猪头 Geant4讲座第六讲——几何模型

上一讲我们讲过了，几何结构类（DetectorConstruction）属于强制初始化类，其主要功能是构建模拟问题的几何结构，包括各部分的材料、形状、尺寸、位置等信息。

前面我讲了材料如何定义，今天我主要讲几何模型的建立以及各部分材料的设置。

在讲如何建立几何模型前我想首先讲一下 Geant4中所采用的几何建模思想。

学过 MCNP 或 Fluka 的都知道 MCNP 和 Fluka 采用的是 CG 模型，所谓 CG 模型就是 Combination Geometry，我习惯翻译为组合几何模型，CG 模型顾名思义就是整个模型由一块块小模块组合而成，打个不恰当的比方就是搭积木。这种模型的要求就是“不交不空”，既不能有相交的部分，也不能有空白的地方。CG 模型是粒子输运蒙卡模拟中非常常用的一种。

Geant4采用的模型则不同，目前我没看到有关此模型的确切名称，但我习惯将之称为嵌套模型或盒子模型，因为其建模的方式就如往大盒子里放小盒子。在 Geant4中首先我们要建立一个最大的盒子，称为 World Volume，然后往这个大盒子里面放各种各样的小盒子（部件），然后每个小盒子（部件）里面还可以放更小的盒子（零件），放入的小盒子将自动代替大盒子原有部分。在 Geant4中，将大盒子称为 Mother Volume（母体），小盒子称为 Daughter Volume（子体）。

除了 World Volume 之外，每个 Volume（体）都必须且只能有一个母体，但可以没有子体，也可以有多个子体。

Geant4的这种盒子模型的要求是：“不交不超”“不交”就是要求同一个大盒子里面的小盒子，

即同一等级的子体不能有相交的部分；“不超”就是要求小盒子不能超过大盒子的范围，即子体不能超出母体。用集合的语言描述就是

Volume i(level n)∩Volume j(level n)=Φ; Daughter Volume∈Mother Volume.

下面就可以讲如何设置这些盒子（Volume）。

Geant4中每个这样的盒子的安放又分为两步。第一步是构建盒子，称为 Logical Volume（逻辑体）；第二步是将盒子摆放到正确位置，变为 Physical Volume（物理体）。

构建盒子又分为两步，第一步是确定盒子形状，第二步是确定盒子的材料等属性。

形状在 Geant4中被称为 Solid。在 Geant4中提供了多种固有的形状，如球形、长方体、锥体等，可以在$G4INSTALL/source/geometry/solids 里面查找。

用户也可以通过 G4VSolid 类构建自己的形状，请参见$G4INSTALL/source/geometry/management/include/G4VSolid.hh。

此外，对于一些复杂的形状，用户也可以利用基本形状通过交并补等布尔运算方式完成，布尔运算的方式请参考$G4INSTALL/source/geometry/solids/Boolean。

确定了盒子形状后，就是设置盒子的材料、磁场等属性。

这些属性的设置通过 G4LogicalVolume 类来完成，设置方法如下：

```
G4LogicalVolume(G4VSolid* pSolid, //形状
G4Material* pMaterial, //材料
const G4String& name, //逻辑体名字
G4FieldManager* pFieldMgr=0, //场管理
G4VSensitiveDetector* pSDetector=0, //是否 SD 探测器
```

G4UserLimits* pULimits=0, //用户限制
G4bool optimise=true); //是否优化

盒子造完了就该摆放盒子了。

摆放盒子也有两种方法，一种是直接构建物理体，另一种是指定摆放方法。

直接构建物理体是通过 G4VPhysicalVolume 类，其定义方法如下：

G4VPhysicalVolume(G4RotationMatrix *pRot, //旋转方式

const G4ThreeVector &tlate, //摆放坐标
const G4String &pName, //物理体名字
G4LogicalVolume *pLogical, //对应的逻辑体
G4VPhysicalVolume *pMother); //母体

如果 pMother=0就表明这个体是一个 World Volume，World Volume 必须且只能有一个。

在实际应用中，我们通常采用指定摆放方法的方式来完成物理体的构建。

指定摆放方法是通过 G4PVPlacement 类完成。G4PVPlacement 类是 G4VPhysicalVolume 的派生类，该类提供了多种方法描述 Logical Volume 的摆放方法。具体可以参考$G4INSTALL/source/geometry/volumes/include/G4PVPlacement.hh。

用这种方法可以建立具有相同 Logical Volume 的物理体，同时给可以给每个物理体分配一个编号，以便区分具有相同 Logical Volume 的物理体。这些编号在 UserSteppingAction 等类中处理数据时有时会非常有用处。

需要注意的是，在 Geant4中摆放坐标都是指的相对坐标，是子体中心相对母体中心的坐标。而 World Volume 建立后就等于建立了几何模型的绝对坐标系。

下面简单地讲一下第一个 novice 例子中几何模型的建立。

```
G4double expHall_x = 3.0*m;
G4double expHall_y = 1.0*m;
G4double expHall_z = 1.0*m;
G4Box* experimentalHall_box
= new G4Box("expHall_box",expHall_x,expHall_y,expHall_z);
```

这是建立了一个长方体，需要注意的是 Geant4中 Solid 的原点通常是设置在这个形状的中心的，而 MCNP 和 Fluka 中则大多是设置在某个顶点或某个底面的。

此外，定义长方体等形状时大多是用半长度/半宽度作为参数，而不是整个长度和宽度。

```
experimentalHall_log = new G4LogicalVolume(experimentalHall_box, //对应的 Solid
Ar, //材料

"expHall_log", //名字

0, //无场管理

0, //不是 SD

0); //无用户限制
```

这里需要注意的是最后省略了 optimise，而是采用了默认值。

```
experimentalHall_phys = new G4PVPlacement(0, //无旋转

G4ThreeVector(), //放置在(0,0,0)
```

experimentalHall_log, //对应逻辑体

"expHall", //名字

0, //母体

false, //pMany

0); //Copy No.

1、这里母体为0表明这是个 World Volume；

2、pMany 目前没有用处，根据 Geant4的描述，将来也许会用于重复结构；

3、这里同样省略了最后一个参数 pSurfChk。