# V 3.0 Liner 优化版

## 更新说明

相比与2.0版，3.0版优化了一个问题：如何在考虑Liner单元的创建对于物理模型的影响的情况下导出Liner单元！

在2.0版中，可以直接从Hypermesh中导出Liner单元的，在不考虑地下连续墙的创建过程所产生的应力位移的情况下，可以直接在已经创建好墙体的模型下进行地应力的计算，然后将位移场清零。但是如果要考虑地下连续墙施工所产生的位移，则必须先在无墙体而且土体节点耦合的情况下进行地应力计算，再将位移清零，然后再将Liner单元创建以模拟墙体施工，在这个过程中，会遇到一个问题，对于几何形态较复杂的Liner的建模，如何对Liner进行定位，又如何将无Liner区域的节点进行耦合。

在2.0版中，解决了Liner单元的定位问题，同时，经研究发现，在Flac3d中创建好Liner后，即使Liner单元两侧的Zone的重合节点被merge起来，此Liner依然保持与两个的Zone之前的接触关系，而且接触属性不变，这一特性使得在如下操作可以正常进行：



在上面的流程中，第5步是在网格建模中最麻烦的问题，因为在分割Zone时，Generate Seperate Range Group groupA 命令会将指定Rang的Zone单元的所有边界表面进行分割。在一般的模型中，Liner单元的位置比较容易定位（比如在一个柱形基坑土体区域的四周），此时，只需要用 Generate Merge 指定将墙底所在平面区域进行节点耦合即可，此区域可以很方便地进行限定。但是，当Liner单元的位置比较复杂（可以自行想像极端复杂的情况）时，在将这些Liner所在的Zone进行seperate以暴露出对应的Surface之后，如何进行Liner的定位，以及如何再将无Liner的surface所对应的gridpoint进行重新耦合，都成了难以解决的问题。

在2.0版中，解决了第一个问题，即如何进行复杂位置的Liner单元的创建与定位，但是没有解决无Liner的Surface face上的gridpoint的重新耦合的问题，因此，3.0版对此进行了优化，并解决了此问题。其解决的基本思路为：

1. 在Hypermesh中通过Find Face，在实体单元表面创建出对应的S3或者S4单元，记住千万不要将这些壳单元强行Split为三角形单元，除非此壳单元是用来在Flac3D中模型Shell单元的；
2. 将这些壳单元中，真正要作为Liner单元的分配到Liner特征名的Component中，而将其他的壳单元删除；
3. 将要进行耦合的节点集合通过Hypermesh中的Node Set保存起来，并且确保其名称以LM开头，比如lmBottom、LM\_Bottom或Lm-Bottom；
4. （扩展，可以考虑通过 Generate Separate Face <Range Group groupA Group groupB . . . > 来根据真实Liner所在的壳单元的位置来直接进行指定Face的分割，而不通过 Generate Seperate Range Group groupA 将指定的 groupA的所有表面进行分割然后再补上不需要分割的部分。）
5. 在将inp文件进行转换时，将Liner单元分开导出到一个单独的文本中。事实上，可以将所有不同类型的结构单元都分开导出到不同的.dat文本文件中，在Flac3D中需要时，再将对应的结构通过 call命令导入到模型中；
6. 在进行模型转换时，对于LM中的节点，会根据节点编号所对应的坐标，生成类似 Generate Merge 0.001 Range x= (23.73, 23.78) y =( -0.01, 0.01) z= ( 19.65, 19.67) 的命令，以进行gridpoint的耦合；
7. 在Flac3D中创建Liner时，先将Liner所依附的Zone Group进行分割，然后通过call 命令执行Liner单元的创建，最后再通过上面的 generate merge 命令，将无Liner的Surface face所对应的gridpoint进行耦合，以此实现复杂形态的Liner单元的建模。

## Hypermesh to Flac3d 转换规则

|  |  |
| --- | --- |
| Flac3d中的单元类型 | 在Hypermesh文件中的判断规则 |
| Beam（两个节点） | 单元类型为“B31”或“B21”，而且Component名称以“beam”开头（不区分大小写） |
| Pile（两个节点） | 单元类型为“B31”或“B21”，而且Component名称以“pile”开头（不区分大小写） |
| 六面体8节点Zone单元 | 单元类型指定为“C3D8”开头，比如“C3D8R” |
| 五面体6节点（三棱柱）Zone单元 | 单元类型指定为“C3D6”开头，比如“C3D6R” |
| 四面体4节点Zone单元 | 单元类型指定为“C3D4”开头，比如“C3D4R” |
| Shell（三角形单元） | 单元类型以“S3”开头，而且Component名称以“shell”开头（不区分大小写） |
| Liner（三角形单元） | 单元类型以“S3”或者“S4”开头，而且Component名称以“liner-附着组”开头（不区分大小写），具体详见章节“Liner 单元的转换说明”  注意Liner单元周围的Zone的网格尺寸的值不能小球0.001（当模型以m为单位时，即不能小于1 mm）。 |
| 要耦合的节点 | 保存Node Set中，其名称以LM开头。 |
| \* B31 and B21 are both 2-node linear beams，where B21 is used in a plane with 3 degrees of freedom:1, 2, 6, while B31 is used in a space, with 6 degrees of freedom: 1, 2, 3, 4, 5, 6. | |

注意：

* Hypermesh中一个Component 里面可以有多种要输出的单元类型，在Hypermesh转换为.inp格式时，Hypermesh会自动将同一个Component中的不同单元类型进行二次分组。
* 对于Flac3d中的Shell单元，在Hypermesh中必须用S3来进行模拟，因为在导入Flac3d中时， 可以直接用三节点编号的形式创建 ShellSel；
* 对于Flac3d中的Liner单元，在Hypermesh中必须保证与相邻的Zone单元的表面形状相同，因为在导入Flac3d中时，Liner单元是通过搜索Zone单元的表面的形心来创建的。因此，Hypermesh中的Liner单元，既有可能是S4，也有可能是S3（当对应的Zone为四面锥体时，其 face为三角形）。

## Liner 单元的转换说明

### 配合 Set Browser 使用

在Flac3D中，Liner单元如果要在两个侧面都与Zone之间创建柔性接触，只能通过

Sel Liner id 1 em Group ex1 Range x= (23.73, 23.78) y =( -0.01, 0.01) z= ( 19.65, 19.67)

所以，如果要创建此类型的Liner，就必须要先有一个用于附着的Group对象，然后Flac3D会通过搜索此Group中所有边界zone的边界Face，只要此Face的形心位于上面语句中的Range所指定范围中，Flac3D就会根据此Face创建出一个或者多个三角形的 Liner 单元。

此Group对象是通过Hypermesh中的Set（通过Tool>Set Browser打开Sets面板）来实现的。即将要用来进行附着的Zone组合为多个Element Set，然后通过 Liner 的 Componet 的名称来指定要附着到哪一个 Element Set中。在将Element Set从inp文件中转换到Flac3D文件中去时，会将其分配一个单独的Group中，而且此Group所在的Slot编号会大于1，因为Hypermesh中的Component中的Zone单元也会被放置到Group中，而在默认情况下，Group如果不显式指定Slot的话，会自动分配到Slot 1中。

### Element Set 与对应的 Liner Component的命名规范

Element Set 与对应的 Liner Component的命名规范如下：

* Element Set必须以“LG”开头，而且名称中不能包含“-”。比如“LG”、“lgzone”、“LG\_Zone”都是可以的；
* Liner Component的名称必须以“Liner-附着组名”开头，更多的名称信息可以在组名后用“-”进行分隔。比如当其要附着到组LG中时，“Liner-LGLeft”、“Liner-LG-Left”都是可以的，但是“Liner-LGLeft”会将此Liner单元附着到组“LGLeft”中，但是如果Flac3D中并没有创建一个组“LGLeft”的话，自然是会出现异常的。

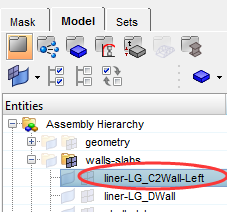
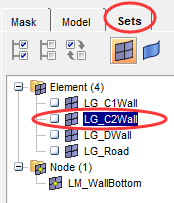
 

图1 Hypermesh中与Liner 的创建相关的Set 与 Component 的命名示例

### inp 导出 S4 的规则

Hypermesh中S3或者S4单元的输出到inp文件中的节点顺序（通过 Hypermesh 中的 2D面板 > Edit element > Create来进行创建单元的测试）：

1. 绕边界环线进行编号；

2. 如果创建S4单元时是按边界环线点击节点，则inp中的节点顺序与创建时点击的节点顺序一致；

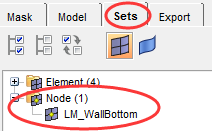
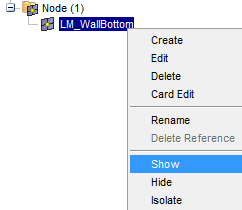
3. 如果创建S4单元时并不是按边界环线点击节点，在Hypermesh中自动将第3、4个节点顺序进行调整，以将其重排为边界环路的顺序，而不改变前面的两个节点顺序。

4. 对于异型的S4单元（比如箭头形这种有凹角的），不论节点点击顺序如何，在Hypermesh中都会确保其在inp中的顺序形成一个边界环路,但是要注意，这种四边形网格是有错误的，在计算时肯定会出现异常。

## 使用说明

### Hypermesh 建模与导出

1. 在Hypermesh中绘制好土体与结构（桩、支撑、地下连续墙）的网格。注意整个Hypermesh模型中，Liner单元处的实体单元之间是否分隔，并无影响，因为在Flac3D中都要进行一次分割，所以为了减少导入时的节点数目，推荐在Hypermesh中将包含土体与结构的整个模型进行耦合；
2. Hypermesh中一个Component 里面可以有多种要输出的单元类型，在Hypermesh转换为.inp格式时，Hypermesh会自动将同一个Component中的不同单元类型进行二次分组。
3. 对于Flac3d中的Shell单元，在Hypermesh中必须用S3来进行模拟，因为在导入Flac3d中时， 可以直接用三节点编号的形式创建 ShellSel；
4. 对于Flac3d中的Liner单元，在Hypermesh中必须保证与相邻的Zone单元的表面形状相同，因为在导入Flac3d中时，Liner单元是通过搜索Zone单元的表面的形心来创建的。因此，Hypermesh中的Liner单元，既有可能是S4，也有可能是S3（当对应的Zone为四面锥体时，其 face为三角形）。
5. 对于要用来进行Flac3d中的节点耦合命令的节点，要将其放置在Hypermesh中的Node Set中，并保证命名以LM开头，比如“lm”、“lmBottom”、“LM-bottom”都是可以的；

注意：当节点被添加到Node Set中后，并不会在模型中创建新的节点，而只是索引了节点编号。在进行模型转换时，此类节点是在zones.inp中进行处理的，所以请保证在导出土体时这一部分节点是显示出来的。为了保证这一点，只需要确保LM Node Set中的节点编号对应的节点都已经包含在土体单元的节点中。但有时这一点容易被忽略，比如Node Set 是通过索引结构单元的节点创建的，而结构单元的节点并未与土体单元的节点耦合。如果不想太复杂，最稳妥的方法是在导出zones.inp时，将LM Node Set中的节点通过Show菜单显示出来。

另外，测试中发现，Generate Merge 命令在Flac3D中执行较慢，因此将所有的节点的耦合都转换为一条Generate Merge 命令非常不可取，因此，推荐的方法是：对于容易表达的几何区域，直接通过Flac3D中的命令进行Range级别的节点耦合，对于复杂的几何区域，则通过本程序转换出来的命令进行节点级别的耦合。此理论也同样适用于创建Liner时是要通过Flac3D中的命令还是通过转换出来的命令的选择情况。

1. 在模型导出时，先将三维土体单元导出为Abaqus的inp文件；再将一维与二维的结构单元导出为Abaqus的inp文件。注意导出时只将要导出的单元显示出来，然后选择“Displayed模式”；

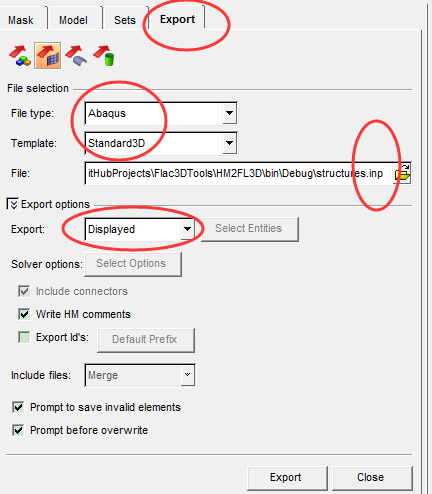


图 1-1 将Hypermesh导出到 inp 文件时的选项

### Hm2Flac3D 使用说明

Hm2Flac3D 专注做一件事：将Hypermesh中的可视化信息转换并拆解为Flac3D代码。

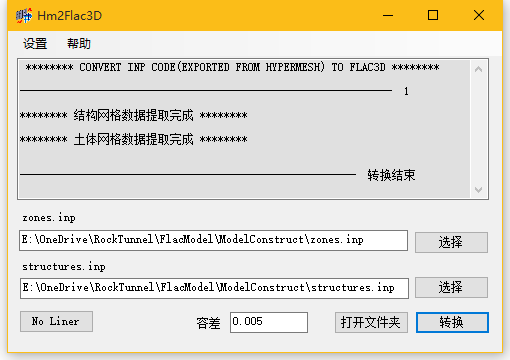
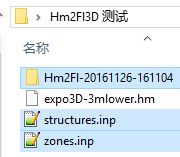
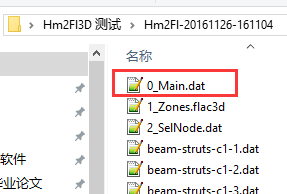


图 1-2 Hm2Flac3D操作界面

1. 分别将上面两个inp文件用本Hm2Flac3D程序进行模型转换，最终的转换结果会全部保存在一个新的文件夹中，比如“”，在此文件夹中，有一个0\_Main.dat文件，里面包含了对于每一个拆解部分的ImpGrid或者Call命令。

### Flac3D 建模说明

1. 整个Hypermesh模型已经转换并拆解为不同的操作步骤，在Flac3d中建模时，只要参考0\_Main.dat文件中的语句，在需要的位置将对应的操作通过Call命令添加到Flac3D中即可；
2. 在Flac3D中利用Call进行结构单元的创建之前，必须先通过“Call 2\_SelNode.dat”将结构单元的节点创建出来，否则后面创建结构单元时显然会出错；
3. 对于Join的问题，对于Beam与Pile单元之间的Join，如果在Hypermesh中建模时，它们的相交节点就已经耦合了，那么在Flac3D中就不需要也不能再用Join将二者连接起来，因为它们在相交处是共用一个节点，如果强行Join，则在Solve时会出现“Recursive rigid join …”的报错；另一方面，涉及到与Liner相关的Join时，不论在Hypermesh中，某结构单元是否与Liner单元共节点，在Flac3D中都必须重新设置它们之间的Join，因为Flac3D中的Liner单元是通过“Sel Liner id 16 em group LGC2Wall Range …”生成的，所以生成的Liner单元的节点编号已经不是Hypermesh中的Liner单元的节点编号了。
4. Enjoy it!

## 注意事项

### 将zones单元与结构单元分开导出

在Hypermesh中进行网格导出时，建议将Zone单元与结构单元分开来导出和转换。如果将土体单元与结构单元同时导出并转换时，本程序并不会报错，导出的网格也不会有问题，\_\_但是，zones.flac3d文件中的grid数目可能会多于所有的zone单元所用到的grid，同样地，structures.dat文件中的node的数目可能会多于所有的结构单元所用到的node。另外，如果将土体单元与结构单元同时导出，在将模型导入到Flac中时也看不出很明显的问题，但是在此模型中创建Liner单元时，可能会很慢，最终计算结果也会有问题\_\_。

### 在Hypermesh中调整节点的编号（推理未测试）

在通过如下语句生成Liner时，Flac3D会自动为生成出来的Liner单元的每一个Node分配节点编号。为了避免Node的编号的冲突，在分配节点时会对当前模型中已经存在的nodes集合的编号进行搜索（猜测会在内存中保存一个SortedSet<UInt64> 的集合）。

Sel Liner id 1 em Group ex1 Range x= (23.73, 23.78) y =( -0.01, 0.01) z= ( 19.65, 19.67)

比如在生成Liner之前，Flac3D模型中的节点（不论结构单元还是实体单元）已经占据了编号1~5000，则在“Sel Liner …”时，就必须要从1开始搜索可用的节点编号，这样的遍历与Contains()的判断会降低一定的计算效率，而且已有的节点越多，这种搜索就越费时。

最后的结论就是：建议用户自行在Hypermesh中去调整网格节点的编号，使Hypermesh中的 nodes.MinmumId > nodes.Count。

### C3D8类型的单元

对于六面体八节点单元，其在inp文件中的最后一个节点是写在第二行的。如果从hypermesh中导出的inp文件中。C3D8类型的最后一个节点不是写在第二行，则此程序不会导出这些单元。



### 单元法向

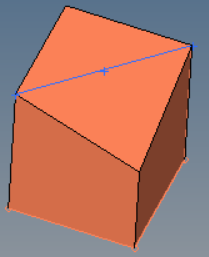
对于三维网格单元，其在Hypermesh中的法向（Normal）并不要求一致。

### Merge的问题

对于 Hm2Flac3D 或者 Ansys 导出的网格，虽然在Hypermesh中已经用Equivilance进行了节点合并，但是在Flac3D中进行Solve时，还是有极小的可能会报出“Zero stiffness in grid-point 18545” 这种错误。此时可能并不是土体单元未被赋上材料属性，而是需要对初始网格在Flac3D中再进行一次合并（用 gen merge 0.1），然后就可以正常计算了。

### 非平面四边形的处理

在Hm2Flac3D中，计算四边形的形心的算法将其拆开成两个三角形，再计算其各自形心并按面积权重组合起来的（这也是常规的计算非共面四边形形心的算法），对于一个六面体单元，有可能出现其某些由四个点组成的面并不共面的情况，这种情况下，inp文件中还是以C3D8类型存储，也可以正常导出为flac3d的W8类型单元，但是，对于依附于其上的Liner单元，由于Hm2Flac3D中是通过计算其所在面的形心点，再将此点扩展为一个小立方体来作为创建Liner的搜索基础，而此时的形心点并不在这个异形四边形上，因此小立方体的大小必须相应扩大，否则如果此小立方体并不能与此异形面相交，则不会创建出对应的Liner单元。



为了解决此问题，可以将图 1-2中的“容差”值设置得大一些（比如从0.005调大为0.25），其效果就是在转换生成的“Liner-LG\_CircularCube.dat”文件中，用来生成每一个Liner的搜索范围将变大，比如：Sel Liner id 2 em group LG\_CircularCube Range x=(6917.11568,6917.36568) y=(4023.39637,4023.64637) z=(1965.73297,1965.98297)，即可使得小立方体与异形四边形在空间上相交。不同的容差值效果比较如下：

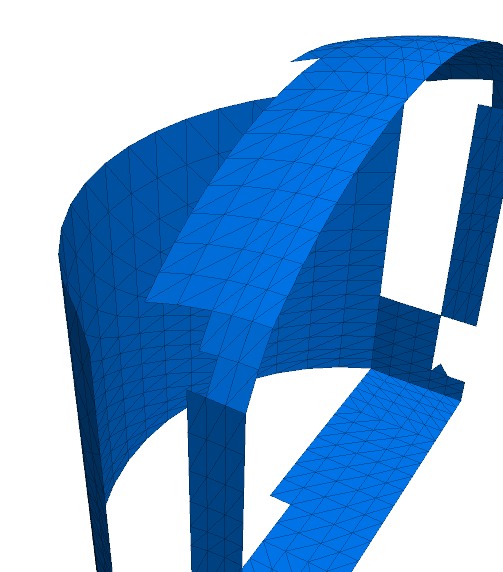
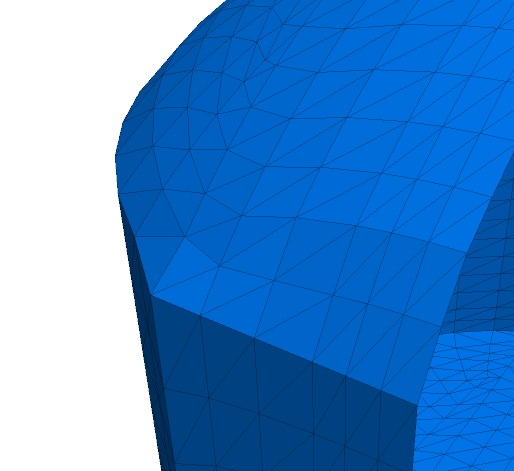
 

图 1-2 Hm2Flac3D转换时容差为0.005与0.25时生成Liner的效果

### 不同类型二维单元附着于同一Element Set时的分组

S3、S4

# V 2.0 Liner 模式

## 更新说明：

相比与1.0版，2.0版优化了一个问题：如何将考虑双面接触的Liner单元通过Hypermesh导入Flac3D！

在1.0版中，可以在Hypermesh中创建Zone、Pile、Beam与Shell单元，这些单元的定义都可以通过Hypermesh中导出的inp文件中直接读取，但是，对于Liner单元，由于通过LinerSel命令与三节点编号创建出的Liner单元只有一个面与Zone之间有接触关系，并不能满足需要。所以，在2.0版本中，通过Hypermesh中的S3或者S4来模拟Liner单元，然后通过其节点的坐标计算出S3或S4的形心位置，最后，在Flac3D中通过“Sel Liner id 1 em Group ex1 Range x= (23.73, 23.78) y =( -0.01, 0.01) z= ( 19.65, 19.67)” 的方式实现了Liner单元的Hypermesh可视化模拟。

## Hypermesh to Flac3d 转换规则

注意：一个Component中可以有多种要输出的单元类型，在Hypermesh转换为Inp格式时，Hypermesh会自动将同一个Component中的不同单元类型进行二次分组。

|  |  |
| --- | --- |
| Flac3d中的单元类型 | 在Hypermesh文件中的判断规则 |
| Beam（两个节点） | 单元类型为“B31”或“B21”，而且Component名称以“beam”开头（不区分大小写） |
| Pile（两个节点） | 单元类型为“B31”或“B21”，而且Component名称以“pile”开头（不区分大小写） |
| 六面体8节点Zone单元 | 单元类型指定为“C3D8”开头，比如“C3D8R” |
| 五面体6节点（三棱柱）Zone单元 | 单元类型指定为“C3D6”开头，比如“C3D6R” |
| 四面体4节点Zone单元 | 单元类型指定为“C3D4”开头，比如“C3D4R” |
| Shell（三角形单元） | 单元类型以“S3”开头，而且Component名称以“shell”开头（不区分大小写） |
| Liner（三角形单元） | 单元类型以“S3”或者“S4”开头，而且Component名称以“liner-附着组”开头（不区分大小写）  注意Liner单元周围的Zone的网格尺寸的值不能小球0.001（当模型以m为单位时，即不能小于1 mm）。 |
| \* B31 and B21 are both 2-node linear beams，where B21 is used in a plane with 3 degrees of freedom:1, 2, 6, while B31 is used in a space, with 6 degrees of freedom: 1, 2, 3, 4, 5, 6. | |

## Liner 单元的转换说明

### 配合 Set Browser 使用

在Flac3D中，Liner单元如果要在两个侧面都与Zone之间创建柔性接触，只能通过

Sel Liner id 1 em Group ex1 Range x= (23.73, 23.78) y =( -0.01, 0.01) z= ( 19.65, 19.67)

所以，如果要创建此类型的Liner，就必须要先有一个用于附着的Group对象，然后Flac3D会通过搜索此Group中所有边界zone的边界Face，只要此Face的形心位于上面语句中的Range所指定范围中，Flac3D就会根据此Face创建出一个或者多个三角形的 Liner 单元。

此Group对象是通过Hypermesh中的Set来实现的。即将要用来进行附着的Zone组合为多个Element Set，然后通过 Liner 的 Componet 的名称来指定要附着到哪一个 Element Set中。在将Element Set从inp文件中转换到Flac3D文件中去时，会将其分配一个单独的Group中，而且此Group所在的Slot编号会大于1，因为Hypermesh中的Component中的Zone单元也会被放置到Group中，而在默认情况下，Group如果不显式指定Slot的话，会自动分配到Slot 1中。

### Element Set 与对应的 Liner Component的命名规范

Element Set 与对应的 Liner Component的命名规范如下：

* Element Set必须以“GLiner”开头，而且名称中不能包含“-”。比如“GLiner”、“GLiner\_Zone”都是可以的；
* Liner Component的名称必须以“Liner-附着组名”开头，更多的名称信息可以在组名后用“-”进行分隔。比如当其要附着到组GLiner中时，“Liner-GLinerLeft”、“Liner-GLiner-Left”都是可以的，但是“Liner-GLinerLeft”会将此Liner单元附着到组“GLinerLeft”中，但是如果Flac3D中并没有创建一个组“GLinerLeft”的话，自然是会出现异常的。

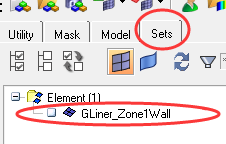
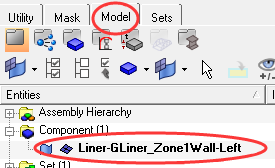
 

图1 Hypermesh中与Liner 的创建相关的Set 与 Component 的命名示例

### inp 导出 S4 的规则

Hypermesh中S3或者S4单元的输出到inp文件中的节点顺序（通过 Hypermesh 中的 2D面板 > Edit element > Create来进行创建单元的测试）：

1. 绕边界环线进行编号；

2. 如果创建S4单元时是按边界环线点击节点，则inp中的节点顺序与创建时点击的节点顺序一致；

3. 如果创建S4单元时并不是按边界环线点击节点，在Hypermesh中自动将第3、4个节点顺序进行调整，以将其重排为边界环路的顺序，而不改变前面的两个节点顺序。

4. 对于异型的S4单元（比如箭头形这种有凹角的），不论节点点击顺序如何，在Hypermesh中都会确保其在inp中的顺序形成一个边界环路,但是要注意，这种四边形网格是有错误的，在计算时肯定会出现异常。

## 使用说明

### 使用流程

1. 在Hypermesh中绘制好土体与结构（桩、支撑、地下连续墙）的网格。注意在有地下连续墙的位置，一定要将墙体左右的Zone单元用Hypermesh中Detach分隔开来；
2. 先将三维土体单元导出为Abaqus的inp文件；再将一维与二维的结构单元导出为Abaqus的inp文件。注意导出时只将要导出的单元显示出来，然后选择“Displayed模式”；
3. 分别将上面两个inp文件用本Hm2Flac3D程序将网格转换为Zones.Flac3D的土体网格与structures.dat的结构单元网格，生成的文本文件会保存在与对应的inp文件相同的文件夹内。
4. 在Flac3d中，先用“import zones.Flac3D”导入土体网格，再用“Call structures.dat”导入结构单元，注意导入的先后顺序；
5. 开始在Flac3D中进行设置与计算。Enjoy it!

### 注意事项

#### 将zones单元与结构单元分开导出

在Hypermesh中进行网格导出时，建议将Zone单元与结构单元分开来导出和转换。如果将土体单元与结构单元同时导出并转换时，本程序并不会报错，导出的网格也不会有问题，\_\_但是，zones.flac3d文件中的grid数目可能会多于所有的zone单元所用到的grid，同样地，structures.dat文件中的node的数目可能会多于所有的结构单元所用到的node。另外，如果将土体单元与结构单元同时导出，在将模型导入到Flac中时也看不出很明显的问题，但是在此模型中创建Liner单元时，可能会很慢，最终计算结果也会有问题\_\_。

#### 在Hypermesh中调整节点的编号（推理未测试）

在通过如下语句生成Liner时，Flac3D会自动为生成出来的Liner单元的每一个Node分配节点编号。为了避免Node的编号的冲突，在分配节点时会对当前模型中已经存在的nodes集合的编号进行搜索（猜测会在内存中保存一个SortedSet<UInt64> 的集合）。

Sel Liner id 1 em Group ex1 Range x= (23.73, 23.78) y =( -0.01, 0.01) z= ( 19.65, 19.67)

比如在生成Liner之前，Flac3D模型中的节点（不论结构单元还是实体单元）已经占据了编号1~5000，则在“Sel Liner …”时，就必须要从1开始搜索可用的节点编号，这样的遍历与Contains()的判断会降低一定的计算效率，而且已有的节点越多，这种搜索就越费时。

最后的结论就是：建议用户自行在Hypermesh中去调整网格节点的编号，使Hypermesh中的 nodes.MinmumId > nodes.Count。

#### C3D8类型的单元

对于六面体八节点单元，其在inp文件中的最后一个节点是写在第二行的。如果从hypermesh中导出的inp文件中。C3D8类型的最后一个节点不是写在第二行，则此程序不会导出这些单元。



#### 单元法向

对于三维网格单元，其在Hypermesh中的法向（Normal）并不要求一致。

#### Merge的问题

对于 Hm2Flac3D 或者 Ansys 导出的网格，虽然在Hypermesh中已经用Equivilance进行了节点合并，但是在Flac3D中进行Solve时，还是有极小的可能会报出“Zero stiffness in grid-point 18545” 这种错误。此时可能并不是土体单元未被赋上材料属性，而是需要对初始网格在Flac3D中再进行一次合并（用 gen merge 0.1），然后就可以正常计算了。

# V 1.0 无 Liner 模式

## Hypermesh to Flac3d 转换规则

|  |  |
| --- | --- |
| Flac3d中的单元类型 | 在Hypermesh文件中的判断规则 |
| Beam（两个节点） | 单元类型为“B31”或“B21”，而且Component名称以“beam”开头（不区分大小写） |
| Pile（两个节点） | 单元类型为“B31”或“B21”，而且Component名称以“pile”开头（不区分大小写） |
| 六面体8节点Zone单元 | 单元类型指定为“C3D8”开头，比如“C3D8R” |
| 五面体6节点（三棱柱）Zone单元 | 单元类型指定为“C3D6”开头，比如“C3D6R” |
| 四面体4节点Zone单元 | 单元类型指定为“C3D4”开头，比如“C3D4R” |
| Shell（三角形单元） | 单元类型以“S3”开头，而且Component名称以“shell”开头（不区分大小写） |
| ~~Liner（三角形单元）~~ | ~~单元类型以“S3”开头，而且Component名称以“liner”开头（不区分大小写）~~ |
| \* B31 and B21 are both 2-node linear beams，where B21 is used in a plane with 3 degrees of freedom:1, 2, 6, while B31 is used in a space, with 6 degrees of freedom: 1, 2, 3, 4, 5, 6. | |

## 使用说明

### 使用流程

1. 在Hypermesh中绘制好土体与结构的网格，不用绘制地下连续墙；

2. 先将三维土体单元导出为Abaqus的inp文件；再将一维与二维的结构单元导出为Abaqus的inp文件；

3. 分别将上面两个inp文件用程序“”将网格转换为Zones.Flac3D的土体网格与structures.dat的结构单元网格；

4. 在Flac3d中，先用“im zones.Flac3D”导入土体网格，再用“Call structures.dat”导入结构单元，注意导入的先后顺序；

5. 开始在Flac3D中进行设置与计算。Enjoy it!

### 注意事项

#### C3D8类型的单元

对于六面体八节点单元，其在inp文件中的最后一个节点是写在第二行的。如果从hypermesh中导出的inp文件中。C3D8类型类型的最后一个节点不是写在第二行，则此程序不会导出这此单元。

