OF中internal surface处理问题

OF中没有internal interface边界条件，ICEM网格输出前，需要显性切开所设置的internal wall，然后再转化为OF网格（fluent3DToFoam），再利用stitchMesh命令，缝合internal interface， 最后再boundary，0\文件 中删除相应internal surface。而没有显性切开的internal wall都默认设置为solid wall。

Cmd: stitchMesh -perfect -overwrite internal-interface1 internal-interface2

注意命令执行条件，

* boundary中internal-interface1/ internal-interface2 设置为patch
* 0/文件 边界条件设置完整

作者：Johnx Wang  
链接：https://www.zhihu.com/question/67574409/answer/254216621  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

以下是笔者学习OF的历程精炼总结以飨后来者，提供学习路径，快速知识回顾，关键点查阅等功能。OF学习方法，知识点繁多，之间关联性强，一次性很难掌握全貌，只有反复来回阅读查考，才能触类旁通。在某一难点死磕，是不智的。

将继续更新。。。

一 OpenFoam（OF）case工作流程

1. 找到软件附带的，和自己的求解问题相关的tutorial文件（$FOAM\_TUTORIALS），复制到自己的文件夹（OF推荐在$FOAM\_RUN/）

2. 打开后，学习OF的文件结构。constant/ system/ ‘time/ (初始情况下一般是 0/ )

3. constant/ 包含

transportProperties文件 – 存储运动粘度系数nu的数据 （以cavity为例）

polyMesh文件 – 存储网格数据（运行blockMesh后生成）

4. system/包含

controlDict文件 – 存储模拟控制相关数据，如模拟时间的起始，终止，数据保存方式等

fvSchemes文件 – 存储模拟所用的方程的离散方式的数据，如transient项的离散方式，gradient项的离散方式，divergence项的离散方式，laplacian项 的离散方式，interpolation的方式等（笔者提醒自己：需优化）

fvSolution文件 – 存储模拟求解计算精确度，帮助收敛的相关数据，如残差设定，松弛系数设定，模拟所用算法（PISO, SIMPLE）的控制参数

5. 0/包含

P文件，U文件 – 存储模拟所求解的变量在求解域的初始值和边界值的数据

6. 全部设定完成后，运行程序icoFoam，开始模拟计算

二 OpenFoam（OF）模拟计算程序运行机理

1. 程序入口 – \*Foam.c（目录“applications/solvers/”里）可找到main函数开始程序

2. #include "createFields.H" – 读取变量场数据，某个程序读取了0/p, 0/U里的数据

3. 待完成。。

三 OpenFoam（OF）编译机理

1. 用户自定义算法程序的源代码编译

算法程序的源代码依赖于所使用的头文件和库文件，头文件主要负责class的声明，包括class的变量和函数的声明，而class的变量和函数具体实现内容则放在相应的 \*.c文件中，\*.c  
文件已经预先编译成了\*.so库文件并保存起来。所以算法程序需知道头文件和库文件的目录，在编译时预处理头文件，在运行时动态链接库文件，则程序才可正常运行。

OF程序编译借助wmake工具，make/options中存放头文件（EXE\_INC=）和库文件的（EXE\_LIB=）目录; make/file中存放算法程序的源代码 \*.c和 编译后可执行文件存放目录（EXE=）

所需知识点：C++源代码的构建过程，动态链接

2. 用户自定义共享库文件的源代码编译

用户自己写的库文件源代码，如新建边界条件，需要编译成库文件机器码 \*.so以供算法程序动态链接。其方法与算法程序的编译相似，只需将EXE\_INC 替换为 LIB\_INC, EXE\_LIB 替换为LIB\_LIB，编译完成后会生成InInclude。然后在etc/controlDict文件中加入 libs(“\*.so”); 即可。

四 OF环境变量列表：

$FOAM\_RUN –

$FOAM\_TUTORIALS -