

MADS操作契约

日期	内容	备注
2021.05.24	初始操作契约，上传了迭代、读取和存储操作的契约	

目录

CO1. load_mesh_vtk

CO2. store_mesh_vtk

CO3. parallel_looping

CO4. serial_looping

操作契约

CO1: load_mesh_vtk

操作: `load_mesh_vtk(const char* pathname)`
交叉引用: 用例: 以给定格式读取网格数据文件并写入内存
前置条件: `pathname`所指向的空间已经被分配; 文件存在且格式合法;
后置条件: `Mesh`对象中的`node`和`cell`向量被修改, 原有数据被覆盖。

CO2: store_mesh_vtk

操作: `store_mesh_vtk(const char* pathname);`
交叉引用: 用例: 以给定格式将内存中网格数据写入文件
前置条件: `pathname`所指向的空间已经被分配; 进程对`pathname`所指定的文件路径有写权限。
后置条件: 在`pathname`所指定的文件路径中有新的 (或者覆盖原有) `vtk`文件生成

CO3: parallel_looping

操作: `void parallel_looping(void(*k_func)(void*, int), int* index, int index_size, void* arg, const char* element_name);`
交叉引用: 用例: 迭代`kernel Function`
前置条件: 被迭代的核函数被定义; 迭代的元素 (`element_name`) 被指定;
后置条件: 取决于用户定义的核函数

CO4: serial_looping

操作: `void serial_looping(void(*k_func)(void*, int), int* index, int index_size, void* arg, const char* element_name);`
交叉引用: 用例: 迭代`kernel Function`
前置条件: 被迭代的核函数被定义; 迭代的元素 (`element_name`) 被指定;
后置条件: 取决于用户定义的核函数

