附件4：

**中物院高性能数值模拟软件中心**

**模块实现报告**

**任务名称：**点在三维非结构网格中定位模块

**任务编号：**201807

**起止时间：**2018.4.17至2018.5.31

**所属项目：**领域编程框架

**项目负责人：**徐小文

**2018年8月**

1. 概述

为了给ICF研究提供模拟数据后处理支撑，需要完成以下功能：给定点的集合和一个三维非结构网格，返回每个点在三维非结构网格中属于哪个单元；给定网格区域外的射线集合和一个三维非结构网格，返回每条射线与三维非机构网格外表面的那个单元相交以及交点坐标。为了提供重映支撑需要完成以下功能：给定两个三维非结构网格，返回这两个网格中那些单元相交。解决上述问题所涉及的网格单元可能为三棱锥、三棱柱、六面体以及其他可能为凹的情况，104点在106网格单元定位计算时间不超过1分钟，104射线在105面计算时间不超过1分钟。

1. 软件模块使用方法

本模块的外部接口为一个类，接口代码如下：

构造函数：

|  |
| --- |
| */\*\**  *\* @brief 构造函数*  *\**  *\* @param source\_patch 用于点相交、射线相交、网格相交的目的网格*  *\*/*  **GridIntersect**(tbox::Pointer<hier::Patch<3> >& source\_patch); |

点与网格相交函数：

|  |
| --- |
| */\*\**  *\* @brief 判断点是否在网格中*  *\**  *\* @param points 输入点坐标*  *\* @param n 点的数量*  *\* @return std::vector<int> 返回每个点所在网格单元，*  *\* -1代表在网格外，*  *\* >=0代表在网格中*  *\*/*  std::vector<int> **pointInGrid**(const double\* points, int n); |

射线与网格相交函数：

|  |
| --- |
| */\*\**  *\* @brief 点与网格外表面相交,输出相交网格单元编号与交点坐标*  *\**  *\* @param start\_points 射线起点集合*  *\* @param direction 射线方向集合*  *\* @param n 射线的数量*  *\* @param ids 输出参数，交点所在网格单元编号*  *\* -1代表射线与网格外表面不相交，交点无意义*  *\* >=0代表射线与网格外表面相交，交点所在网格单元索引*  *\* @param intersection 输出参数，交点坐标*  *\*/*  void **rayIntersectGrid**(const double\* start\_points, const double\* direction,  int n, std::vector<int>& ids,  double\* intersection\_coordinates); |

网格与网格相交函数：

|  |
| --- |
| */\*\**  *\* @brief 输入目的网格片，求解源网格片与目的网格片相交的单元个数*  *\* 以及源单元索引号。注：由于采用区间树算法，所以求出的结果不是*  *\* 所有单元都真正相交，真正相交的网格单元只是输出集合的子集*  *\**  *\* @param dest\_patch 输入参数，目的网格片*  *\* @param intersect\_num 输出参数，相交源网格单元个数*  *\* @param intersect\_index 输出参数，相交源网格单元索引号*  *\*/*  void **gridIntersectGrid**(tbox::Pointer<hier::Patch<3> >& dest\_patch,  int& intersect\_num, std::vector<int>& intersect\_index);  */\*\**  *\* @brief 输入目的网格片，以及指定单元索引号，求解与该网格单元*  *\* 相交的源网格单元索引。注：由于采用区间树算法，所以求出的结果不是*  *\* 所有单元都真正相交，真正相交的网格单元只是输出集合的子集*  *\**  *\* @param dest\_patch 输入参数，目的网格片*  *\* @param focused\_cell\_index 输入参数，目的网格单元索引*  *\**  *\* @return std::vector<int> 输出参数，与输入网格单元相交的源网格单元索引集合*  *\*/*  std::vector<int> **gridIntersectGrid**(tbox::Pointer<hier::Patch<3> >& dest\_patch,  int focused\_cell\_index); |

1. 软件模块实现

（一）软件模块结构

（详细介绍软件模块的内部组织结构、内部模块之间的交互关系及接口。）

（二）子模块一

（详细介绍子模块的组织结构、内部接口、重要数据结构、核心算法、实现技巧。）

（三）子模块二

（详细介绍子模块的组织结构、内部接口、重要数据结构、核心算法、实现技巧。）

1. 其它需要说明的问题