# 前言

exampleB1模拟了光子与质子入射人体组织时的能量沉积，其几何模型为长方体与圆台，统计对象为能量沉积，输出对象为人体组织的吸收剂量(单位：Gy)。在此物理模型基础上，我们可以对几何模型、统计对象以及输出对象进行修改，来为我们的研究工作所服务。

本人的研究工作是：根据exampleB1的物理模型，将几何模型修改为薄板，将输出对象修改为能量沉积值(单位：MeV)，进而来计算不同物质的质量阻止本领。经验证，此方法计算出的质量阻止本领，对于质子而言，在入射动量为0.1~1GeV/*c*时，与Bethe公式符合的非常好；对于电子而言，在入射能量为5~40MeV时，与PDG在review of particle physics2020中所给出的结果相比，质量碰撞阻止本领符合得很好，质量辐射阻止本领符合得相对较差。

本工作的不足之处在于：

(1) 对于质子而言，当质子动量低于0.1GeV/*c*或高于1GeV/*c*时，GEANT4模拟所得结果与Bethe公式符合得较差，并且，Bethe公式中的密度修正项未研究清楚。

(2) 电子质量辐射阻止本领的GEANT4模拟结果与文献所给出的结果符合得较为不好。

# 1. 如何修改几何模型

## 1.1 修改几何体尺寸

几何模型的所有内容在B1DetectorConstruction.cc中，包括几何体尺寸、几何体位置坐标、几何体旋转角度、几何体材料、几何体布尔运算等内容。本节首先介绍如何修改几何体尺寸

G4double env\_sizeXY = 20\*cm, env\_sizeZ = 30\*cm;

G4Box\* solidEnv =

new G4Box("Envelope", 0.5\*env\_sizeXY, 0.5\*env\_sizeXY, 0.5\*env\_sizeZ);

以上为定义envelope几何尺寸的代码，修改env\_sizeXY与env\_sizeZ即可进行envelope几何尺寸的修改。注意：newG4Box中env\_sizeXY与env\_sizeZ的值乘以了0.5，但是在模拟计算过程中，程序会自动将模型尺寸还原为env\_sizeXY与env\_sizeZ

G4double world\_sizeXY = 1.2\*env\_sizeXY;

G4double world\_sizeZ = 1.2\*env\_sizeZ;

G4Box\* solidWorld =

new G4Box("World", 0.5\*world\_sizeXY, 0.5\*world\_sizeXY, 0.5\*world\_sizeZ);

以上为定义world几何尺寸的代码，修改world\_sizeXY与world\_sizeZ即可进行world几何尺寸的修改。

G4Box\* box1 = new G4Box("Box1", 10.\*cm, 10.\*cm, 20.\*um);

以上为定义shape1几何尺寸的代码，new G4Box中的值分别代表x,y,z方向上的板厚。注意，这三个值是实际板厚的一半，如z方向板厚20um，实际建模出的板厚为40um。修改new G4Box中的值即可进行shape1几何尺寸的修改。

## 1.2 修改几何体位置坐标

以shape1为例说明如何修改几何体位置坐标。

G4ThreeVector pos1 = G4ThreeVector(0, 0, 0\*cm);

以上为定义shape1三维坐标的代码，修改G4ThreeVector中的参数即可进行几何体位置坐标的修改。

## 1.3 修改几何体材料

以shape1为例说明如何修改几何体材料

G4Material\* shape1\_mat = nist->FindOrBuildMaterial("G4\_C");

以上为定义shape1材料的代码，修改FindOrBuildMaterial中的内容即可进行几何体材料的修改。

建议：将world与envelope的材料设置为真空，即G4\_ Galactic，以便只在shape1中统计到粒子的能量沉积。

# 2. 如何修改统计对象

## 2.1 统计对象类别

GEANT4中可统计与粒子类型、运动、相互作用等有关的多个物理量，具体如下：

//获取相互作用类型

const G4String& processName=

step->GetPostStepPoint()->GetProcessDefinedStep()->GetProcessName();

//获取粒子名称

const G4String& particleName=

step -> GetParticleDefinition()->GetParticleName();

//获取粒子当前坐标

G4ThreeVector xyzTrack = step ->GetPosition();

//获取粒子TrackID

G4int trackID= step ->GetTrackID();

//获取粒子ParentID

G4int parentID= step ->GetParentID();

//获取该step下的步长

G4double steplength= step ->GetStepLength();

//获取stepID

G4int stepID = step ->GetCurrentStepNumber();

//获取该粒子的TrackLength

G4double tracklength= step ->GetTrackLength();

//获取step初始点的动能

G4double KineticEnergypre = step->GetPreStepPoint()->GetKineticEnergy();

//获取step结束点的动能

G4double KineticEnergy = step ->GetKineticEnergy()

//获取该step下的能量沉积

G4double edepStep = step->GetTotalEnergyDeposit();

## 2.2 如何修改统计对象

描述统计对象的代码在B1SteppingAction中，即

// collect energy deposited in this step

G4double edepStep = step->GetTotalEnergyDeposit();

fEventAction->AddEdep(edepStep);

如若要统计其他物理量，即可在此进行修改。

## 2.3 如何分类统计多个对象

在电子入射等情形下，可能会出现多种相互作用类型的情况，当我们需要对每一种相互作用的类型进行研究时，则须分类进行统计。

以统计能量沉积为例，分类统计的代码如下：

1. B1SteppingAction中

G4double edepStep = step->GetTotalEnergyDeposit();

fEventAction->AddEdep1(edepStep);

if (processName == "msc"&&parentID==0)

{

fEventAction->AddEdep2(edepStep);

}

if (processName == "eIoni"&&parentID==0)

{

fEventAction->AddEdep3(edepStep);

fEventAction->AddEdep4(KineticEnergypre-KineticEnergy-edepStep);

}

if (processName == "eBrem"&&parentID==0)

{

fEventAction->AddEdep5(edepStep);

fEventAction->AddEdep6(KineticEnergypre-KineticEnergy-edepStep);

}

(2)B1EventAction.cc、B1RunAction.cc、B1EventAction.hh、B1RunAction.hh也要进行修改，原因是在B1原先的例子中，只讨论了一种统计对象，如今讨论多个统计对象时需对应建立多个变量与函数。具体修改处请见附图文件夹。

# 3. 如何修改输出对象

## 3.1 exampleB1模拟结果的输出方式

exampleB1常见的输出方式有：(1)输出txt格式；(2)输出至UI界面。

输出txt文件中的格式可以使用默认的格式，方式如下：在命令行窗口中到达B1-build的路径后，在之后输入.\Release\exmapleB1 run1.mac>log.txt

结果如图

文本, 表格

描述已自动生成

也可自己定义输出格式，定义方式如下：

#include <iostream>

#include <fstream>

ofstream fqcout2("result.txt",ios::app);

fqcout2<<"eventID,pName"<<" "<<eventID<<" "<<particleName <<G4endl;

如此，在B1-build文件夹中即可得到自己定义的输出结果。

## 3.2 如何在RunAciton中修改输出对象

在exampleB1中，描述输出对象的代码在B1RunAction里，默认的输出结果如下：

--------------------End of Global Run-----------------------

The run consists of 5 alpha of 1 MeV

Cumulated dose per run, in scoring volume : 0 picoGy rms = 0 picoGy

------------------------------------------------------------

即，默认所返回的是吸收剂量值。

在计算阻止本领的过程中，我们需要的是能量沉积值，而不是吸收剂量值。因此，我们对B1RunAction中的代码进行修改，让其输出能量沉积值。具体如下：

1. 删除吸收剂量单位定义代码

文本, 信件

描述已自动生成

1. 删除计算吸收剂量的代码

文本

描述已自动生成

1. 更改输出格式，将下图中红色框线中的代码

文本

中度可信度描述已自动生成

修改为

G4cout

<< G4endl

<< " The run consists of " << nofEvents << " "<< runCondition

<< G4endl

<< " Cumulated energy per run, in scoring volume Total: "

<< edep1<<" MeV"

<< G4endl

根据自己需要，还可以再增加输出内容。