorHeight*noBeam)
        frameSketch.Line(point1=xyCoordsBeamLeft, point2=xyCoordsBeamRight)
```

主要编写了两个脚本文件：createSteelAnalysis.py（用于建立模型）与createXYPlot.py（用于进行后处理）。每个文件中根据需要分成了相关的函数。

脚本语言由Python语言书写。Python语言良好的阅读性可以大大增加脚本的可读性与可维护性，读代码就如同在进行英文阅读一样。

编写脚本对于提高效率是非常重要的，通过精心的编写，可以用很少的代码量完成大量重复的工作，大大节约了时间。

3、PyhonReader



由中国石油大学焦中良博士开发的 PythonReader，可以用来实时记录与 ABAQUS/CAE 操作对应的脚本命令，并能够获取所有的警告信息与错误信息，为初学者学习和创建脚本提供了极大方便。在当前相关帮助文档较少的情况下，是一个十分强大的补足。

七、重难点

1、数据类型的确定

在 ABAQUS 中经常需要设定各种集合，具体包括 Set、Surface 等。经查询可知，这两种结构的数据类型为 Sequence。Python 内置的 Sequence 包含 tuple（元组）、list（列表）、dictionary（字典）三种，不知的数据类型对应着不同的操作方式。因此如何正确确定集合的数据类型、正确选择对应的方法十分重要。

由于帮助文档比较简单，尚未对这一问题寻找到明确的说明。因此目前只能通过较原始的方法来进行操作。个人推测是否这是否是由 ABAQUS 派生出的一种新的数据类型。

2、元素选择

当需要点选几何元素时，需要将元素作为参数传入核心脚本。为了选取元素，常见的思路有：获取元素的坐标、获取元素的 ID、获取元素的索引值。

获取元素的 ID 被 ABAQUS 帮助文档明文禁止。索引值经常需要改变几何形状、ID 值会随之变化，也不方便。目前常用的方法是获取元素上某点的坐标，以此坐标值选择相应的元素。这样将会导致需要反复计算某些点的坐标，可读性差，不易维护。

八、总结与思考

1、编写插件的最终目的，是通过将一些特定的功能重新组合，更改操作界面，使之更符合用户操作习惯；

2、插件的功能宜小而精，通过规模扩大实现大的功能；如果有必要，应该

另行编写专门的插件管理平台，而不应在一个插件中赋予过多的功能，否则将大大降低插件的实用性，将在很大程度上与已有功能重复；

3、对于插件开发，功能的统筹规划是一项重要工作。由于插件的功能比较小巧，因此仔细研究用户习惯、寻找合适的切入点，对一个插件至关重要。在实现功能的基础上应当尽可能提高用户体验；

4、关于 ABAQUS 插件开发的相关文献比较少，目前学习成本比较高，大量编写尚不成熟。为了真正实现 GUI 界面、脚本与 Abaqus 功能三者的对接，尚需要进行一些探索。目前更常用的可能是用 CAE 生成脚本，将其参数化后用于解决大量的简单重复性工作或参数化分析。

九、课程建议

- 1、将课程改为十六周，有更多时间对大作业进行研究；
- 2、重点章节可以适当布置一些小作业。