**单通道分析程序开发文档**

主程序；

需要的类；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 主程序 | Assembly.f90 |  |  |  |
| Module | Drving\_cal\_transient.f90 |  |  |  |
|  | Driving\_output.f90 |  |  |  |
|  | Driving\_pre\_process.f90 |  |  |  |
|  | Mathkerel.f90 |  |  |  |
|  | Assm\_global.f90 |  |  |  |
|  | Sys\_assembly\_header.f90 | 定义assm类 |  |  |

类：sys\_assembly

说明：实现单个组件定义、计算功能。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据名称 | 数据类型 | 数据说明 | 公私 | 来源 |
| Fric | Real | 摩擦因子 | private | Set(Input) |
| Geom | Assmgeom |  | private | Set(Input) |
| Mesh | Assmmesh |  | private | Set(Input) |
| Property | Assmmaterial |  | private | init() |
| boundary | Assmboundary |  | private | Init() |
| Initdata | Assminit |  | private | Set(input) |
| Confactor\_ | confactor | 收敛因子 |  | Set(input) |
| Power(:,:) | Real |  | private | Init() |
| Fq\_core(:,:) | real |  | private | Init() |
| thermal | Assmthermal |  | private | Init(),steady()  Transient() |

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 功能 | 输入 | 操作数 |
| Alloc\_assembly | 分配数组 | This | Power,thermal,material |
| Free\_assembly | 释放空间 | this |  |
| Set\_assembly | 设置参数 | This,reInputdata | Geom,mesh,init |
| Init\_assembly | 初始化参数 | this | Property,thermal,boundary,pow |
| cal\_Assembly\_Steady | 稳态计算 | This,power,fq\_core |  |
| cal\_Assembly\_Transient | 瞬态计算 | This,power,fq\_core |  |

**重要的方法**

cal\_Assembly\_Transient(this,dt)

功能：计算下一时间步thermal中的PVT

调用该方法前需要

call set\_assembly() set:geom,mesh,init

call init\_assembly() init:property,boundary,pow,fq\_core,thermal

计算流程



伪代码

subroutine cal\_Assembly\_Transient(this,dt)

class(sys\_assembly),intent(in out)::this

real dt

real pmodify

type(thermal)::last !上一时刻的热工 local

type(thermal):: [iteration](https://www.baidu.com/link?url=HGRvucAOuRN9coOj8dZH0UYxzhcbMV0TAz5bboECOHb3zQbWESnDqIzBRjyO5DdqmpbMQsZh7rL1rnTsaLosR8mDAC7ywtm86n9nFlxlVX3&wd=&eqid=d9892ffb00004f80000000035a0c5724) !当前迭代步的热工 local

drho=..!

do while(btotal>sigmab)

call solve\_momentum(this,dt,last,iteration)

call solve\_pressureCorrection(iteration,pmodify)

call modify\_PV(this,pmodify,iteration)

end do