清華大學
Tsinghua University

粒 子 物 理 模 拟

第三讲

王喆
清华大学

本节内容

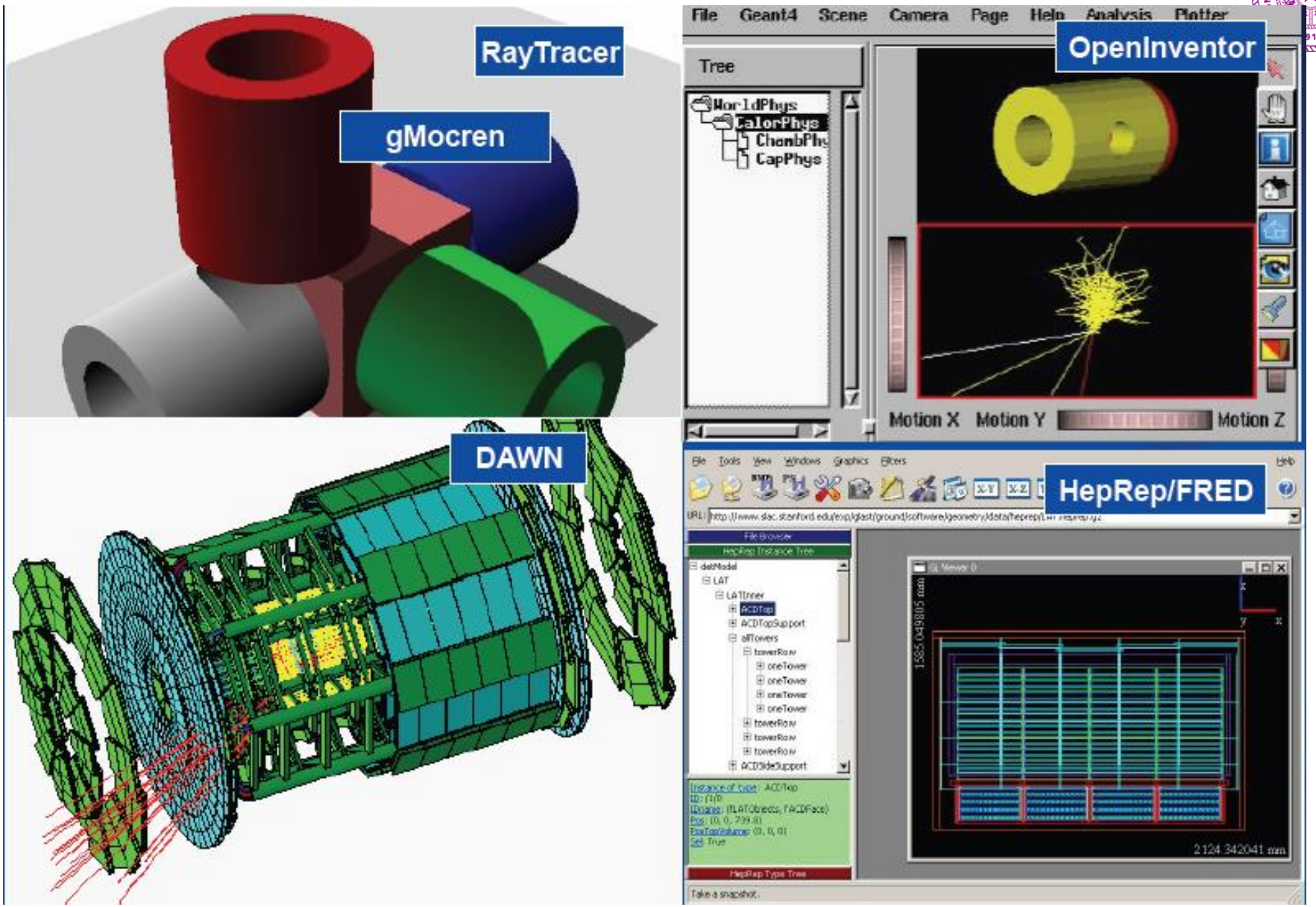
- 各种可视化程序的简介
- 探测器几何
- 探测器几何参数化
- 磁场设置
- 粒子设置
- 物理过程设置
- 设置起始粒子
- 作业

各种可视化方案

各种可视化程序

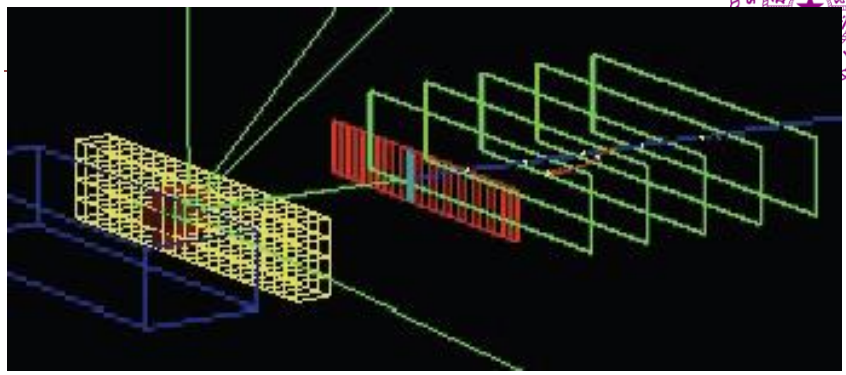
- Geant4提供很多图形驱动：OpenGL，OpenInventor，HepRep，DAWN，VRML，RayTracer，gMocren，ASCIITree
- 可以用来动态的查看，例如 Qt+OpenGL
- 有的只能生成文件，例如DAWN
- 还有的只能生成探测器文本输出，例如Text
- 有的方便的支持网络传输，例如HepRep

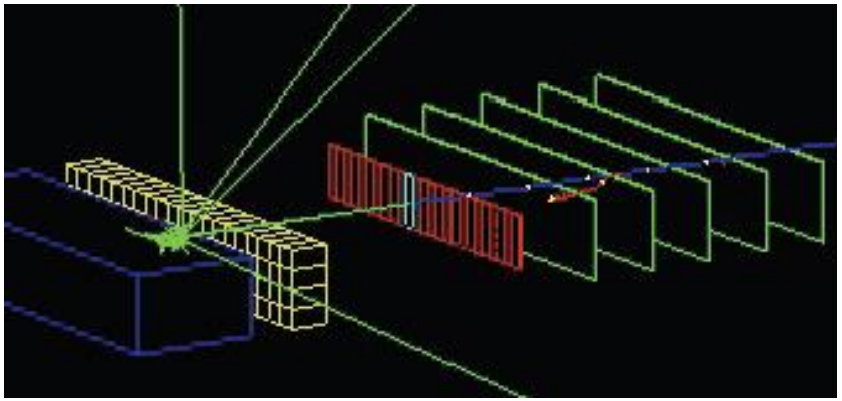
- 注意有些可视化程序需要cmake配置的时候设定：
  cmake -DGEANT4_BUILD_MULTITHREADED=ON
  -DGEANT4_USE_RAYTRACER_X11=ON
  -DGEANT4_USE_QT=ON
  -DGEANT4_USE_NETWORKDAWN=ON
  -DGEANT4_USE_OPENGL_X11=ON
  -DGEANT4_INSTALL_DATA=ON
  -DCMAKE_INSTALL_PREFIX=~/geant4.10.00.p02-install
  ~/studio/sw/geant4.10.00.p02

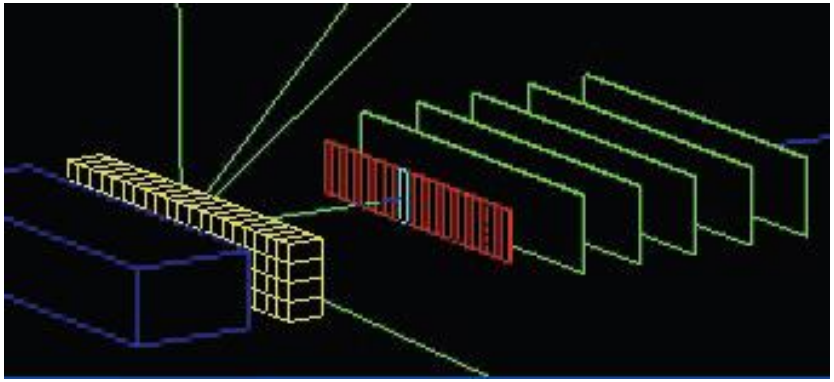

RayTracer
gMocren
DAWN
File Geant4 Scene Camera Page Help Analysis Plotter
Tree
WorldPhys
CalorPhys
ChambPhy
CapPhys
OpenInventor
Motion X Motion Y Motion Z
File Tools View Windows Graphics Filters Help
HepRep/FRED
File Browser
HepRep Instance Tree
detModel
LAT
LATInner
ACDTop
ACDTopSupport
allTowers
towerRow
oneTower
ACDSideSupport
HepRep Type Tree
Take a snapshot.

OpenGL

- 使用
  /vis/open OGL
  /vis/drawVolume
- 文件输出
  /vis/ogl/printEPS
- 输出风格
  /vis/ogl/set/printMode
  /vis/ogl/set/transparency
- 隐线处理
  /vis/viewer/set/hiddenEdge 1
  /vis/viewsr/set/hiddenMarker 1







Text输出

- 最基本的错误检查，或在没有任何可视化程序可用的时候，比如，网络极慢，但又想做一点调试工作。
  用法：
  /vis/open ATree
  /vis/drawVolume
  /vis/viewer/flush
- 它不出径迹，只打印出目录结构
  例如：N02
- 输出如下，
  "World":0
  "Target":0
  "Tracker":0
  "Chamber":0-4 (5 parametrised volumes)

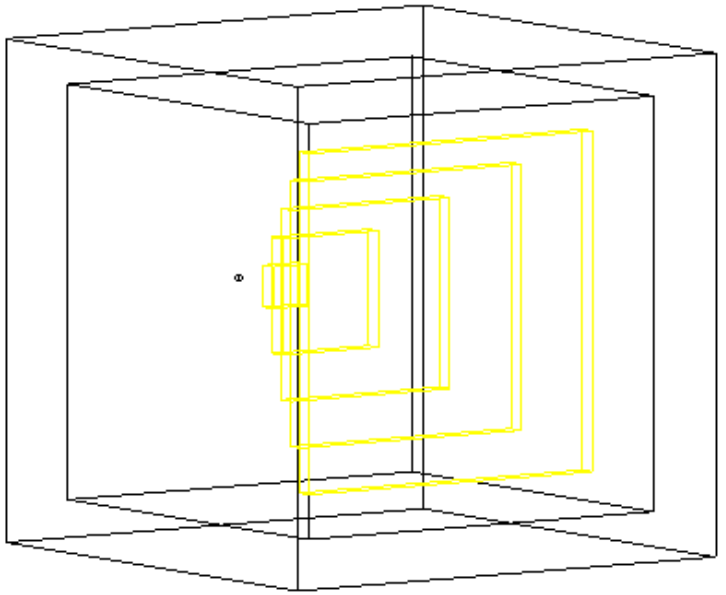
DAWN

- DAWN只有文件输出，最高质量的技术渲染，非常适合用于技术文章的发表用图
- DAWN生成两种文件.eps或者.prim
- DAWN有两种模式，一种为本机输出，一种为网络模式，可以处理远程的数据。
- 用法

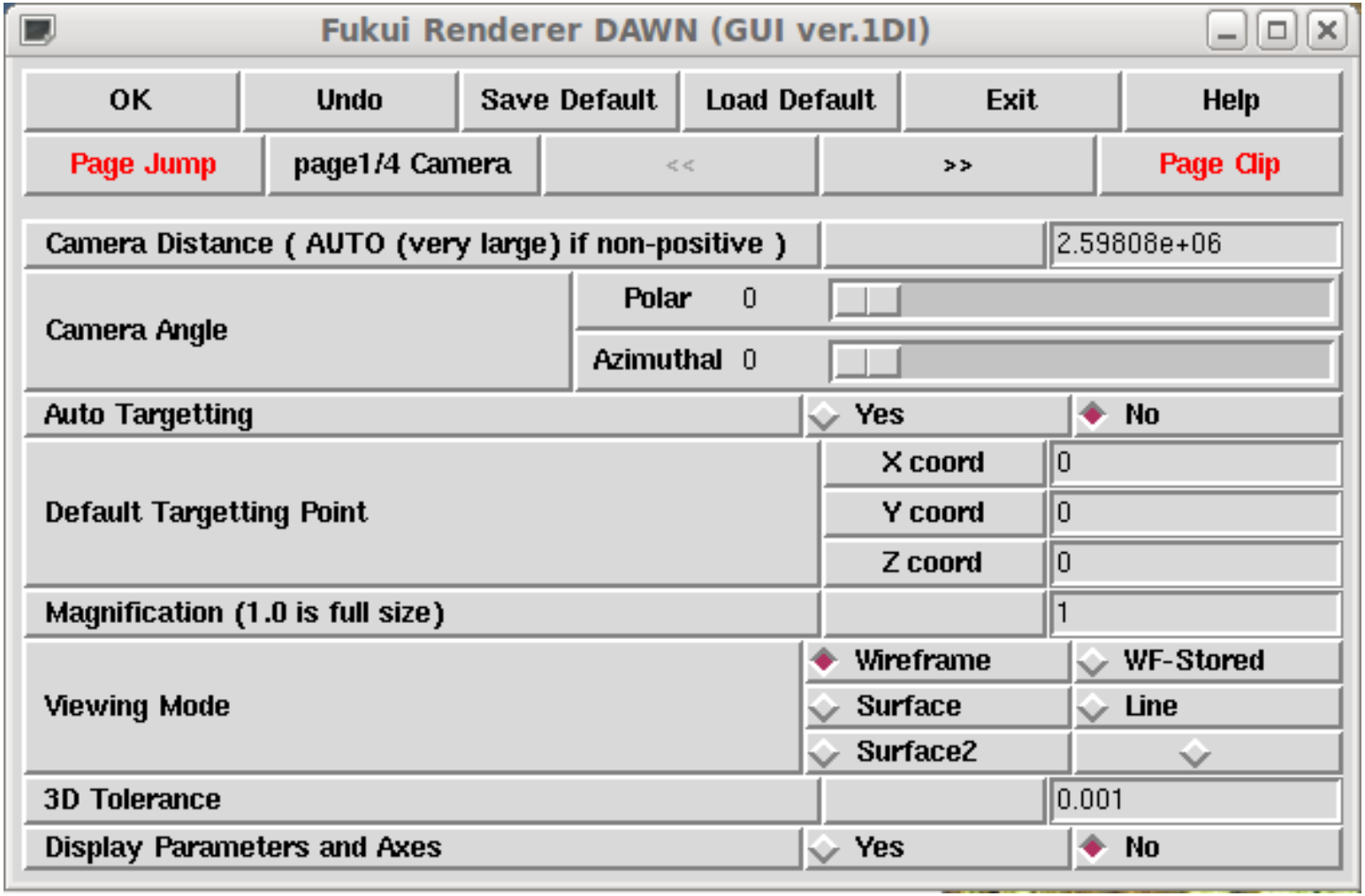
  /vis/open　DAWNFILE

  /vis/drawVolume

  /vis/viewer/flush



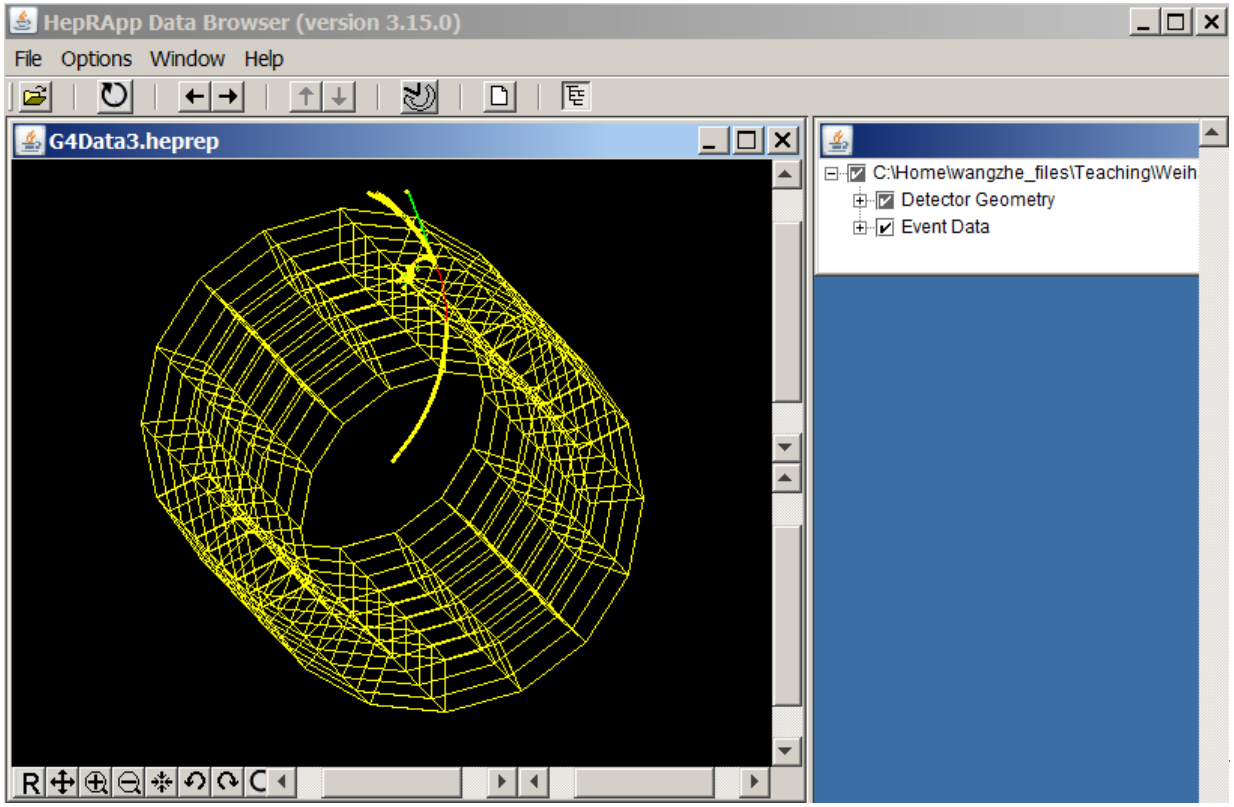
DAWN的配置界面


Fukui Renderer DAWN (GUI ver.1DI)
OK
Undo
Save Default
Load Default
Exit
Help
Page Jump
page1/4 Camera
<<
>>
Page Clip
Camera Distance (AUTO (very large) if non-positive)
2.59808e+06
Camera Angle
Polar 0
Azimuthal 0
Auto Targetting
Yes
No
Default Targetting Point
X coord
0
Y coord
0
Z coord
0
Magnification (1.0 is full size)
1
Viewing Mode
Wireframe
WF-Stored
Surface
Line
Surface2
3D Tolerance
0.001
Display Parameters and Axes
Yes
No

HepRep

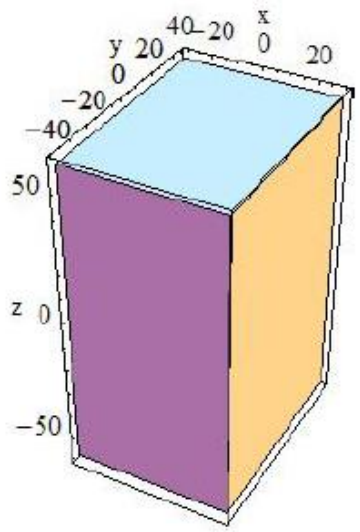
http://geant4.slac.stanford.edu/Presentations/vis/G4HepRAppTutorial/G4HepRAppTutorial.html

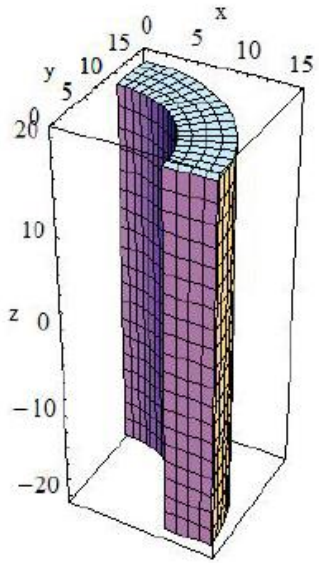
- /vis/open HepRepFile
- 生成HepRep文件
- 需要运行HepRep浏览器HepRApp，需要安装Java SDK。
- 可以显示网络上的数据文件。

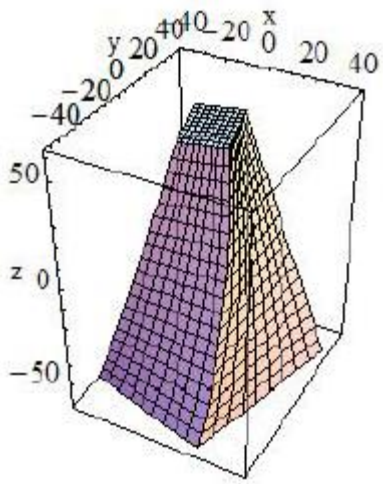


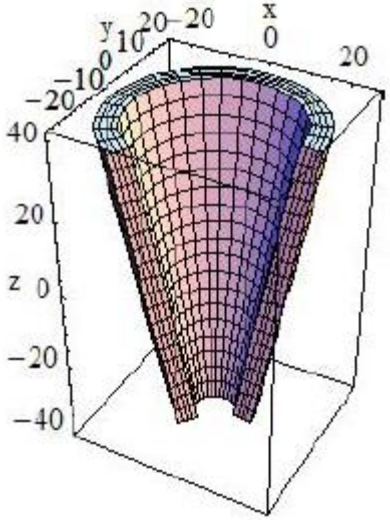
探测器几何

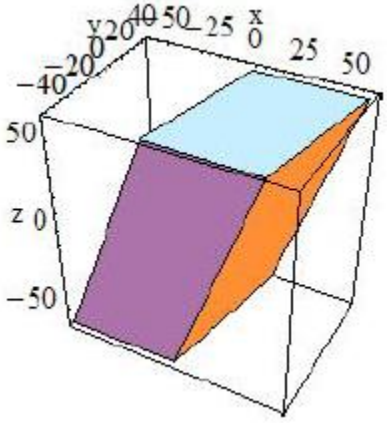
各种几何形状

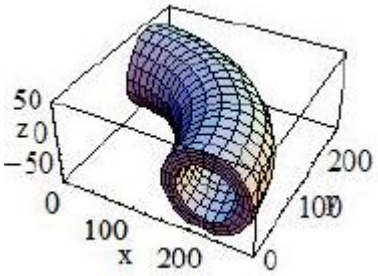


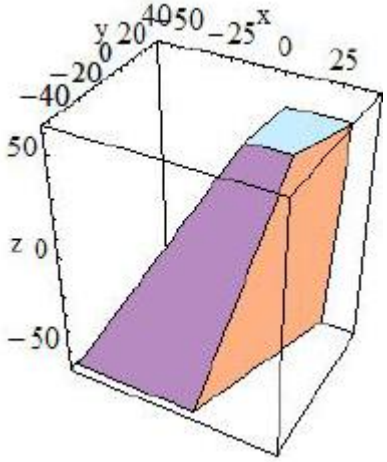


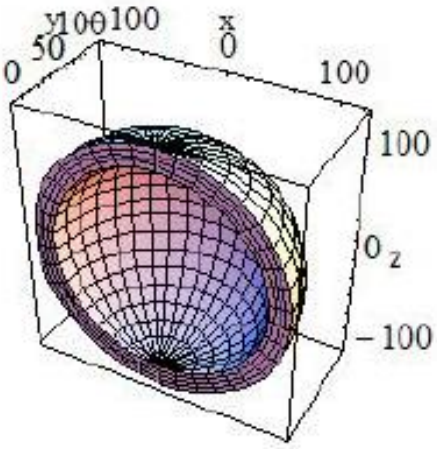






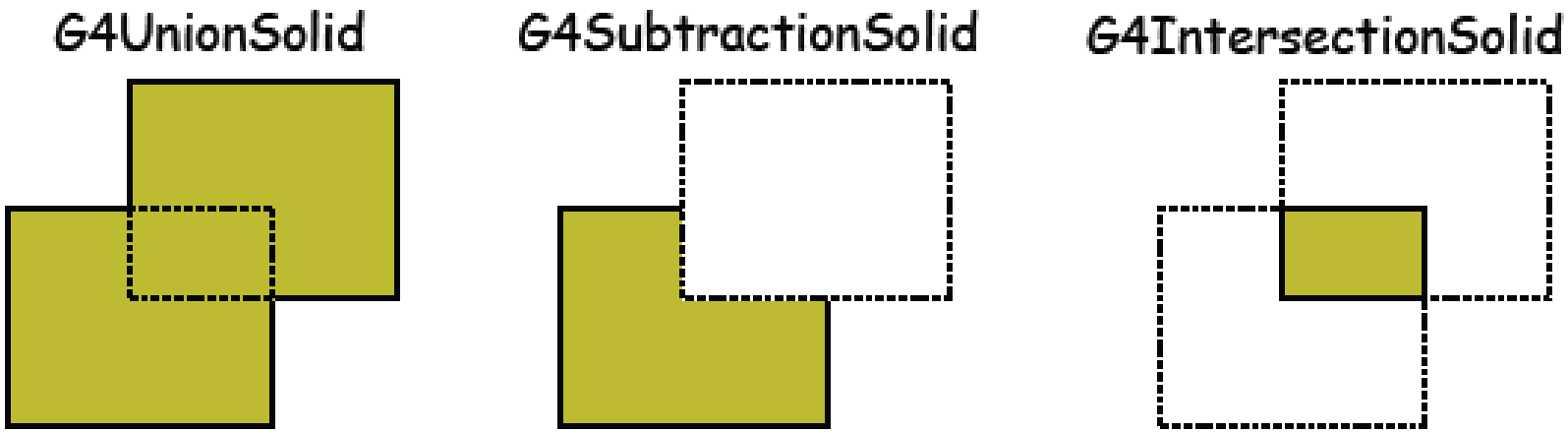




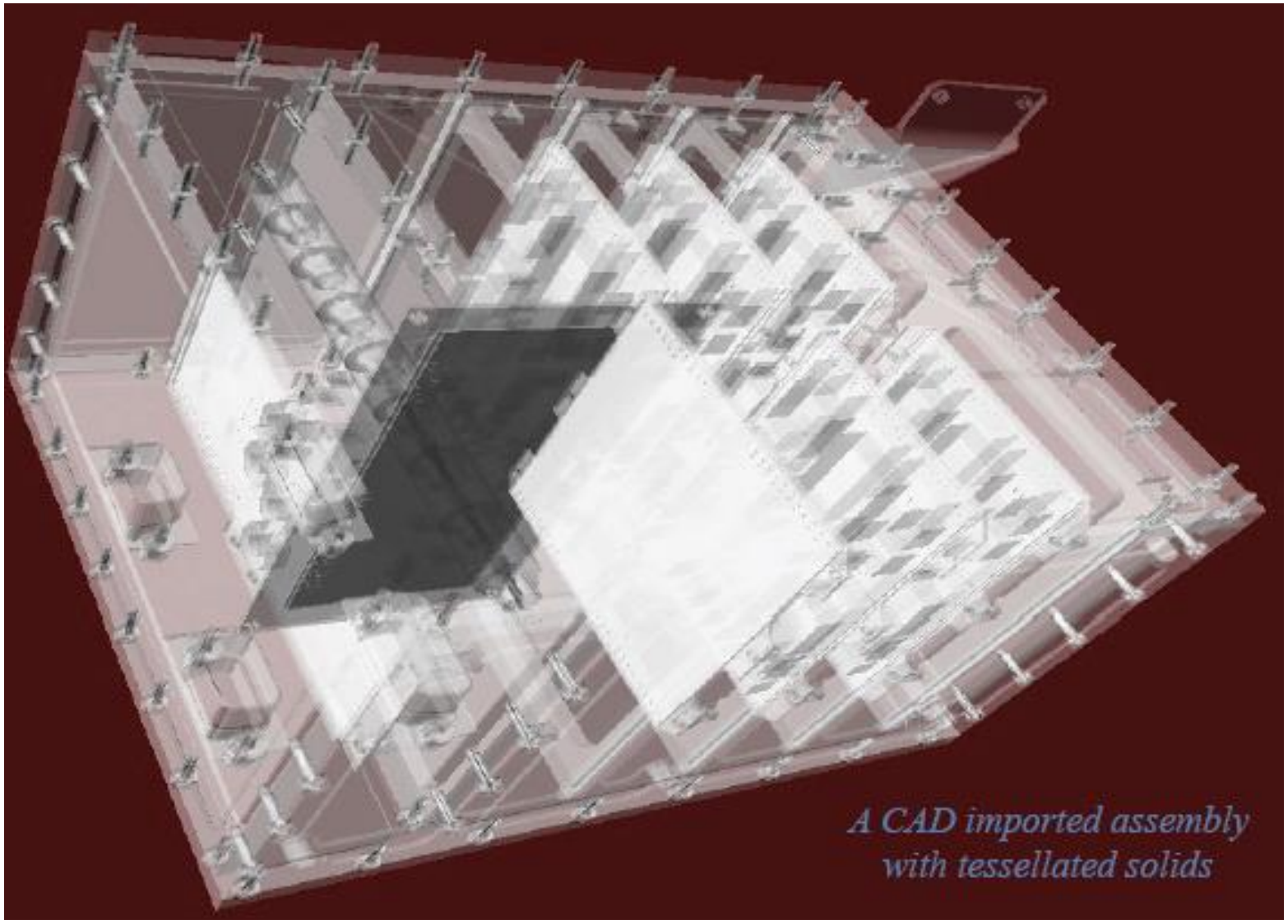


几何的逻辑操作

- 几何可以：
  联合，相减，相交
- 两个几何体先操作，然后可以与第三个几何
- 从而达到构造复杂几何体的目标



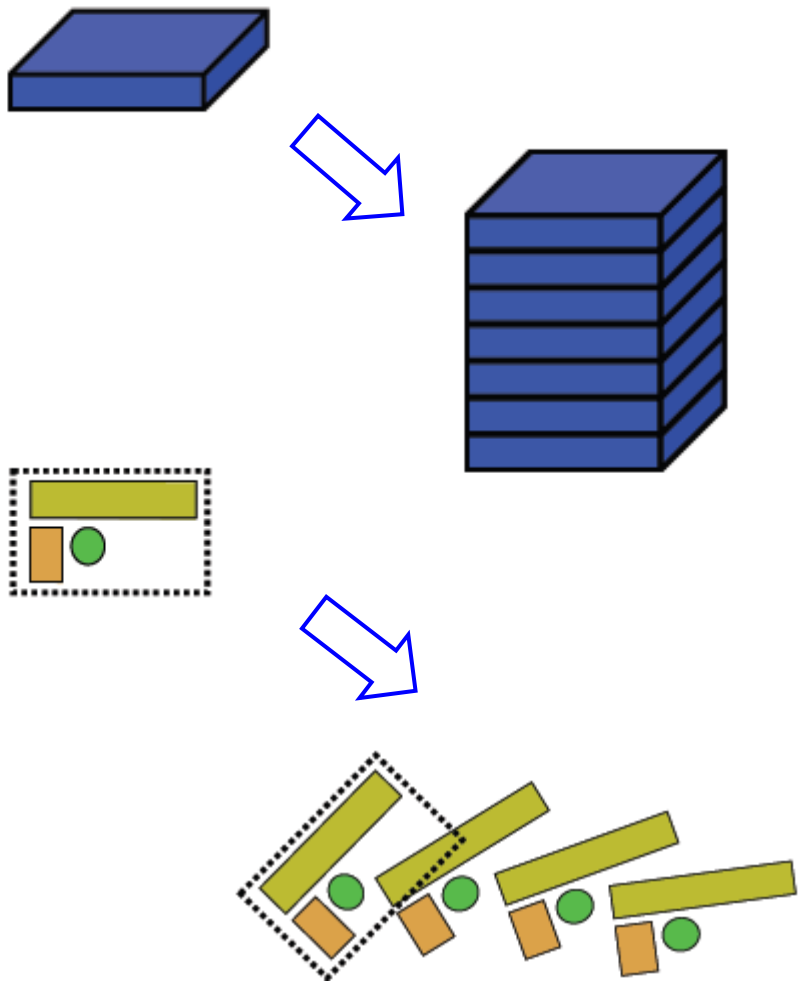
与AUTOCAD联合



探测器几何参数化

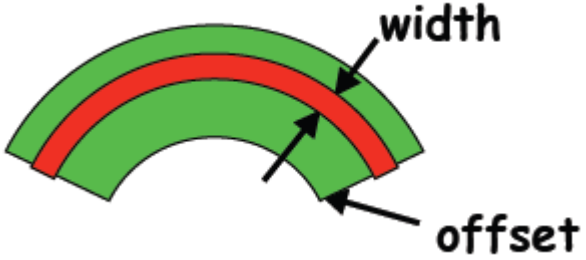
探测器坐标位置、朝向的参数化

- 粒子物理探测器经常由很多全同的小模块堆叠而成
- 模拟的时候也可以如此操作，减少冗长、重复的操作
- 这种复制可以沿x-y-z方向，延着柱坐标的rho方向，沿着极坐标的theta和phi方向
- 甚至完全不规则的排布

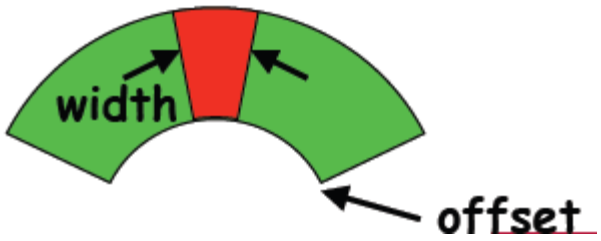


探测器大小的参数化

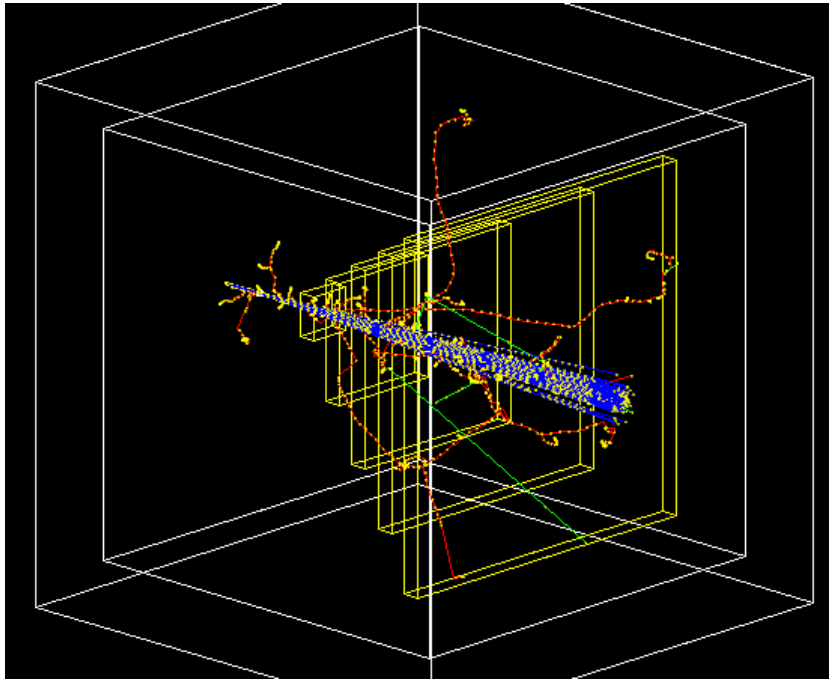
- 除了位置、方向，其大小尺寸也可以参数化
  - 用众多壳型探测器，堆叠而成



  - 用众多扇型探测器，堆叠而成



  - N02例子内也有5个参数化的探测器



N02例子，参数化实现

在ExN02DectoctorConstruction.cc中

// 1. Solid与logic volume的构造

```
solidChamber = new G4Box("chamber", 100*cm, 100*cm, 10*cm);
logicChamber = new G4LogicalVolume(solidChamber,ChamberMater,"Chamber",0,0,0);

G4double firstPosition = -trackerSize + 0.5*ChamberWidth;
G4double firstLength = fTrackerLength/10;
G4double lastLength  = fTrackerLength;
```

// 2. 参数化将由Ex02ChamberParameterisation来控制

```
chamberParam = new ExN02ChamberParameterisation(
                        NbOfChambers,        // NoChambers
                        firstPosition,       // Z of center of first
                        ChamberSpacing,      // Z spacing of centers
                        ChamberWidth,        // Width Chamber
                        firstLength,         // lengthInitial
                        lastLength);         // lengthFinal
```

// 3. 实现有由Ex02ChamberParameterisation来控制的Physics Volume

```
physiChamber = new G4PVParameterised(
                         "Chamber",      // their name
                         logicChamber,   // their logical volume
                         logicTracker,   // Mother logical volume
                         kZAxis,         // Are placed along this axis
                         NbOfChambers,   // Number of chambers
                         chamberParam);  // The parametrisation
```

N02例子参数化的方式

- 在ExN02ChamberParameterisation.cc

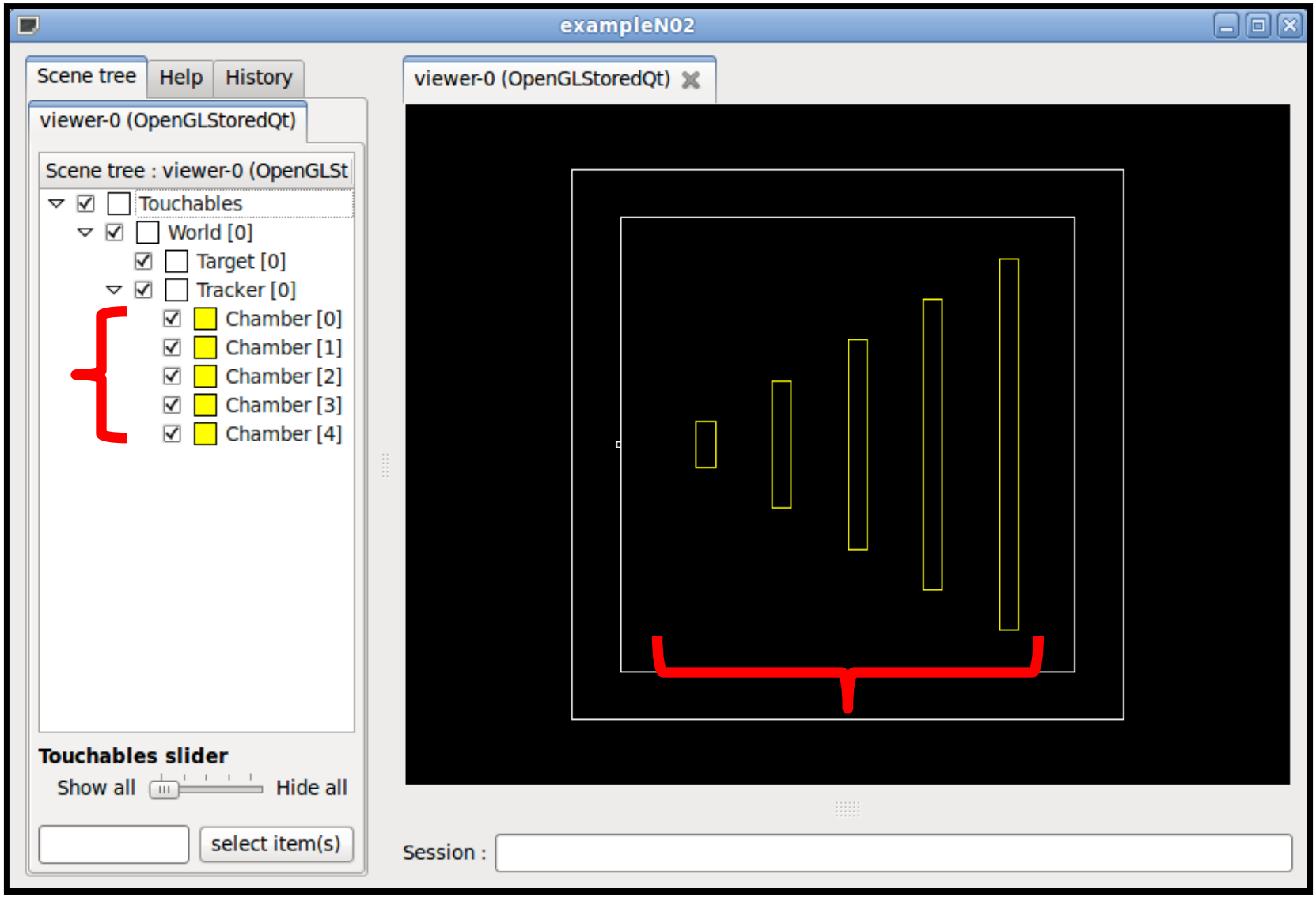
// 4. 对每一个Physics Volume计算新的位置，方向

```
void ExN02ChamberParameterisation::ComputeTransformation
(const G4int copyNo, G4VPhysicalVolume* physVol) const
{
  G4double       Zposition= fStartZ + (copyNo+1) * fSpacing;
  G4ThreeVector origin(0,0,Zposition);
  physVol->SetTranslation(origin);
  physVol->SetRotation(0);
}
```

// 5. 对每一个Physics Volume计算新的尺寸

```
void ExN02ChamberParameterisation::ComputeDimensions
(G4Box& trackerChamber, const G4int copyNo, const G4VPhysicalVolume*) const
{
  G4double  halfLength= fHalfLengthFirst + copyNo * fHalfLengthIncr;
  trackerChamber.SetXHalfLength(halfLength);
  trackerChamber.SetYHalfLength(halfLength);
  trackerChamber.SetZHalfLength(fHalfWidth);
}
```

顺序号码及结果：Chamber0, 1, 2, 3, 4



几何的注意事项

- 子实体物质将覆盖母实体的物质
- 但同级的几何不能覆盖
- 会引起Geant4无法判断粒子在哪个探测器内，无法确定物质和边界
- 这样的程序经常会出现异常的运行状况，突然中断，无线循环，死机等。
- 请参考Geant4手册中关于如何侦测

探测器磁场

磁场

- 可以是全局磁场

```
G4UniformMagField* magField
  = new G4UniformMagField(G4ThreeVector(0.,0.,fieldValue));
```

```
G4FieldManager* fieldMgr
  = G4TransportationManager::GetTransportationManager()
    ->GetFieldManager();
   fieldMgr->SetDetectorField(magField);
```

- 也可以给某一个探测器加磁场

粒子设置

Geant4 中的粒子

- **Geant4 中的粒子由三层类来表示。**
- G4ParticleDefinition
  - 粒子的“静态”特征量，如电荷、质量、寿命等等。
  - 没有能量、方向等信息
- G4DynamicParticle
  - 赋予粒子运动学(动态)属性，如动量，能量，自旋方向等等。
- G4Track
  - 将动态粒子放到具体环境中，给出位置，几何信息等等。

粒子定义(1)

- 首先要定义粒子，即模拟中可能产生的各种粒子
- Geant4提供了各种类型的粒子：

  1.普通粒子：如电子、质子、光子等

  2.共振态粒子：寿命短，如矢量介子等

  3.核子：如氘核、氦核及重离子等

  4.夸克、胶子等
- 定义附带了粒子的各种信息：如名称、质量、电荷、自旋、寿命、衰变道等

粒子定义(2)

- Geant4中粒子分以下六大类

  lepton

  meson

  baryon

  boson

  shortlived

  ion

粒子定义(3)

- PhysicsList中定义粒子

  在ConstructParticle()函数中定义

```
void ExN01PhysicsList::ConstructParticle()
{
  G4Geantino::GeantinoDefinition();//定义geantino
  G4Proton::ProtonDefinition();//定义质子
  G4Positron::PositronDefinition();//正电子
  G4MuonPlus::MuonPlusDefinition();//μ+
  G4AntiNeutrinoE::AntiNeutrinoEDefinition();//反电子中微子
  …
}
```

但是如果过程复杂，需要定义的粒子非常多，需要有更方便的定义方法

粒子定义(4)

```
void ExN01PhysicsList::ConstructLeptons()
{
  // 定义所有轻子
  G4LeptonConstructor pConstructor;
  pConstructor.ConstructParticle();
}
 void ExN01PhysicsList::ConstructBosons()
{
  // 定义所有玻色子
  G4BosonConstructor pConstructor;
  pConstructor.ConstructParticle();
}
 …
```

```
void ExN01PhysicsList::ConstructParticle()
{
  ConstructLeptons();//构造轻子
  ConstructBosons(); //构造玻色子
  …
}
```

除了轻子、玻色子还包括：
G4MesonConstructor
G4BaryonConstructor
G4IonConstructor
G4ShortlivedConstructor

如果对过程中可能需要的粒子不确定，可以用这种办法把所有粒子都构造出来。

物理过程设置

Geant4 的物理过程

- 要模拟真实的物理，必须首先知道粒子在物质中哪些相互作用是最主要的，或者说哪些物理过程是重要的。**Geant4**提供了**7**大类物理过程描述粒子与物质的相互作用。**$G4INSTALL/data**目录存放物理模型的数据
  - electromagnetic:电磁相互作用过程(标准的和低能的)
  - hadronic:强子相互作用过程(纯强子、辐射衰变)
  - decay:衰变过程
  - photolepton-hadron:光、轻子与强子的相互作用过程
  - optical:光学光子过程
  - parameterization:参数化过程(即fast simulation)
  - transportion:输运过程

要根据事例中的粒子以及材料，指定必要的物理过程
其中输运过程是必须添加的过程。

物理过程的添加

```
void ExN01PhysicsList::ConstructProcess()
{
  AddTransportation(); //添加输运过程
}
```

```
void ExN02PhysicsList::ConstructProcess()
{
  AddTransportation(); //添加输运过程
  //添加电磁过程，自定义，
  //见void ExN02PhysicsList::ConstructEM()
  ConstructEM();
  //添加一般过程(实际上是衰变过程)，自定义，
  //见void ExN02PhysicsList::ConstructGeneral()
  ConstructGeneral();
}
```

设定起始粒子信息

产生Primary Event

- 必须指定如何产生一个事件，才能进行模拟，在G4VUserPrimaryGeneratorAction的具体类中用G4VPrimaryGenerator的具体类来完成。
- 有两种PrimaryGenerator

G4ParticleGun:发射指定能动量的特定粒子

G4HEPEvtInterface:利用提供的接口，读取外部产生子产生的事例。外部产生子的结果按照HEPEvt的格式写成ASCII文件

G4ParticleGun

```
//参数n_particle表示一次发射的粒子数目
G4ParticleGun* particleGun = new G4ParticleGun(n_particle);

G4ParticleGun有很多设置函数，如：
SetParticleDefinition(G4ParticleDefinition*); //粒子类型
SetParticleMomentumDirection(G4ThreeVector); //动量方向
SetParticleEnergy(G4double);  //能量
SetParticlePosition(G4ThreeVector); //发射位置
...
```

粒子枪的属性设置好之后，才调用generatePrimaryVertex()函数，产生事例的主顶点。

粒子枪本身不提供随机性，发射的粒子都是指定的。如果需要按照某分布随即发射粒子，需要在调用generatePrimaryVertex()之前，利用Geant4提供的随机数产生子自己写出需要的分布。

参见ExN01PrimaryGeneratorAction::generatePrimaries(G4Event*)函数

初始粒子

- 在ExN02PrimaryGeneratorAction.cc

// 1. 拿到质子的定义，设置初始方向，能量

```
G4int n_particle = 1;
particleGun = new G4ParticleGun(n_particle);

/ default particle

G4ParticleTable* particleTable = G4ParticleTable::GetParticleTable();
G4ParticleDefinition* particle = particleTable->FindParticle("proton");

particleGun->SetParticleDefinition(particle);
particleGun->SetParticleMomentumDirection(G4ThreeVector(0.,0.,1.));
particleGun->SetParticleEnergy(3.0*GeV);
```

// 2. 设置初始位置

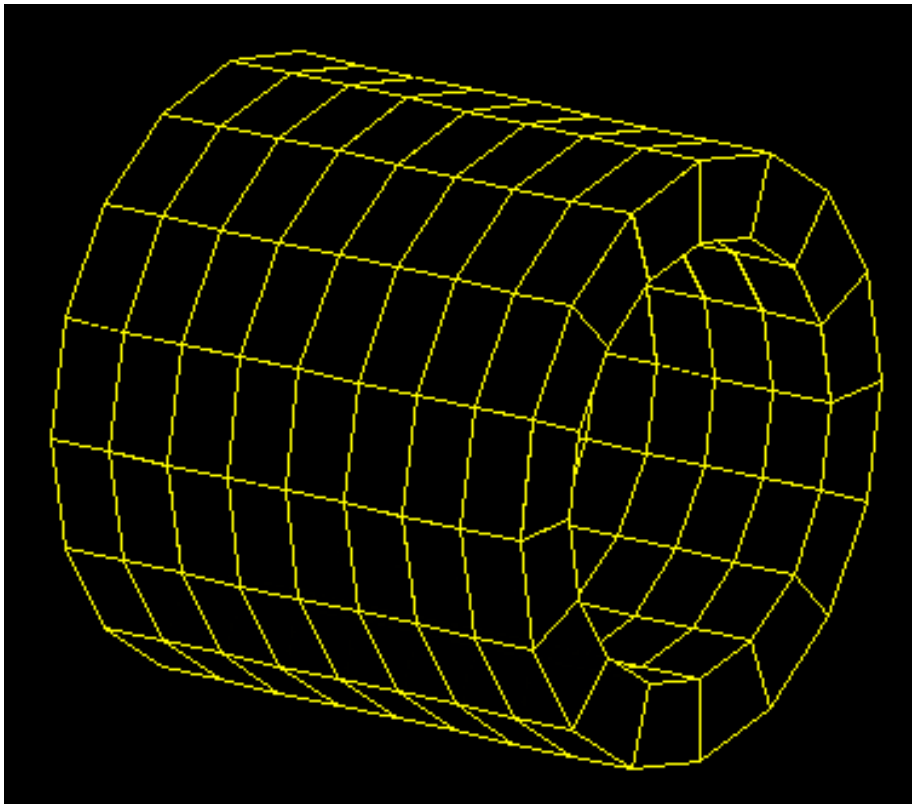
```
G4double position = -0.5*(myDetector->GetWorldFullLength());
particleGun->SetParticlePosition(G4ThreeVector(0.*cm,0.*cm,position));

particleGun->GeneratePrimaryVertex(anEvent);
```

作业

生成这样一个简单的探测器：

- 由一组晶体组成一桶装的量能器，提示：由G4Tubs构成
- 晶体物质为CsI
- Phi向有20组
- Z向也有20组
- R_in为0.8米
- R_out为1.1米
- Z向2米
- 加0.5 T延轴向的磁场



作业

- 从探测器的几何中心向外射出一个质子，打在该量能器上，模拟该粒子。
- 成功运行，程序不出错。