中国科学技术大学
University of Science and Technology of China

脚踩GEANT4(I)

薛明萱

核探测与核电子学国家重点实验室
中国科学技术大学

*（仅供内部学习使用请勿外传，如有雷同怎么不可能）

内容概要

- Geant4简要介绍
- 模拟中整体框架搭建
  - 探测器（Detector Construction）
  - 物理过程（Physics List）
  - 粒子产生（Primary Generator Action）
- 运行及结果分析

安装/工具参考书等

蒙特卡洛方法简介

➢ 蒙特卡洛方法（Monte Carlo method）

- 也称统计模拟方法，是20世纪40年代由数学家乌拉姆（Stanislaw Ulam）在曼哈顿项目中工作时最先提出
- 科学技术的发展和电子计算机的发明催生了这种以**概率统计理论**为指导的数值计算方法，即用**随机数**（或更常见的伪随机数）来解决很多计算问题的方法

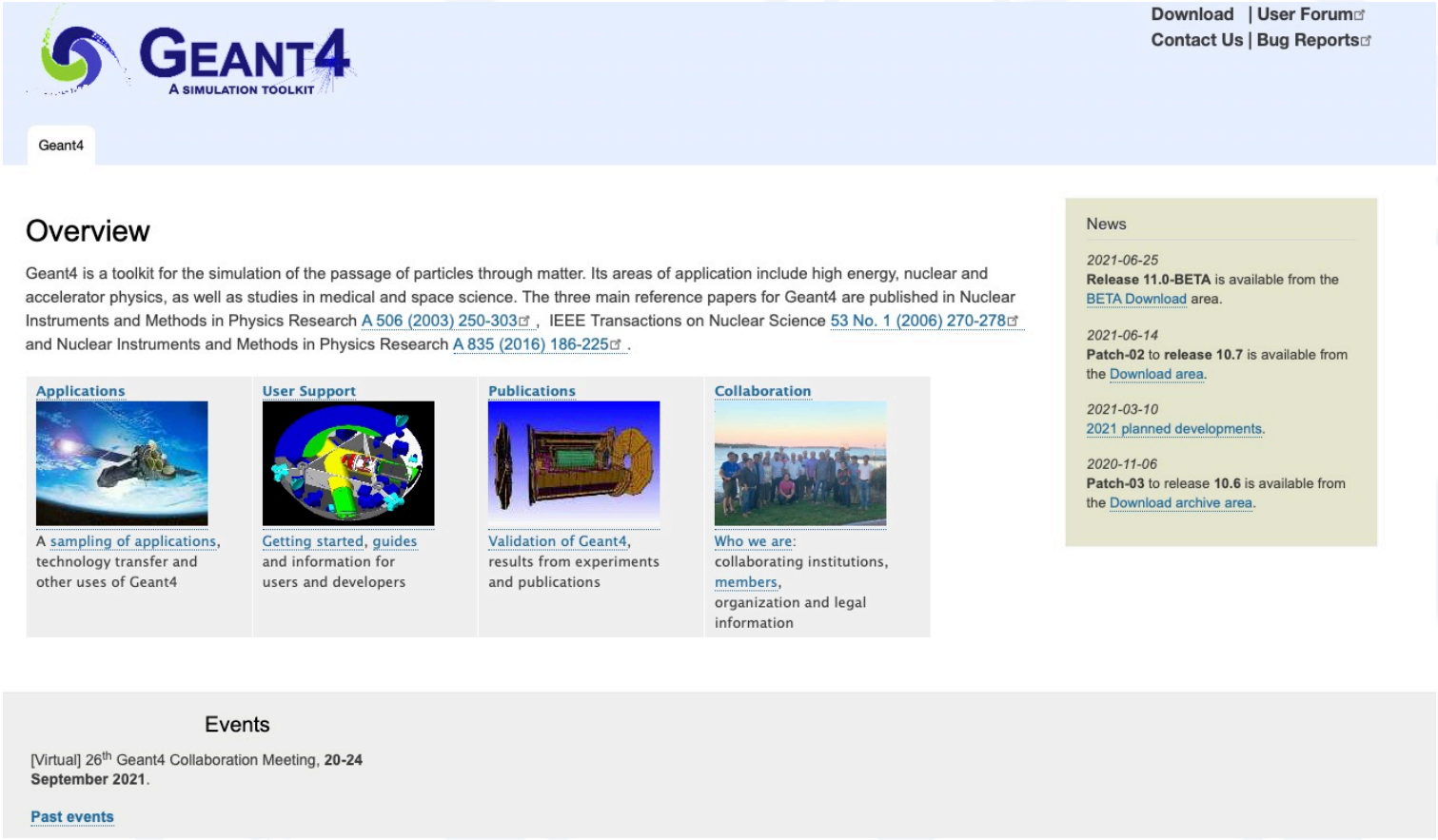
举例：核物理研究中，分析**中子在反应堆中的传输过程**

根据量子力学原理，人们只知道中子与物质中的原子核发生相互作用的概率，而无法准确获得其相互作用时的位置以及裂变产生的新中子的速率和方向。

物理学家们依据其概率进行随机抽样得到裂变位置、速度和方向信息，借助计算机的运算能力直接模拟这种过程，经过模拟大量中子的行为后，通过统计就能获得中子的传输范围，从而作为反应堆设计的依据。

GEANT4简介

官网 ***https://geant4.web.cern.ch***



GEANT4简介

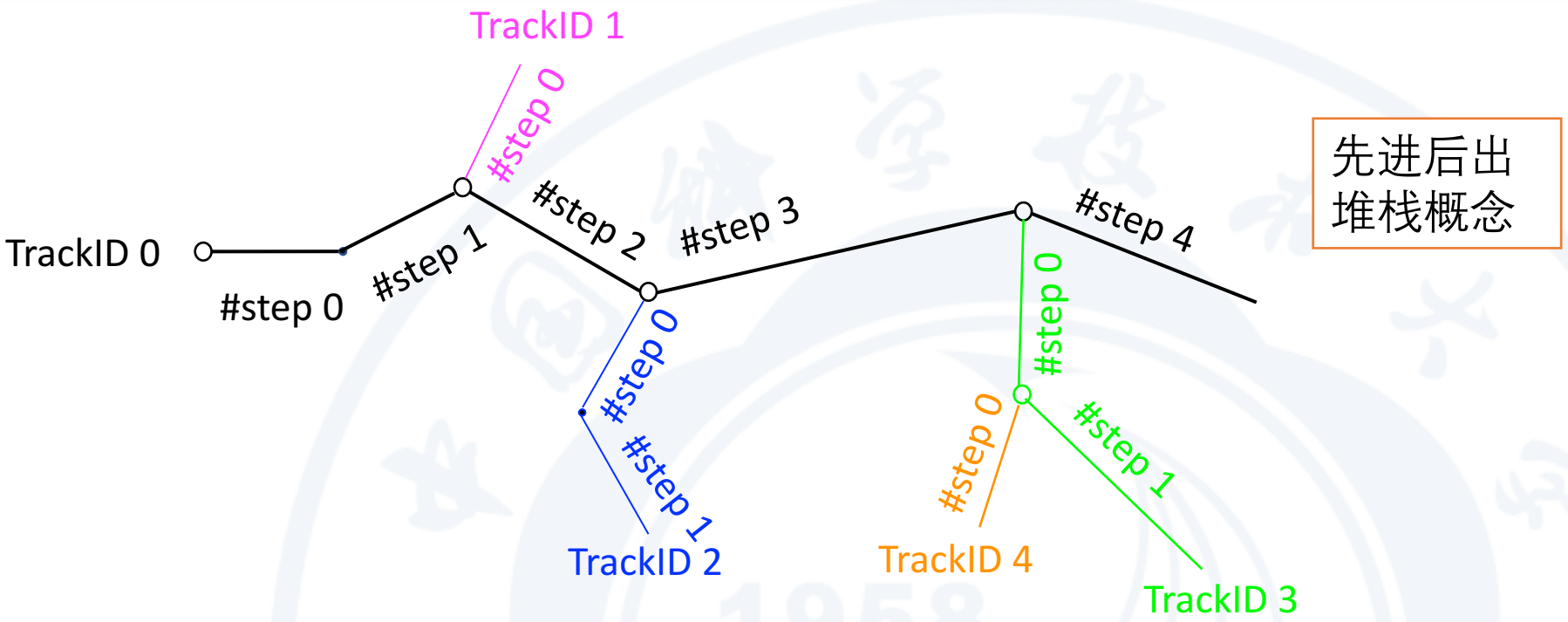
➢ GEANT（GEometry ANd Tracking）

- Geant4计划始于1994年，主要开发机构是CERN
- 使用**蒙特卡洛方法**模拟**粒子在物质中输运的物理过程**的应用**软件包**
- 基于C++面向对象技术，源代码完全开放，具有良好的可扩展性和兼容性
- 广泛应用于粒子物理、核物理、加速器物理、空间天文物理以及医学研究等领域

- Geant4采取模块化实现完备的模拟功能，最重要的三部分可以总结为：
  1）**探测器几何结构搭建**，包括探测器的布局、形状、大小、材料等；
  2）**粒子与物质相互作用**，涵盖电磁、强弱、衰变及光学等过程；
  3）**初始化粒子**，涉及粒子的类型、能动量、方向、时间等信息；

我们的思路是：关注基本粒子模拟，追溯每一个**微观粒子的行为**，利用已知的粒子与物质（粒子）相互作用截面，查看、计算粒子在某块物质内的**能量沉积、时间特性、空间分布**，来模拟探测器的实际响应。

G4典型模拟过程处理



先进后出
堆栈概念

明确**Run、Event、Track和Step**所代表的不同概念

- Run即一次运行模拟，程序的一个运行周期，从BeamOn函数开始执行到之行结束为一个Run；
- Event即一次事件模拟，一个源粒子的整个输运过程；
- 每一个Event中粒子与物质相互作用，会产生很多次级粒子，每个粒子都有一条径迹，即**Track**；
- 而每一个粒子（初始的或次级的）的径迹又有很多“步”组成，称为Step。

G4典型模拟算法

- 首先建立一次模拟，在G4中称为一次Run
- Run建立后，需要对几何结构、物理过程进行初始化
- 初始化完成后开始模拟过程
- 首先发射一个粒子，每一步都按照蒙卡方法进行模拟。。。。

我们感兴趣的结果可以在上述循环过程中自行添加

举例：能量沉积信息

✓ 每一步step能量沉积-->每一条track能量沉积

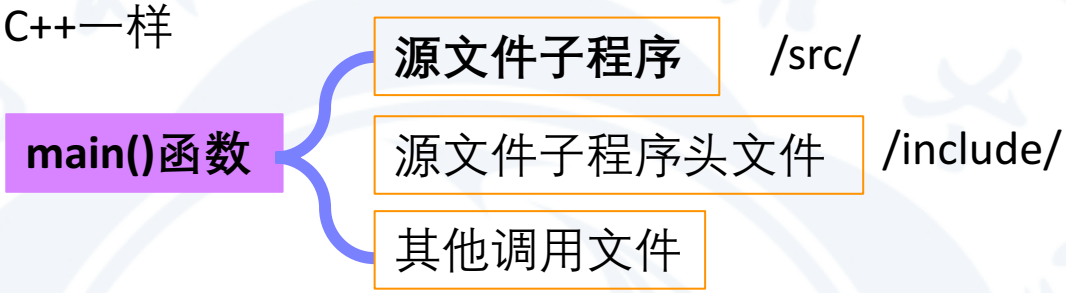
✓ 相同时刻（时间段）能量沉积

```
A Run Start -> 初始化物理模型/几何模型
-> An Event Start -> 调用粒子发射器发射粒子
   -> A Track Start
      -> A Step Start
      -> A Step End
      -> Next Step Start
      -> ……
      -> All Step End
   -> A Track End
   -> Next Track Start
      -> ……
   -> All Track End
-> An Event End
-> Next Event Strat
   -> ……
-> All Event End(All Primaries Shot)
-> A Run End
-> Next Run Start
   -> ……
```

G4程序结构

- Geant4是一个针对物理问题封装好的C++工具包，C++类的集合
- Geant4的过程就是对于其中的接口类进行重载和实现，具体的就是继承关键接口类（强制类+可选类），重载其中所有的关键函数，具体在**main()函数**中调用

G4采用gcc编译器，程序结构和C++一样



*G4为了和C++有所区别，程序后缀都是**.cc**，头文件后缀都是**.hh**

举例：/examples/extended/radioactivedecay/rdecay01/

```
lupengzhongdeMacBook-Pro:rdecay01 lupengzhong$ ls
CMakeLists.txt          README                  neutron.mac             src
Cf238.mac               Ra228.mac               plotHisto.C             timeWindow.mac
Co60.mac                UserData                proton.mac              timeWindowBiased.data
GNUmakefile             alpha.mac               rdecay01.cc             timeWindowBiased.mac
Gd158.mac               debug.mac               rdecay01.in             vis.mac
History                 electronicCapture.mac   rdecay01.out
No252.mac               fullChain.mac           references
Po212.mac               include                 singleDecay.mac
```

main()函数

main()函数

```
//运行管理器（大管家）
G4RunManager* runManager = new G4RunManager

//探测器（身体框架四肢）
G4VUserDetectorConstruction* detector = new MyDetectorConstruction;
runManager->SetUserInitialization(detector);

//物理过程（经络）
G4VUserPhysicsList* physics = new MyPhysicsList;
runManager->SetUserInitialization(physics);

//粒子产生（血液）
G4VUserPrimaryGeneratorAction* gen_action = new MyPrimaryGeneratorAction;
runManager->SetUserAction(gen_action);

//初始化内核（准备）
runManager->Initialize();

//进行模拟（一次跑步运动）
G4int numberOfEvent=1000000;
runManager->BeamOn(numberOfEvent);
```

rdecay01.cc

```
  //construct the default run manager
#ifdef G4MULTITHREADED
  G4MTRunManager* runManager = new G4MTRunManager;
  runManager->SetNumberOfThreads(std::min(4,G4Threading::G4GetNumberOfCores()));
#else
  //my Verbose output class
  G4VSteppingVerbose::SetInstance(new SteppingVerbose);
  G4RunManager* runManager = new G4RunManager;
#endif

  //set mandatory initialization classes
  //
  runManager->SetUserInitialization(new DetectorConstruction);
  runManager->SetUserInitialization(new PhysicsList);

  runManager->SetUserInitialization(new ActionInitialization);

  //initialize G4 kernel
  runManager->Initialize();

  //initialize visualization
  G4VisManager* visManager = nullptr;

  //get the pointer to the User Interface manager
  G4UImanager* UImanager = G4UImanager::GetUIpointer();
```

创建运行管理器

通过Set函数设置几何结构、物理过程，两个强制类/初始化类

通过Set函数设置源描述类，强制类/用户干涉类

Initialize()函数初始化G4内核

可视化初始化

创建UI（User Interface）管理器

源文件子程序/src/

```
lupengzhongdeMacBook-Pro:rdecay01 lupengzhong$ cd src/
lupengzhongdeMacBook-Pro:src lupengzhong$ ls
ActionInitialization.cc   HistoManager.cc           Run.cc                    TrackingAction.cc
DetectorConstruction.cc   PhysicsList.cc            RunAction.cc              TrackingMessenger.cc
EventAction.cc            PrimaryGeneratorAction.cc SteppingVerbose.cc
lupengzhongdeMacBook-Pro:src lupengzhong$ ls ../include/
ActionInitialization.hh   HistoManager.hh           Run.hh                    TrackingAction.hh
DetectorConstruction.hh   PhysicsList.hh            RunAction.hh              TrackingMessenger.hh
EventAction.hh            PrimaryGeneratorAction.hh SteppingVerbose.hh
```

插一句：G4里面的几个基本类基本上与src一一对应

- 几个重要子类
  - 强制类：**必须有！缺少任一个程序都无法运行！**
    - *几何结构类（DetectorConstruction）*
    - *物理设定类（PhysicsList）*

    } *初始化类*

    - 粒子发射器类/源描述类（PrimaryGenerator）
  - 可选类：
    - 运行处理类（RunAction）
    - 事件处理类（EventAction）
    - 径迹处理类（TrackingAction）
    - 步处理类（SteppingAction）

    } *用户干涉类*

DetectorConstruction

探测器的几何形状；在探测器中使用的材料；探测器敏感区域的定义；敏感区域的读出方式

```
G4VPhysicalVolume* DetectorConstruction::Construct()
{
  //
  // define a material
  //
  G4Material* Air =
  G4NistManager::Instance()->FindOrBuildMaterial("G4_AIR");

  //
  // World
  //
  G4Box*
  solidWorld = new G4Box("World",                                  //its name
                   fWorldSize/2,fWorldSize/2,fWorldSize/2);//its size

  G4LogicalVolume*
  logicWorld = new G4LogicalVolume(solidWorld,                     //its solid
                                   Air,                            //its material
                                   "World");                       //its name
  G4VPhysicalVolume*
  physiWorld = new G4PVPlacement(0,                          //no rotation
                                 G4ThreeVector(),            //at (0,0,0)
                                 logicWorld,                 //its logical volume
                                 "World",                    //its name
                                 0,                          //its mother  volume
                                 false,                      //no boolean operation
                                 0);                         //copy number

  //
  //always return the physical World
  //
  return physiWorld;
```

定义材料——空气Air

定义世界体World——立方体；
里面充满空气Air；
放在（0, 0, 0）坐标；

最大的几何体叫世界World，她包含所有的几何体。

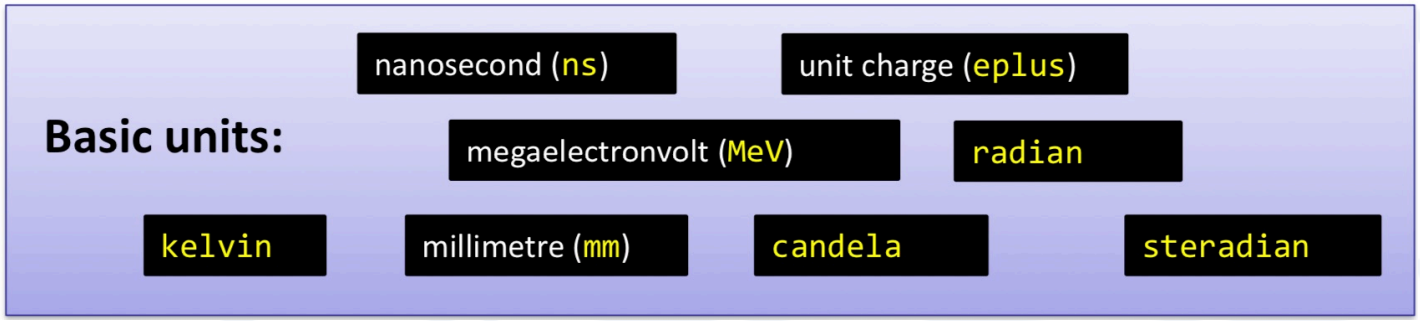
解读：可以理解为创建了一间实验室，之后所有的“行为”等都发生在这个实验室内。

实体(solid volume)
逻辑体(logic volume)
物理体(physical volume)

解读：逻辑体就是实体的所有几何特性+物理特性（材料/敏感单元/磁场等等）

G4物理量单位

- G4常用物理量默认单位



每个物理量可以带任意用户方便的单位，如长度单位默认是mm，但用户可以改为m。

其他单位都由Basic units演化而来，G4中对每个单位都用常变量进行了定义，可以很方便地对物理量进行描述方法。而且大部分常用单位都有全名和缩写两种。

举例：定义长度1cm

```
G4double length=10.;
G4double length=10. * mm;
G4double length=10. * millimeter;
G4double length=1. * cm;
G4double length=1. * centimeter;
```

以上描述均等价

建议大家在定义变量时候写清楚单位，尽量不使用默认

DetectorConstruction——材料定义I

- 使用NIST material database

```
// Material ---> Vacuum
G4Material* Vacuum =
G4NistManager::Instance()->FindOrBuildMaterial("G4_Galactic");
```

定义材料——真空

“拿来主义”！

```
//Material --->Air
G4Material* Air =
G4NistManager::Instance()->FindOrBuildMaterial("G4_AIR");
```

定义材料——空气

```
//Material --->Fe
G4Material* Fe =
G4NistManager::Instance()->FindOrBuildMaterial("G4_Fe");
```

定义材料——金属单质铁Fe

```
//Material --->BGO
G4Material* BGO =
G4NistManager::Instance()->FindOrBuildMaterial(“G4_BGO”);
```

定义材料——BGO晶体

```
//Material --->POLYETHYLENE
G4Material* PE =
G4NistManager::Instance()->FindOrBuildMaterial("G4_POLYETHYLENE");
```

定义材料——塑料聚乙烯

```
//Material --->Nylon
G4Material* Nylon =
G4NistManager::Instance()->FindOrBuildMaterial("G4_NYLON-6-6");
```

定义材料——尼龙

Geant4中NIST材料数据库

- 元素氢～锎(H->Cf, Z=1～98)
- 自然同位素丰度，超过3000种核素
- 已预定义单质、化合物、混合物等材料

```
===================================
###   Elementary Materials from the NIST Data Base
      ================================
 Z Name       ChFormula        density(g/cm^3)       I(eV)
===================================
1    G4_H    H_2                  8.3748e-05        19.2
2    G4_He                          0.000166322      41.8
3    G4_Li                           0.534                 40
4    G4_Be                          1.848                63.7
5    G4_B                            2.37                 76
6    G4_C                            2                       81
7    G4_N    N_2                  0.0011652          82
8    G4_O    O_2                  0.00133151        95
9    G4_F                           0.00158029        115
10  G4_Ne                          0.000838505     137
11  G4_Na                          0.971                149
12  G4_Mg                          1.74                 156
13  G4_Al                            2.6989             166
14  G4_Si                            2.33                 173
```

```
==================================
###   Compound Materials from the NIST Data Base
==================================
 N Name    ChFormula      density(g/cm^3)  I(eV)
==================================
13  G4_Adipose_Tissue          0.92         63.2
        1    0.119477
        6    0.63724
        7    0.00797
        8    0.232333
       11    0.0005
       12    2e-05
       15    0.00016
       16    0.00073
       17    0.00119
       19    0.00032
       20    2e-05
       26    2e-05
       30    2e-05
4  G4_Air                   0.00120479   85.7
        6    0.000124
        7    0.755268
        8    0.231781
       18    0.012827
2  G4_CsI                   4.51         553.1
       53    0.47692
       55    0.52308
```

使用正确的名字（Name）**G4_xx**，关注密度（density）、结合能等参数

DetectorConstruction——材料定义II

自己动手，丰衣足食！

物质定义：

- isotopes <> G4Isotope
- elements <> G4Element

描述原子属性：原子序数/核子数/原子质量/壳层能量/及其他

- molecules, compounds, mixtures <> G4Material

描述物质的宏观属性：密度/状态/温度/压强/辐射长度/平均自由程/单位长度损伤等

```
//Materials defined by self
G4String symbol;
G4double a;                                          // atomic mass
G4double z;                                          // atomic number
G4double density;
G4int ncomponents, natoms;

G4Element* Cd = new G4Element("Cadmium",    symbol= "Cd", z= 48., a= 116.00*g/mole);
G4Element* Mo = new G4Element("Molybdenum", symbol= "Mo", z= 42., a= 100.00*g/mole);
G4Element* O  = new G4Element("Oxygen",     symbol= "O" , z= 8. , a= 16.00*g/mole);
G4Element* C  = new G4Element("Carbon",     symbol= "C",  z= 6.,  a= 12.00*g/mole);
G4Element* H  = new G4Element("Hydrogen",   symbol= "H",  z= 1.,  a= 1.00*g/mole);
G4Element* Br = new G4Element("Bromine",    symbol= "Br", z= 35., a= 80.00*g/mole);
G4Element* La = new G4Element("Lanthanum",  symbol= "La", z= 57., a= 139.00*g/mole);
```

```
//Material--->CdMoO4
G4Material* CdMoO4 = new G4Material("CdMoO4",  density=6.207*g/cm3, ncomponents=3);  //
CdMoO4->AddElement(Cd, natoms=1);
CdMoO4->AddElement(Mo, natoms=1);
CdMoO4->AddElement(O,  natoms=4);

//Material--->PET C10H8O4
G4Material* PET = new G4Material("PET", density=1.4*g/cm3, ncomponents=3); //PET 200um
PET->AddElement(C, natoms=10);
PET->AddElement(H, natoms=8);
PET->AddElement(O, natoms=4);
```

DetectorConstruction——材料定义II

自己动手，丰衣足食！

```
a = 14.01*g/mole;
G4Element* elN  = new G4Element(name="Nitrogen",symbol="N", z= 7., a);
a = 16.00*g/mole;
G4Element* elO  =  new G4Element(name="Oxygen",symbol="O", z= 8., a);
density = 1.290*mg/cm3;
G4Material* Air = new G4Material(name="Air", density, ncomponents=2);
Air->AddElement(elN, 70.0*perCent);
Air->AddElement(elO, 30.0*perCent);
```

定义材料——空气

```
G4Element* elC  = …;    // define “carbon” element
G4Material* SiO2 = …;   // define “quartz” material
G4Material* H2O = …;    // define “water” material
density = 0.200*g/cm3;

G4Material* aerogel = new G4Material("Aerogel",
                                     density, ncomponents=3);
aerogel->AddMaterial(SiO2,fractionmass=62.5*perCent);
aerogel->AddMaterial(H2O, fractionmass=37.4*perCent);
aerogel->AddElement (elC, fractionmass= 0.1*perCent);
```

定义材料——气凝胶

注意⚠ AddMaterial与AddElement

DetectorConstruction——材料定义实训

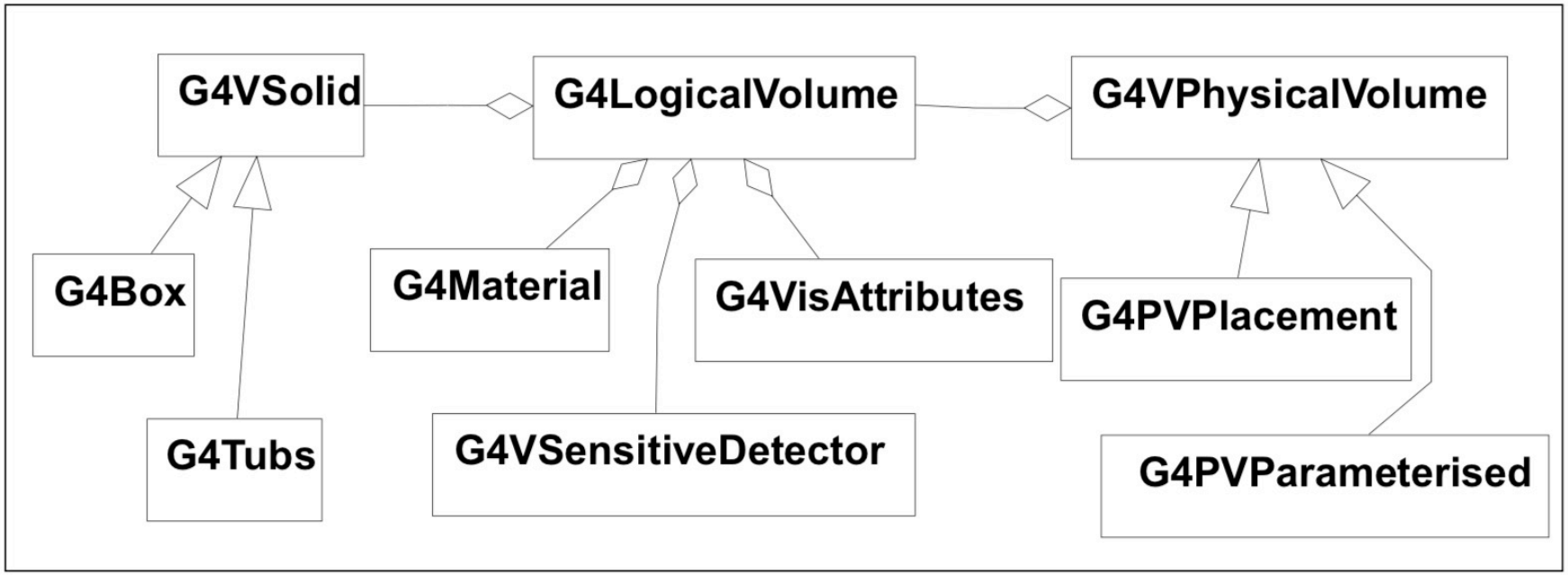
自己动手，丰衣足食！

举例：定义CLYC材料，Cs_2LiYCl_6，其中Li为富集，^6Li为90%，^7Li为10%

思路：Cs/Li/Y/Cl 使用G4Elements，其中Li为富集（使用G4isotope），后合为Material

DetectorConstruction——探测器几何体

- 定义探测器几何应明确Volume的概念，G4中探测器几何是由很多Volume构成
- 三个概念层
  - G4VSolid：形状(shape)，尺寸(size)
  - G4LogicalVolume：材料(material)，灵敏区(sensitivity)，磁场(magnetic fields)等
  - G4VPhysicalVolume：位置(position)，转动(rotation)
- 体积(Volume)调用示意图



DetectorConstruction——探测器几何体rdecay01

- 边长为2cm的立方体

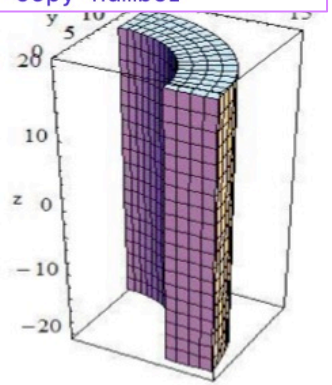
```
// World
//
//
fWorldSize = 20*mm;

G4Box*
solidWorld = new G4Box("World",                                    //its name
                  fWorldSize/2,fWorldSize/2,fWorldSize/2);//its size
                      Half x       Half y       Half z
G4LogicalVolume*
logicWorld = new G4LogicalVolume(solidWorld,                       //its solid
                                 Air,                              //its material
                                 "World");                         //its name
G4VPhysicalVolume*
physiWorld = new G4PVPlacement(0,                                  //no rotation
                               G4ThreeVector(),                    //at (0,0,0)
                               logicWorld,                         //its logical volume
                               "World",                            //its name
                               0,                                  //its mother  volume
                               false,                              //no boolean operation
                               0);                                 //copy number
```

DetectorConstruction——探测器几何体

- 内径7mm，外径14mm，厚度3.5mm，高度3.325mm材料为Ni的圆筒

```
// Ni hat  Tubs
G4double hatr0 = 0.7*cm;          // 14mm
G4double hatr  = 0.35*cm;         // Be phi = 0.7 cm, 7mm
G4double hath  = 0.3325*cm;       // distance2(1.4mm)+Be(12.5um)+0.5*SicrystalsizeC(500um)  *2 =

G4double hatZ  = 0.5*(polyfilmsizeC+hath);

G4Tubs*
solidHat = new G4Tubs("Hat",                                          // its name
              hatr, hatr0, 0.5*hath, 0.*deg, 360.*deg );              // its size


G4LogicalVolume*
logicHat = new G4LogicalVolume(solidHat,                              // its solid
                               Ni,                                    // its material
                               "Hat");                                // its name

              new G4PVPlacement(0,                                    // no rotation
                               G4ThreeVector(0,0,hatZ),               // at (0,0,0)
                               logicHat,                              // its logical volume
                               "Hat",                                 // its name
                               logicWorld,                            // its mother  volume
                               false,                                 // no boolean operation
                               0);                                    // copy number
```

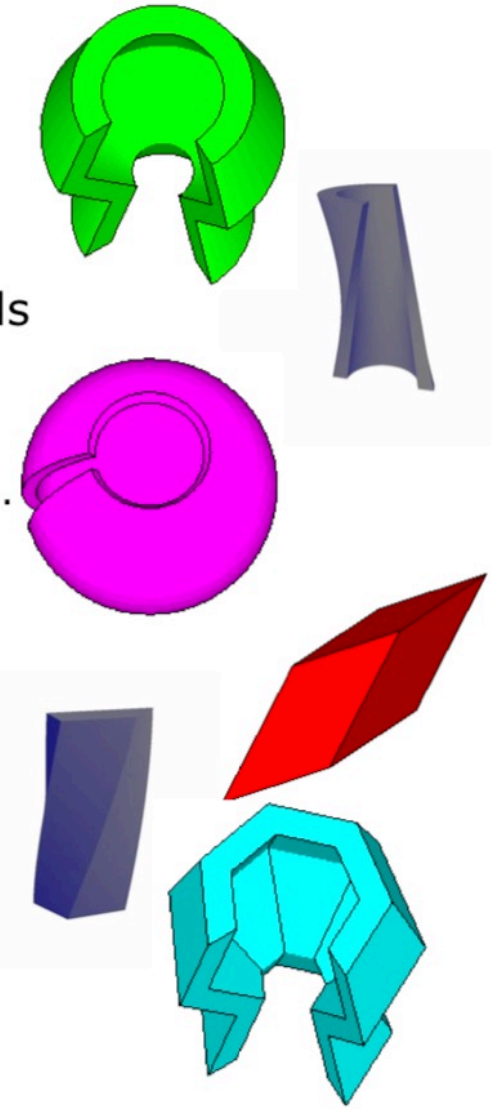
```
G4Tubs(const G4String& pname,  // name
             G4double  pRmin,  // inner radius
             G4double  pRmax,  // outer radius
             G4double  pDz,    // Z half length
             G4double  pSphi,  // starting Phi
             G4double  pDphi); // segment angle
```



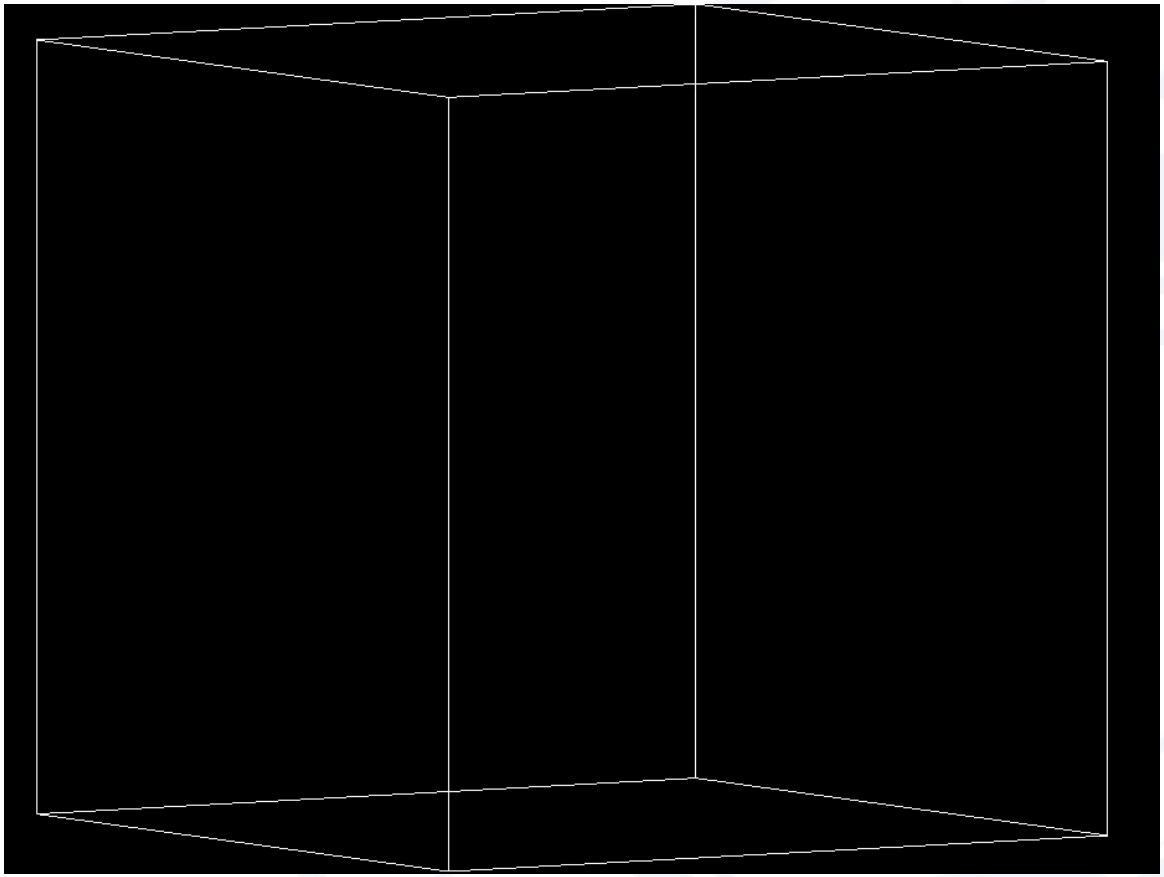
DetectorConstruction——探测器几何体solids

Solids

- Solids defined in Geant4:
  - CSG (Constructed Solid Geometry) solids
    - G4Box, G4Tubs, G4Cons, G4Trd, …
  - Specific solids (CSG like)
    - G4Polycone, G4Polyhedra, G4Hype, …
    - G4TwistedTubs, G4TwistedTrap, …
  - BREP (Boundary REPresented) solids
    - G4BREPSolidPolycone, G4BSplineSurface, …
    - Any order surface
  - Boolean solids
    - G4UnionSolid, G4SubtractionSolid,



DetectorConstruction——探测器几何体rdecay01运行

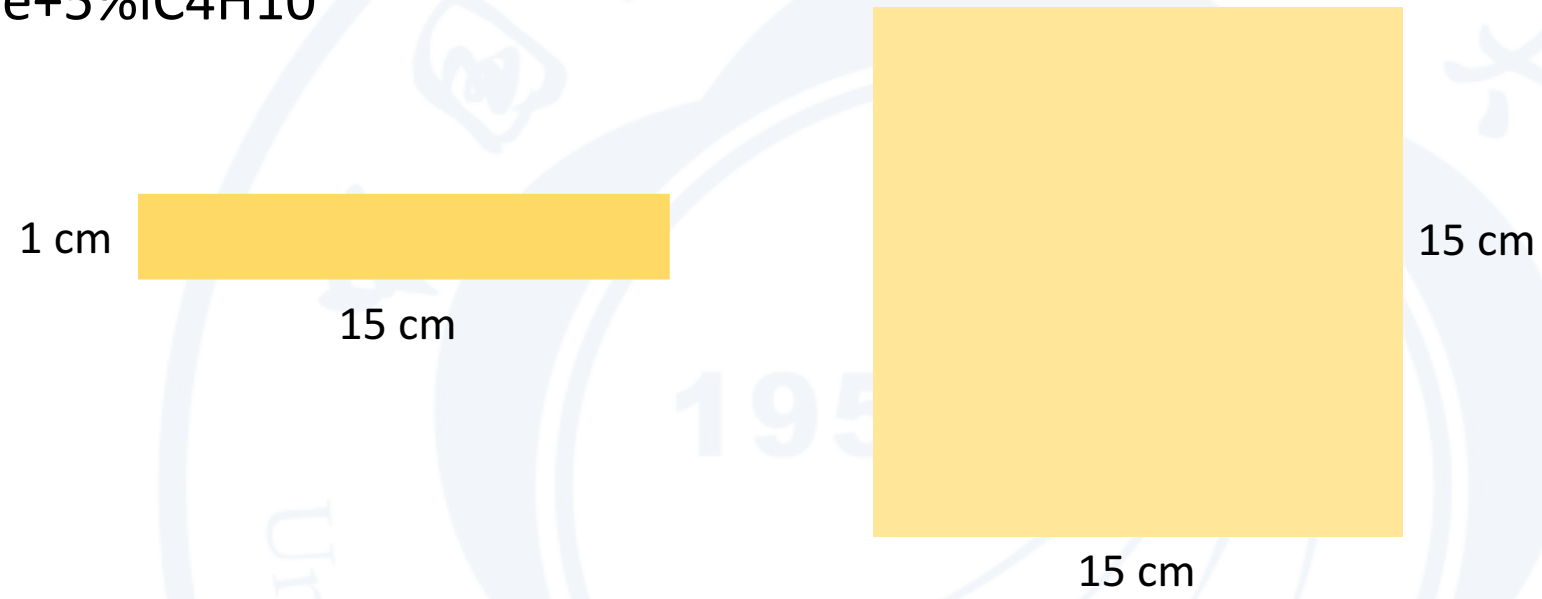


可尝试运行例子rdecay01

```
$ mkdir rdecay01-build
$ cd rdecay01-build
$ cmake ../rdecay01
…
$ make
….
$ ./rdecay01
```

DetectorConstruction——探测器几何体实训

举例：在rdecay01例子内在世界体内部搭建探测器框架，微结构气体探测器MMD，探测器大小15cm*15cm，气隙厚度1cm，气体为95%Ne+5%iC4H10



注意⚠：世界体应为最大

学习/使用Geant4的几条建议：

1. 不需要知道所有的点，你也没法知道所有的点，主线拎清楚，在之后的应用中遇到问题解决问题，需要什么学什么；

2. 遇到问题先自己思考，自己想办法解决，多查阅资料；不能解决的，在自己有一定思考的基础上，向师兄师姐老师们请教，你的坑也许是前人也淌过的坑；

3.逻辑拎清楚，多打印cout一些东西出来，写几行就打印，写几行就打印，拎清楚各“模块”之间调用的逻辑；

4.在做科研模拟小任务前先想清楚，要模拟什么，需要什么信息，这个课题是否有例子可以参考（其实你总会找到参考的例子），不需要你写完整的程序，只是在“前人”的基础上加入自己的东西；

5.检查模拟结果是否正确，来来回回反反复复多次检查/交叉检查等。。。

欲知后事如何 请待下次（II）

谢谢捧场 ~

PhysicsList