

后处理、Beam单元

Contributor: 郑蕴哲

2019 年 6 月 15 日

1 后处理

由于并不需要我们单独画图而只需使用商用软件进行后处理，故后处理的任务简要可以认为将我们的计算结果转化为商用软件的输入形式。

首先是软件的选择。虽然tecplot并非开源（需要用盗版），但是在了解tecplot之后发现其对于有限元的后处理极为友好，输入格式简明。因而选择tecplot作为我们的后处理软件。

我们在STAPPP中后处理的思路如下：我们需要一个合适的.dat的后处理文件，而原有的.out输出文件由于格式已经固定，故唯一的办法是额外添加一个作为后处理的输出文件。我们在STAPPP中的main程序中创建一个新的.dat文件按tecplot格式存储数据。具体的写入过程通过一个 PostOutputter 实现。类似于Outputter的实现方式，我们在每个单元中定义了专门计算后处理信息的函数ElementPostInfo，然后在PostOutputter中调用，按照tecplot的格式输出。我们的后处理信息包括：位移后坐标、应力不变量、Mises应力以及6个应力分量。具体代码请查看主程序文件。我们由下图可以看到我们的后处理最终效果和abaqus是十分相近的。

2 Beam单元

由于Beam单元与Bar单元十分类似，因此在Beam单元实现的过程中很多地方都借鉴了Bar单元的思路和细节。

首先Beam的刚度阵如下图所示

对于Beam单元，类似Bar单元，我们可以写出其位移与外力形式。

$$d^e = \begin{pmatrix} u_{x1} & u_{y1} & u_{z1} & \theta_{x1} & \theta_{y1} & \theta_{z1} & u_{x2} & u_{y2} & u_{z2} & \theta_{x2} & \theta_{y2} & \theta_{y3} \end{pmatrix}^T$$

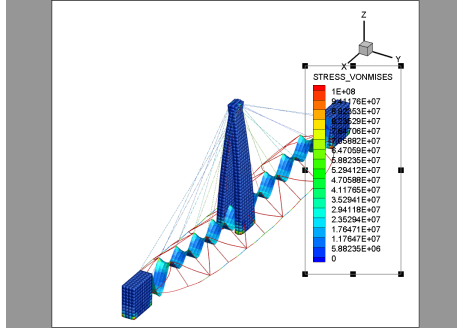


图 1: tecplot后处理

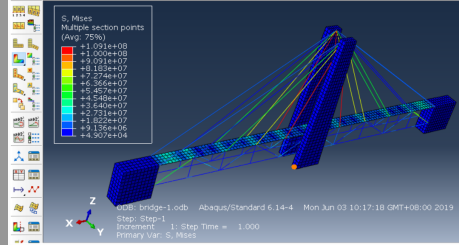


图 2: abaqus后处理

$$\begin{bmatrix}
 \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & \frac{12EJ_x}{l^3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & \frac{12EJ_y}{l^3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & \frac{GJ_z}{l} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & -\frac{6EJ_x}{l^2} & 0 & \frac{4EJ_x}{l} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & \frac{6EJ_x}{l^2} & 0 & 0 & 0 & 0 & \frac{4EJ_x}{l} & 0 & 0 & 0 & 0 & 0 & 0 \\
 -\frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 \\
 0 & -\frac{12EJ_x}{l^3} & 0 & 0 & 0 & 0 & -\frac{6EJ_x}{l^2} & 0 & \frac{12EJ_x}{l^3} & 0 & 0 & 0 & 0 \\
 0 & 0 & -\frac{12EJ_y}{l^3} & 0 & \frac{6EJ_y}{l^2} & 0 & 0 & 0 & 0 & \frac{12EJ_y}{l^3} & 0 & 0 & 0 \\
 0 & 0 & 0 & -\frac{GJ_z}{l} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{GJ_z}{l} & 0 & 0 \\
 0 & 0 & \frac{6EJ_x}{l^2} & 0 & \frac{2EJ_x}{l} & 0 & 0 & 0 & \frac{6EJ_x}{l^2} & 0 & 0 & \frac{4EJ_x}{l} & 0 \\
 0 & \frac{6EJ_x}{l^2} & 0 & 0 & 0 & \frac{2EJ_x}{l} & 0 & -\frac{6EJ_x}{l^2} & 0 & 0 & 0 & 0 & \frac{4EJ_x}{l}
 \end{bmatrix}
 \quad \text{(对称)}$$

图 3: Beam刚度阵

$$f^e = \left(F_{x1} \quad F_{y1} \quad F_{z1} \quad M_{x1} \quad M_{y1} \quad M_{z1} \quad F_{x2} \quad F_{y2} \quad F_{z2} \quad M_{x2} \quad M_{y2} \quad M_{y3} \right)^T$$

因而我们得到了刚度方程实现的具体细节。当然为了能够在三维计算，我们还需要对杆进行一个旋转变换。

材料的输入属性包括弹性模量和泊松比。由于很多情况下梁是空心的，因此我们除了长、宽外还允许输出内部空心矩阵的参数。具体请参照源代码。

3 本人所承担的小组任务

1. 4Q单元的整合与调试（最终小组4Q使用的本人的代码）
2. Beam单元的设计
3. 后处理的实现
4. 桥梁竞赛的桥梁模型设计与参数优化