**程序耦合开发文档**

目录

[2 耦合工作 3](#_Toc500920823)

[3 程序开发日志 5](#_Toc500920824)

[在daisy输入卡feedback中增加is\_imp关键字； 6](#_Toc500920825)

[空间高度网格的对应； 6](#_Toc500920826)

[首先解决网格的耦合问题 7](#_Toc500920827)

[写imp\_Pre\_process类用以imp模块的前处理； 8](#_Toc500920828)

# 耦合工作

几何参数对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **daisy** | | | **imp** | | |
|  | | | 修改前 | 修改后 |  |
| **输入卡** | **Pitch** | 节距 | **AcrossFlat** | pitch | 输入卡 |
| **输入卡** | **Rod** | 燃料棒半径 | **/** | Rod | 输入卡 |
| **输入卡** | **Cladth** | 包壳厚度 | **ShellThick** | **Cladth** | 输入卡 |
| **输入卡** | **Bond** | 气隙厚度 | **GasGap** | Bond | 输入卡 |
| **输入卡** | **Hole** | 内部小孔孔径 | **/** | / 不需要 | 输入卡 |
| **输入卡** | **N\_pin** | 通道数 | **/** | N\_pin | 输入卡 |
| **输入卡** | **N\_fuelpin** | 燃料通道数 | **N\_pin** | **N\_fuelpin** | 输入卡 |
| **Set计算** | **pellet** |  | **rFuel** | **Pellet** |  |
| **Set计算** | **area** | 芯块横截面积 | **/** | Area |  |
| **Set计算** | **flowarea** |  | **/** | / hydrau |  |
| **Set计算** | **P2D** |  | **pd** | Pd |  |
| **Set计算** | **Dh** | 水力学直径 | **/** | / hydrau |  |
| **Set计算** | **X\_point** | 径向位置 | **/** | / grid |  |
| **Set计算** | **X\_surface** | 径向面积 | / | / |  |
| **Set计算** | **Df** |  | / | / |  |
| **Set计算** | **Dg** |  | / | / |  |
| **Set计算** | **dc** |  | / | / |  |
|  |  |  | **Height** |  |  |
|  |  |  | **AssmShellThick** |  |  |

新建一个工程，单独测试热工模块；为何要改，因为涉及空间坐标的对应问题；要改模式也需要用到；

仿照Daisy的结构，修改热工模块中的geom和mesh；

先把几何类调整一致，再在输入卡中写入几何类的读入；

接着就是mesh类；

**修改geom类：**

~~相同的参数改名字：shellThick,GasGap,N\_pin,rFue~~

~~删除组件外壳厚度：AssmCladth~~

~~增加rod,N\_pin,area,pitch~~

搞懂Daisy的输入卡，从输入卡读入geom中需要读入的数据；

在driving\_plain\_read中增加read\_feedback\_imp（）测试输入卡

在read\_feedback\_imp中写要通过输入卡输入的数据；

改写geom的set函数；

改写想要的hydrau类；

**Grid模块保持不变；**

**水力学模块中修改flowarea，Dh；**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **daisy** | | | **imp** | | |
| **数据来源** |  | | | 修改前 | 修改后 | 数据来源 |
| **输入卡** | Mesh(nf,nc) | **Nf** | 芯块节点数 | **Nf** | **Nf** | **输入卡** |
| **输入卡** | Mesh(nf,nc) | **Nc** | 包壳节点数 | **Ns** | **Ns** | **输入卡** |
| **Nf+Nc+2** | Mesh()cal | **N\_mesh** | 径向节点总数 | **/** | **/** |  |
| **函数设置** | Set() | **Nr** | 径向控制体数 | **/** | **/** |  |
| **函数设置** | Set() | **Na** | 轴向网格数 | **ny** | **Ny** | **Set()** |
| **函数设置** | Set() | **Na\_strat** | 活性区起始 | **/** | **Ny\_start** | **Set()** |
| **函数设置** | Set() | **Na\_end** | 活性区结束 | **/** | **Ny\_end** | **Set()** |
|  |  | **N\_assm\_geom** | 组件几何类型 | **/** | **/** |  |
|  |  |  |  | **Ng** | **Ng** | **输入卡** |
|  |  |  |  | **R(:,:)** |  |  |
|  |  |  |  | **Z(:,:)** |  |  |

**daisy径向热工网格的设置：输入卡输入**

**daisy轴向热工网格的设置：**

daisy.f90: daisy

driver\_pre\_process.f90: Run\_pre\_process()

input\_driver.f90 : Driving\_input()-

allocate\_after\_canning()

Line 192::

call nth%set (ns%state%zone, ns%state%layer, ns%state%layer\_top, ns%state%layer\_bottom)

**height变量的输入**

daisy.f90: daisy

Th\_pre\_process.f90 : Driving\_th\_pre\_process（）

call geom\_th%set\_height (geom%height)

this%height = this%height \* CM2METER\_ 注意单位转换

**物理中的zone,layer,layer\_top和layer\_bottom,height由输入卡输入：**

input\_plain.f90 (2 hits)::

case ('LAYER')

read(unit=aline, fmt=\*, iostat=io\_error) keyword, dummy\_int(1:3)

Line 802: ns%state%**layer** = dummy\_int(1

case ('SCALE')

read(unit=aline, fmt=\*, iostat=io\_error) keyword, dummy\_int(1:4)

Line 693: **ns%state%zone** = dummy\_int(4)

case ('REF\_AXIAL')

read(unit=aline, fmt=\*, iostat=io\_error) keyword, dummy\_int(1:2)

Line 812: ns%state%**layer\_top** = dummy\_int(2)

Line 1985: ns%state%**layer\_top** = dummy\_int(2)

case ('HEIGHT')

Line 1991: geom%height = dummy\_real(1: ns%state%layer)

# 程序开发日志

~~检验sys输入输出中的目录设置./功能，方便移植；~~

~~将sys的main改成了module，其中的主函数变成contains里面的subroutine，成为一个显式接口；~~

~~将assmebly主函数重命名为sys\_th\_interface；~~

~~将所有自己的全局变量名前加IMP防止混淆；~~

~~首先改~~**~~模块名~~**~~，其次改~~**~~文件名~~**~~，这两个是全局变量；（main,module,外部函数这三个是全局名），全局名精确到IMP，其他类的名称遇到问题再说。~~

~~git只能对文档中的改变进行操作，测试能否对文件夹中文件的删除、重命名等进行操作？~~

~~增加一个文件需要add，能否返回？可以。~~

~~删除一个文件，也可以返回；删除一个文件并在回收站中清除，无法恢复；清空以后恢复了空文件；都可以恢复；只用来进行文档的管理？新增文件需要add；重命名文件对git来说相当于delete原来文件名的文件；delete的文件可以恢复；~~

~~修改IMP\_th\_interface为perform\_th\_imp;~~

在Ns%feedback中增加变量is\_imp;

在check\_feed\_steady(),中增加is\_imp的反馈分支，模仿is\_inner写is\_imp的计算流程；

feedback开头：use TH2NK\_interface\_imp

写imp接口，Perform\_TH\_imp()第一版中预留瞬态就可，不必实现；

在assm类中拆分数据和方法，因为assm类已经是全局变量，没必要在私有化方法；

修改sys\_assmebly\_header模块，将其中的方法全都删除，steady和transient移动到imp\_single\_channel中,其他移动到imp\_singel\_kernel中 ；

**写类：imp\_single\_channel,里面放steady和transient两个方法；对全局变量进行操作；**

~~更改这两个方法名为：driving\_imp\_steady和driving\_imp\_transient~~

需要改写的方法：assm%pow%set(),assm%th\_boundary%update(ctime)

~~直接移动的方法：solve\_momentum,solve\_pressureCorrection,modify\_PV,solve\_temperature,update\_property~~

**写类：imp\_single\_kernel,里面放需要调用的方法；通过虚实结合进行操作数据；**

首先创建上述两个模块，然后分别进行移动；

assembly类中需要移除掉和物理共用的全局变量，如几何、网格、功率；

~~上级往下级改造，一层一层改；~~

~~删除driving\_cal\_transient模块，将模块中的两个函数移动到imp\_sigle\_kernel中去；~~

~~删除assembly类中alloc等方法，移动到driving\_pre\_process中去\_~~

~~将修改后的assm代码复制一份到NK的src中，在NK中重新添加这部分代码，替换以前的代码；编译通过；~~

~~修改TH2NK\_interface\_IMP~~

~~复制一份IMP的可运行版单独调试；~~

daisy中的mesh只精确到组件，而assm中更加细：轴向的网格对应；

搞明白matrix变量的定义；

问题：Daisy中并不是等高分布；热工和物理先用同一套网格，而后用不同的网格；

~~查看Daisy中热工网格的处理模式：ThermalScale 对应自己的mesh;Daisy中物理热工用了同一套网格；~~

模仿Daisy中的th\_geom，th\_mesh改写自己的热工geom，mesh；

热工模块中网格：径向固体网格由输入卡输入，流体网格（zone,layer,bottom,top）由ns赋值，轴向网格由ns赋值；

热工模块中的几何：高度由geom%height赋值；

理清自己的前处理框架，处理好前处理。

几何的设置中，高度由daisy给出；

参照daisy中热工模块的几何网格函数方法定义imp中的方法

几何空间的分配Allocate\_after\_scanning()

基本参数的输入Read\_feedback，包括：n\_pin,n\_fuelpin,pitch,rod,cladth,bond,hole,次级参数的计算，geom\_assm%set()

20171209

### 在daisy输入卡feedback中增加is\_imp关键字；

*需要从输入卡输入的信息，暂时先从自己的输入卡输入，不去研究他的输入卡；*

### 空间高度网格的对应；

明确需要耦合的高度信息；

修改求解器，使其能够求解变空间步长的问题；

在geom中增加height(:)属性，并配方法set\_height（）在imp\_pre\_process中进行调用；

看daisy中热工网格类，input,set,layer三者的处理模式；

imp\_geom/imp\_mesh的改写，主要参数的确认与赋值；关键还是类的设计；

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **daisy** | | | **imp** | | |
| **数据来源** |  | | | 修改前 | 修改后 | 数据来源 |
| **输入卡**  **MESH\_SIZE** | Mesh(nf,nc) | **Nf** | 芯块节点数 | **Nf** | **Nf** | **输入卡** |
| **输入卡**  **MESH\_SIZE** | Mesh(nf,nc) | **Nc** | 包壳节点数 | **Ns** | **Ns** | **输入卡** |
| **Nf+Nc+2** | Mesh()cal | **N\_mesh** | 径向节点总数 | **/** | **/** | **/** |
| **函数设置** | Set() | **Nr** | zone控制体数 | **/** | **N\_zone** | **Set()** |
| **函数设置** | Set() | **Na** | 轴向网格数 | **ny** | **Ny** | **Set()** |
| **函数设置** | Set() | **Na\_strat** | 活性区起始 | **/** | **Ny\_start** | **Set()** |
| **函数设置** | Set() | **Na\_end** | 活性区结束 | **/** | **Ny\_end** | **Set()** |
|  |  | **N\_assm\_geom** | 组件几何类型 | **/** | **/** |  |
|  |  |  |  | **Ng** | **Ng** | **输入卡** |
|  |  |  |  | **R(:,:)** |  |  |
|  |  |  |  | **Z(:,:)** |  |  |

Nf,Ng,Ns,Ny,R(:,:),z(:,:)保持不变；

~~增加两个变量Ny\_start,Ny\_end；~~

~~set方法读取（Ny,Ny\_start,Ny\_end）~~,剩余数据直接由输入卡输入;

~~对应修改set的调用；~~

~~mesh模块调整完毕；~~

新建一个工程，主程序为daisy，主要测试耦合程序的预处理模块；

reInputdata在ny后面，增加Ny\_start,Ny\_end.；Nf,Ng,Ns另外赋值；

先单独修改，把geom和mesh两个类都改好再说。

通过查看height的alloc确认layer是指的节点还是控制体；

~~查看物理geom中有无轴向位置信息，以确认能否进行物理、热工两套不同的网格间的映射；~~

搞清楚daisy中物理网格的具体节点方式；

复制过来是为了使用notepad搜索时带来便利。而且大面积修改，用复制粘贴比较稳妥，到时把主程序中的use部分，以及剩余代码都复制过去即可。

### 首先解决网格的耦合问题

~~修改网格类~~

定义一个全局的类；由于是稳态的耦合，先不考虑程序中的瞬态参量。

~~在driving\_input.f90 中call assm1%mesh%set (ns%state%layer, ns%state%layer\_top, ns%state%layer\_bottom) imp\_assm\_global~~

~~在mesh类中增加N\_zone变量,set方法中增加N\_zone，删除之前的call set~~

~~在input\_plain.f90中use imp\_global,~~添加MESH\_SIZE2,进行网格赋值,先备注掉MESH\_SIZE2,画输入卡模块的流程图；

set\_assembly方法中增加需要从daisy中输入的数据，不分类。有以下：  
mesh中的zone,layer,layer\_start,layer\_end

geom中的height(:)

~~mesh的set函数中增加nf,ng,ns,zone~~

~~reInputdata中删除需要从daisy中导入的数据，ny,ny\_start,ny\_end,删除~~

完成了轴向网格的统一；

修改geom的类以及相关计算，主要增加height变量以及相关的计算调试；

~~height改成height(1:layer)，数值来源于geom%height,热工计算的长度是整个组件的长度，其中；为height写alloc函数，在assembly中;~~

~~k=assm%mesh%layer\_bottom+i~~

用何明涛原来的模型计算稳态热-物耦合，看是否可以计算；

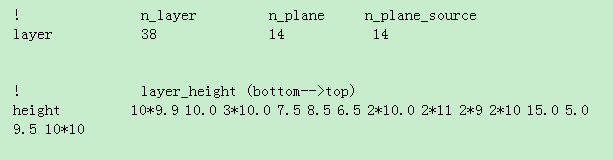
完成稳态热-物耦合；

~~修改height的单位~~

feedback关闭的情况下依旧无法运行稳态计算，说明哪里出了问题，首先要求小高进行稳态计算；

自己的热工模块问题=》assm%geom%height(:)为何无法调用？

### 写imp\_Pre\_process类用以imp模块的前处理；



开始测试耦合是否成功。

## 20171213

### 在服务器上debug代码：

服务安装git;

github建立耦合文件夹；

## 20171216

在服务器上布置编译环境和版本管理环境；

版本环境布置完毕；

布置编译环境；

由于在各个计算器中外部库的路径设置不同，所以一般只是源代码用于管理，工程不用于git。即git只负责管理源码和文档，其他都不负责。

第一目标完成编译，第二目标布置版本管理；

调整编译环境：编译环境调整成功；

添加输入文件，对计算进行debug

问题描述：

当thread=7的时候可以顺利编译，当thread>7的时候报错：  


Debug表明问题出在以下这一步：

Coefficient\_header\_transverse.f90

**!$omp parallel default(shared)**

尝试在编译的时候扩大核数，看是否有效：*不行！*

一边跑7个核的debug，一边查看这句命令的意思。

设置内：

http://www.hivmr.com/db/8mpf31jkkzs7ffxjapck89dd1d9am3p7

Fortran-Optimization - Parallelization = Yes ( /Qparallel )

Linker-System -  
Heapk Commit = 268435456；Heap Reserve = 268435456；Stack Commit = 268435456  
Stack Reserve = 268435456 268435456 = 256MB

*直接退出；用7核也直接退出了。。重新将stack调回0核1e8*

Release中include和lib的路径调整为./，打开parallelization;重新生成工程；

*直接退出；*

可以考虑用cmake生成工程文件；

学会使用cmake