Material and Geometry I

歳藤利行
KEK/JST

ねらい

- Geant4 – 放射線の伝播をシミュレーション
- どこを伝播するのか？
  - 検出器、ビームライン、遮蔽体、人体など
- これらを物体の要素(volume)の組み合わせで定義する
- volumeが持つべき情報
  - 何でできているか？(物質)
  - 形状、寸法は？
  - どこに、どのような向きに置かれているのか？

- 物質、ジオメトリーの定義の仕方の初歩について講義します。

Contents

- 物質の定義
- NIST materials DB
- G4VUserDetectorConstruction class
- Geometry definition
- Solid, logical volume, physical volume (placement), world volume
- Geometry collision detection

物質の定義

Geant4における物質

- 物質(化合物、混合物)は元素からできている
  元素は同位体からできている

  → 3つの主なクラス
  - 同位体 ←→ G4Isotope
  - 元素 ←→ G4Element
  - 分子、化合物、混合物 ←→ G4Material

- G4Isotope と G4Element は原子レベルでの特性を記述する:
  - 原子番号、核子の数、分子量、エネルギー準位
  - 原子ごとの断面積など

- G4Material は物質のマクロな性質を記述する:
  - 温度、圧力、状態、密度
  - 放射長、吸収長など

- G4Material クラス
  - 検出器などの定義に使われる
  - tracking, geometry, physicsから見ることができる
  - 関連する元素や元素をつくる同位体に関する全ての情報を含む

単純な物質

- 単一の元素からなる物質
  - 名前、密度、1モルあたりの質量、原子番号で定義

```
double density = 1.390*g/cm3;
double a = 39.95*g/mole;
G4Material* mat_lAr = new
  G4Material("liquidArgon",z=18.,a,density);
```

物質: 分子・化合物

- 元素からなる物質
  - 分子中の原子数(整数)で定義 (化学式)

```
a = 1.01*g/mole;
G4Element* ele_H = new
  G4Element("Hydrogen",symbol="H",z=1.,a);
a = 16.00*g/mole;
G4Element* ele_O = new G4Element("Oxygen",symbol="O",z=8.,a);
density = 1.000*g/cm3;
G4Material* H2O = new G4Material("Water",density, ncomp=2);
G4int natoms; //整数
H2O->AddElement(ele_H, natoms=2);
H2O->AddElement(ele_O, natoms=1);
```

- 注意
  - CH_2 は C_4H_8 ,,, などと等価

物質: 混合物

- 元素からなる物質
  - 物質を構成する元素の質量比(実数)で混合比を定義する例

```
a = 14.01*g/mole;
G4Element* ele_N = new
    G4Element(name="Nitrogen",symbol="N",z=7.,a);
a = 16.00*g/mole;
G4Element* ele_O = new
    G4Element(name="Oxygen",symbol="O",z=8.,a);
density = 1.290*mg/cm3;
G4Material* Air = new
    G4Material(name="Air",density, ncomponents=2);
G4double fracMass; //実数
Air->AddElement(ele_N, fracMass=70.0*perCent);
Air->AddElement(ele_O, fracMass=30.0*perCent);
```

物質: 混合物

- あらかじめ定義された物質や元素からなる物質
  - 各要素の質量比で定義

```
G4Element* ele_C  = …;   // define “carbon” element
G4Material* SiO2 = …;  // define “quartz” material
G4Material* H2O = …;   // define “water” material


density = 0.200*g/cm3;
G4Material* Aerog = new
   G4Material("Aerogel",density,ncomponents=3);
Aerog->AddMaterial(SiO2,fractionmass=62.5*perCent);
Aerog->AddMaterial(H2O ,fractionmass=37.4*perCent);
Aerog->AddElement (ele_C ,fractionmass= 0.1*perCent);
```

ユーザーによって同位体の存在比を定義した元素

- 必要に応じて、元素はあらかじめ定義された同位体から作ることができる
  - 各々の同位体の存在比で定義

- 例、原子力発電のための濃縮ウラン

```
G4Isotope* iso_U235 = new G4Isotope(“U235", iz=92, ia=235,
                                    a=235.0439242*g/mole);
G4Isotope* iso_U238 = new G4Isotope(“U238", iz=92, ia=238,
                                    a=238.0507847*g/mole);

G4Element* ele_enrichedU = new G4Element(“enriched U",
   symbol=“U" ,                                  ncomponents=2);

ele_enrichedU->AddIsotope(iso_U235, abundance=5.*perCent);
ele_enrichedU->AddIsotope(iso_U238, abundance=95.*perCent);
                                                  ← 個数比

G4Material* mat_enrichedU =
   new G4Material(“U for nuclear power generation" , density=
   19.050*g/cm3 ,  ncomponents = 1 , kStateSolid );
mat_enrichedU->AddElement( ele_enrichedU , fractionmass = 1 );
```

G4Material の属性

- G4Materialに関連付けられた属性
  - 温度、圧力、状態、密度
  - 物理過程を決めるパラメータに影響する
  - 放射長、吸収長などの属性を与えることも可能

- 例: 気体
  - 温度、圧力を指定する
    (dE/dxの計算に影響する)

```
G4double density = 27.*mg/cm3;
G4double temperature = 325.*kelvin;
G4double pressure = 50.*atmosphere;

G4Material* CO2 = new G4Material(“CarbonicGas", density,
   ncomponents=2, kStateGas, temperature, pressure);
CO2->AddElement(ele_C,natoms = 1);
CO2->AddElement(ele_O,natoms = 2);
```

kStateGasはG4Material.hhに定義されている列挙型。
kStateUndefined, kStateSolid, kStateLiquid, kStateGasの4種類がある。

注意） 完全な真空は定義できない
universe_mean_density (=1.e-25*g/cm^3) というグローバル変数が予め用意されている

NIST material database in Geant4

- あらかじめ定義されたmaterialが用意されている

- ねらい:

  主要なパラメータの精度を保障する
  - 密度
  - 平均励起ポテンシャル
  - 化学結合
  - 元素構成
  - 同位体構成

- National Institute of Standards and Technology (NIST)のデータベースがGeant4に取り入れられている (http://physics.nist.gov/PhysRefData)

NIST Elements from Isotopes

Z		A	m	error	(%)	A_{eff}
14	Si	22	22.03453	(22)		28.0855(3)
		23	23.02552	(21)		
		24	24.011546	(21)		
		25	25.004107	(11)		
		26	25.992330	(3)		
		27	26.98670476	(17)		
		28	27.9769265327	(20)	92.2297 (7)	
		29	28.97649472	(3)	4.6832 (5)	
		30	29.97377022	(5)	3.0872 (5)	
		31	30.97536327	(7)		
		32	31.9741481	(23)		
		33	32.978001	(17)		
		34	33.978576	(15)		
		35	34.984580	(40)		
		36	35.98669	(11)		
		37	36.99300	(13)		
		38	37.99598	(29)		
		39	39.00230	(43)		
		40	40.00580	(54)		
		41	41.01270	(64)		
		42	42.01610	(75)		

- Natural isotope compositions
- 3000種以上の同位体が定義されている

Geant4におけるNIST materials

```
=====================================
###  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
```

- NIST Elementary Materials
  - H → Cf (Z=1→98)
- NIST compounds
  - E.g. “G4_ADIPOSE_TISSUE_ICRP” (脂肪組織)
- HEP and Nuclear Materials
  - E.g. liquid Ar, $PbWO_4$

- NIST materialとユーザー定義のmaterialを混ぜることも可能

NIST: 使い方

- 主要なユーザーインターフェイス:

```
G4NistManager* manager = G4NistManager::Instance();

G4Element* elm = manager->FindOrBuildElement("symb", G4bool iso);

G4Element* elm = manager->FindOrBuildElement(G4int Z, G4bool iso);

G4Material* mat = manager->FindOrBuildMaterial("name", G4bool iso);

G4Material* mat = manager->ConstructNewMaterial("name",
                        const std::vector<G4int>& Z,
                        const std::vector<G4double>& weight,
                        G4double density, G4bool iso);

G4double isotopeMass = manager->GetMass(G4int Z, G4int N);
```

H,Oなど

G4_WATER
G4_AIRなど

- UI commands

```
/material/nist/printElement    ← 定義されたelementを表示
/material/nist/listMaterials   ← 定義されたmaterialを表示
```

ジオメトリー I

ジオメトリーを定義する

- 抽象基底クラスG4VUserDetectorConstructionを継承しユーザー独自の具象クラスを実装する

- Construct()関数を実装する

  1)すべての必要なmaterialを定義

  2)ジオメトリーを定義する

  1. shape/solidを定義
  2. logical volumeを定義
  3. Volumeを配置
  4. 磁場を与える *(optional)*
  5. sensitive detectors / scorers を定義し、対応するvolumeに設定する。*(optional)*
  6. 可視化(visualization)属性を与える *(optional)*

- main関数でユーザークラスをG4RunManagerに設定する

G4VUserDetectorConstruction

G4VUserDetectorConstruction

```
// $Id: G4VUserDetectorConstruction.hh,v 1.4 2001/07/11 10:08:33 gunter Exp $
// GEANT4 tag $Name: geant4-08-00-patch-01 $
//

#ifndef G4VUserDetectorConstruction_h
#define G4VUserDetectorConstruction_h 1

class G4VPhysicalVolume;

// class description:
//
//  This is the abstract base class of the user's mandatory initialization class
// for detector setup. It has only one pure virtual method Construct() which is
// invoked by G4RunManager when it's Initialize() method is invoked.
//  The Construct() method must return the G4VPhysicalVolume pointer which represents
// the world volume.
//

class G4VUserDetectorConstruction
{
  public:
    G4VUserDetectorConstruction();
    virtual ~G4VUserDetectorConstruction();

  public:
    virtual G4VPhysicalVolume* Construct() = 0;
};

#endin
```

Construct()関数は純粋仮想関数

派生クラスで実装(ジオメトリーを定義)しなければならない。

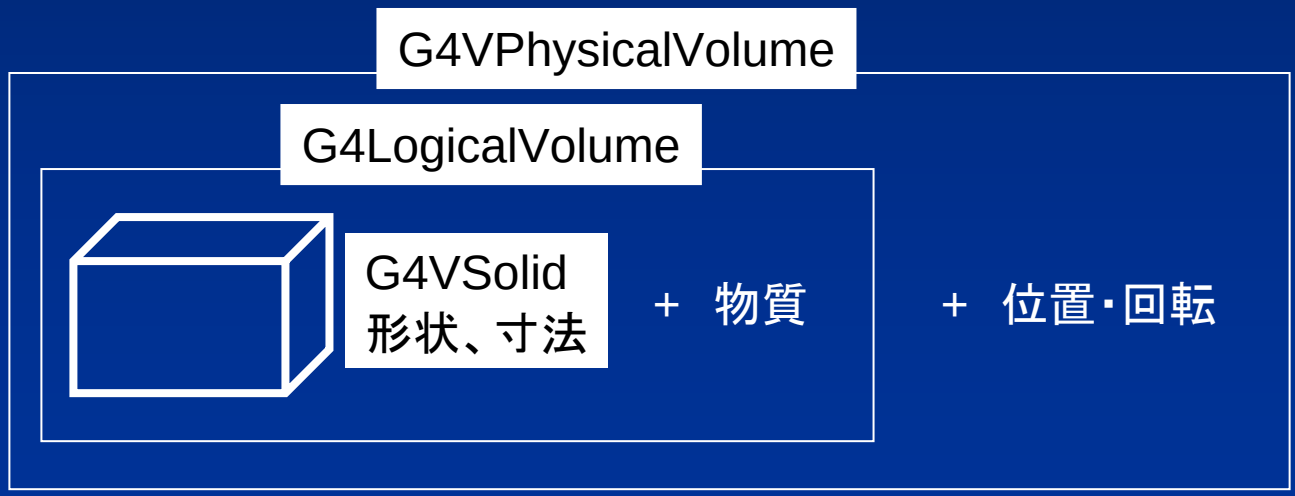
world physical volumeへのポインタを返す。

“My” detector construction

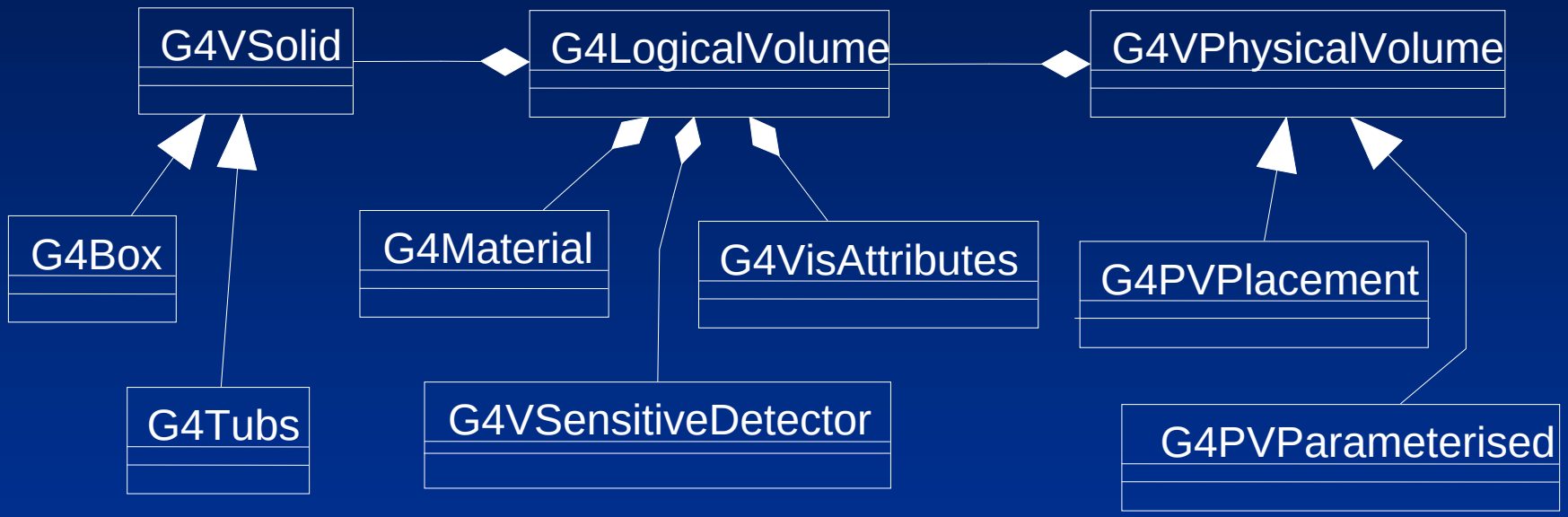
```
#ifndef MyDetctorConstruction_h
#define MyDetctorConstruction_h 1
#include “G4VUserDetectorConstruction.hh”
class MyDetctorConstruction
     : public G4VUserDetectorConstruction
{
 public:
  G4VUserDetectorConstruction();
  virtual ~G4VUserDetectorConstruction();
  virtual G4VPhysicalVolume* Construct();
 public:
  // set/get methods if needed
 private:
  // granular private methods if needed
  // data members if needed
};
#endif
```

検出器のジオメトリー(幾何学的構造)

- 3つの階層
  - G4VSolid -- *形状、寸法*
  - G4LogicalVolume --*物質、感度属性、User Limits*
    daughter physical volumesなど
  - G4VPhysicalVolume – *位置、回転など*



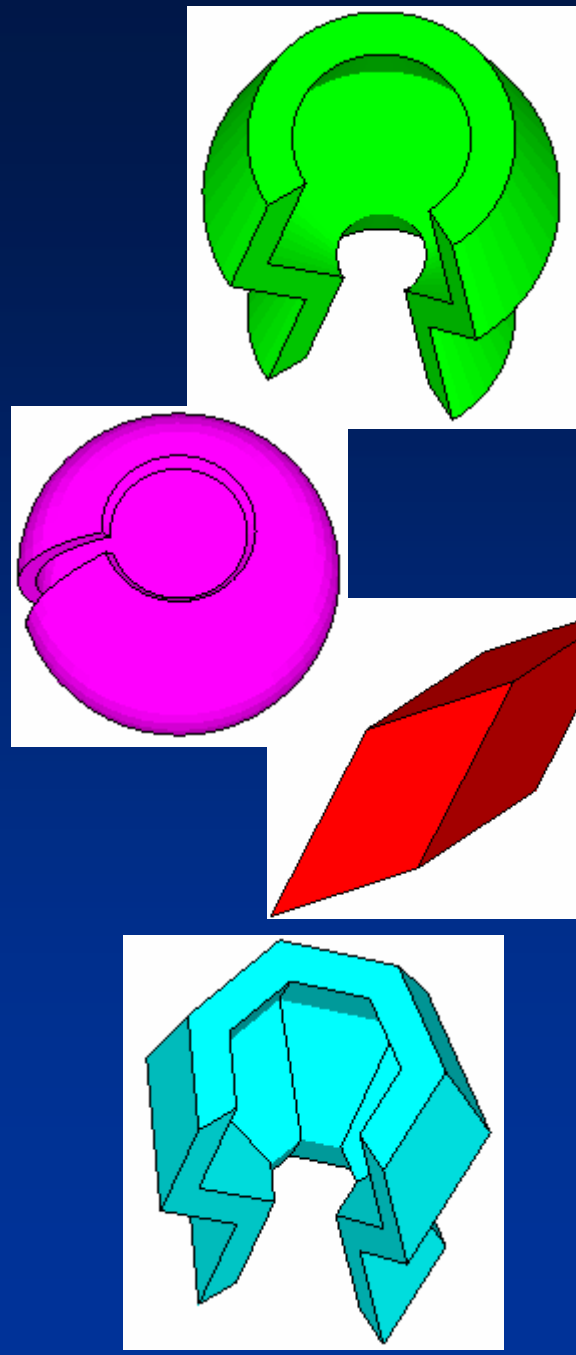
クラスの関係


G4VSolid
G4LogicalVolume
G4VPhysicalVolume
G4Box
G4Material
G4VisAttributes
G4PVPlacement
G4Tubs
G4VSensitiveDetector
G4PVParameterised

Solid and shape

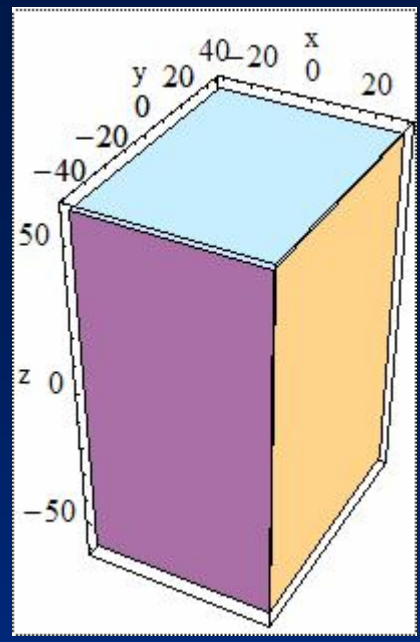
Solids

- Geant4で定義されているSolid:
  - CSG (Constructed Solid Geometry) solids
    - G4Box, G4Tubs, G4Cons, G4Trd, …
  - Specific solids (CSG like)
    - G4Polycone, G4Polyhedra, G4Hype, …
  - BREP (Boundary REPresented) solids
    - G4BREPSolidPolycone, G4BSplineSurface, …
    - Any order surface
  - Boolean solids
    - G4UnionSolid, G4SubtractionSolid, …
  - Tessellated solid and Extruded solid

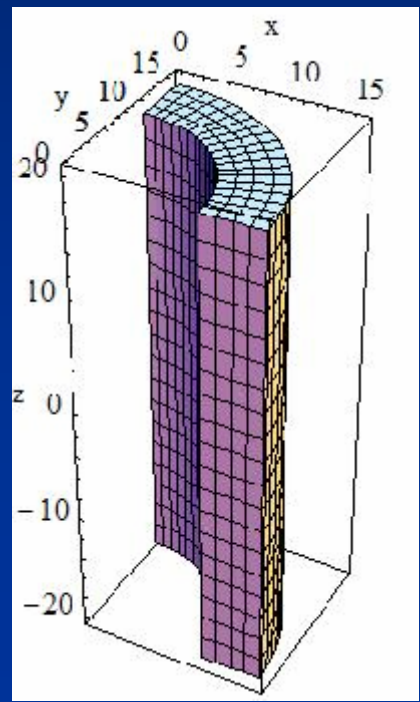


CSG: G4Box, G4Tubs

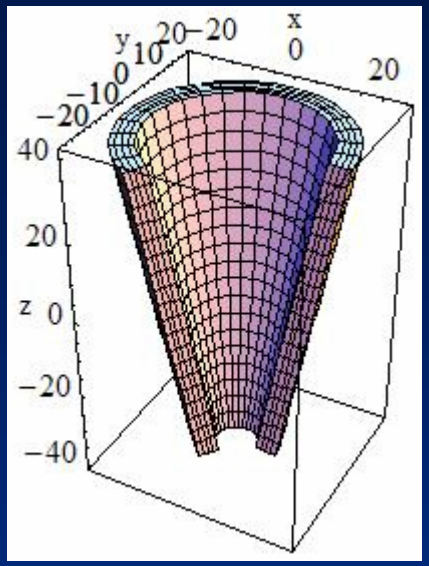
```
G4Box(const G4String &pname,    // name
            G4double half_x,    // X half size
            G4double half_y,    // Y half size
            G4double half_z);   // Z half size
```



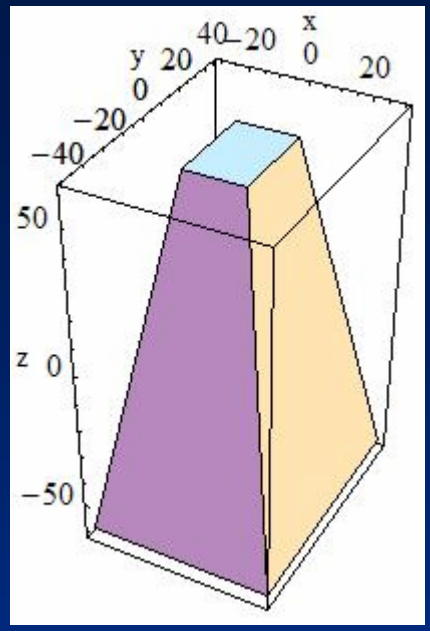

```
G4Tubs(const G4String &pname,   // name
             G4double  pRmin,   // inner radius
             G4double  pRmax,   // outer radius
             G4double  pDz,     // Z half length
             G4double  pSphi,   // starting Phi
             G4double  pDphi); // segment angle
```



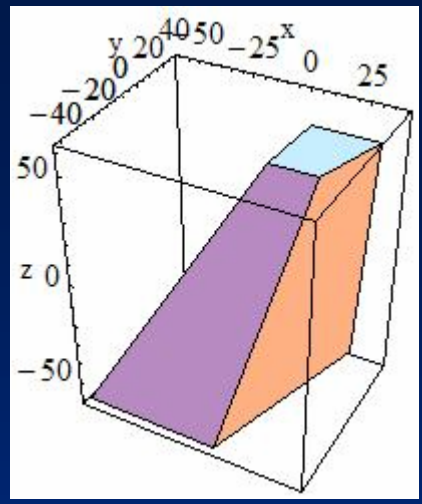
Other CSG solids



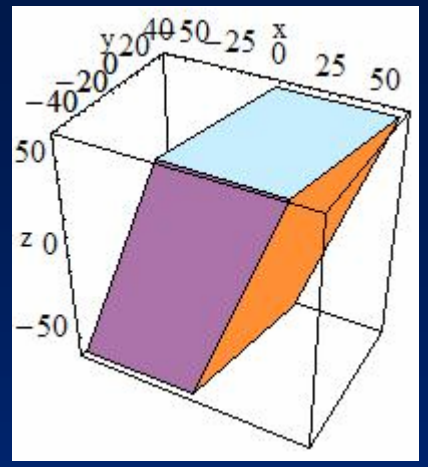
G4Cons



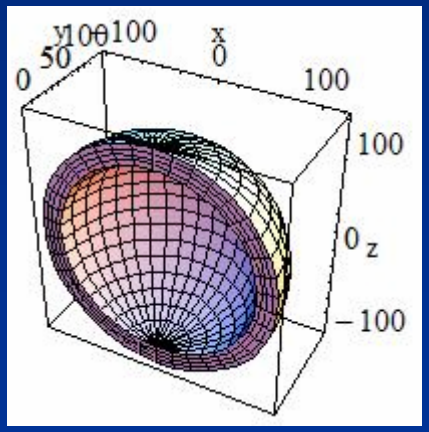
G4Trd



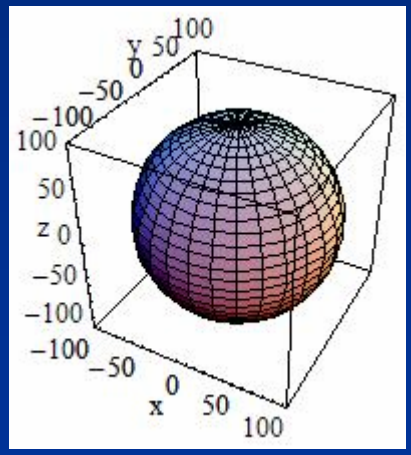
G4Trap



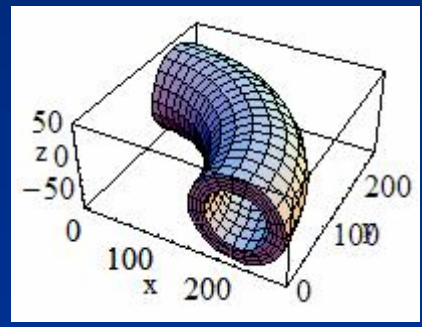
G4Para
(parallelepiped)



G4Sphere



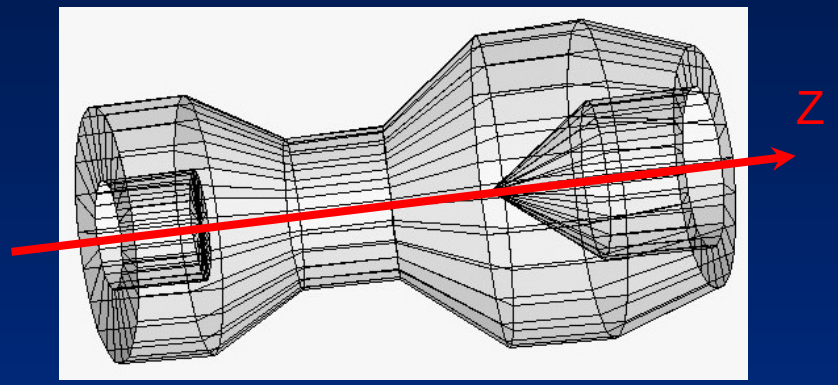
G4Orb
(full solid sphere)



G4Torus

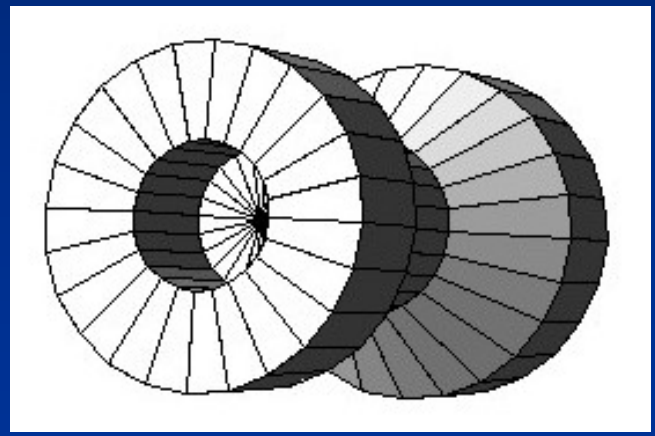
全ての利用可能な形状に関しては
Section 4.1.2 of Geant4 Application Developers Guide を参照

Specific CSG Solids: G4Polycone

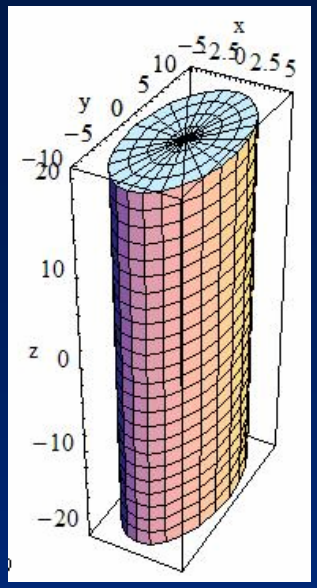



```
G4Polycone(const G4String& pName,
           G4double phiStart,
           G4double phiTotal,
           G4int numRZ,
           const G4double r[],
           const G4double z[]);
```

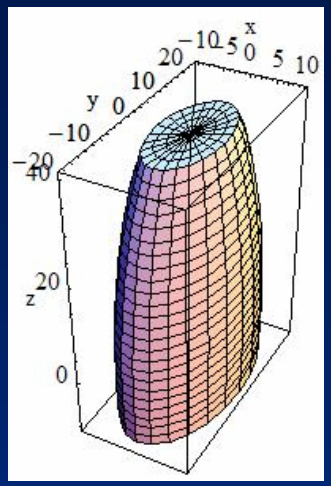
- **numRZ** - numbers of corners in the **r,z** space
- **r, z** - coordinates of corners



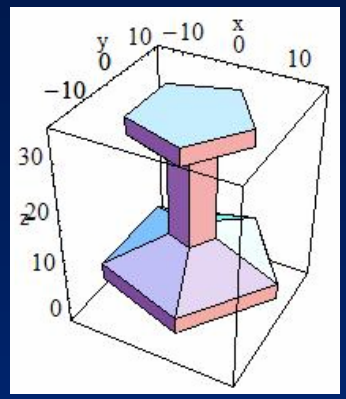
Other Specific CSG solids



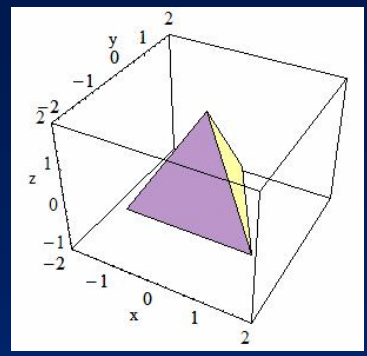
G4EllipticalTube



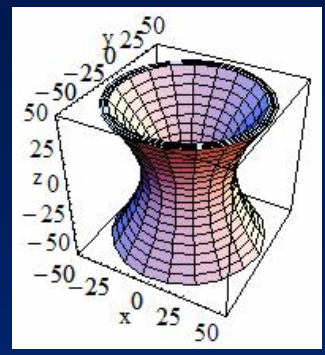
G4Ellipsoid



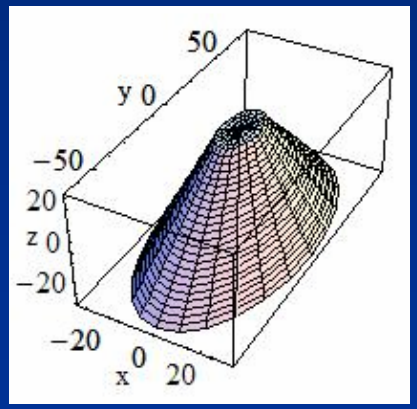
G4Polyhedra



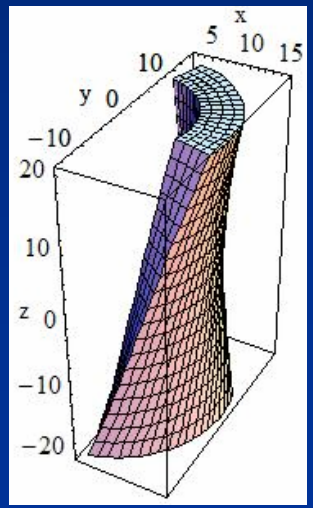
G4Tet
(tetrahedra)



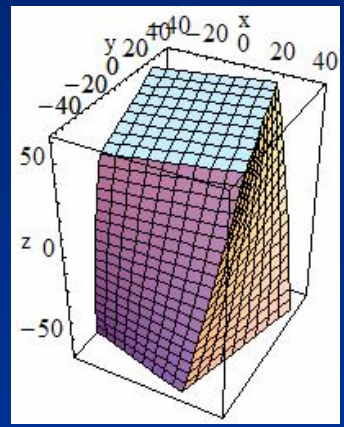
G4Hype



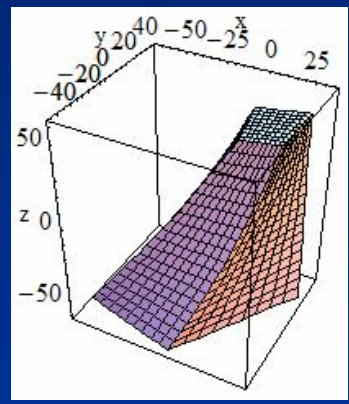
G4EllipticalCone



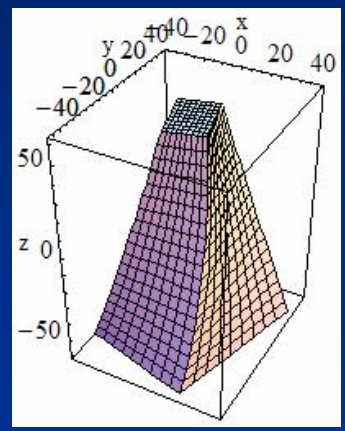
G4TwistedTubs



G4TwistedBox



G4TwistedTrap

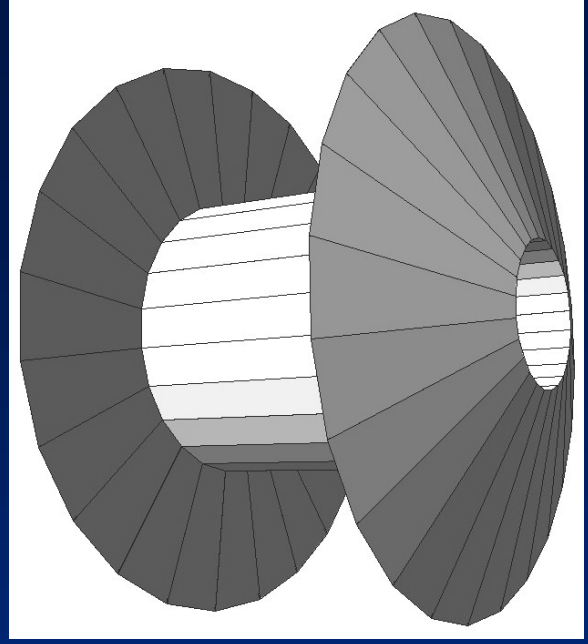


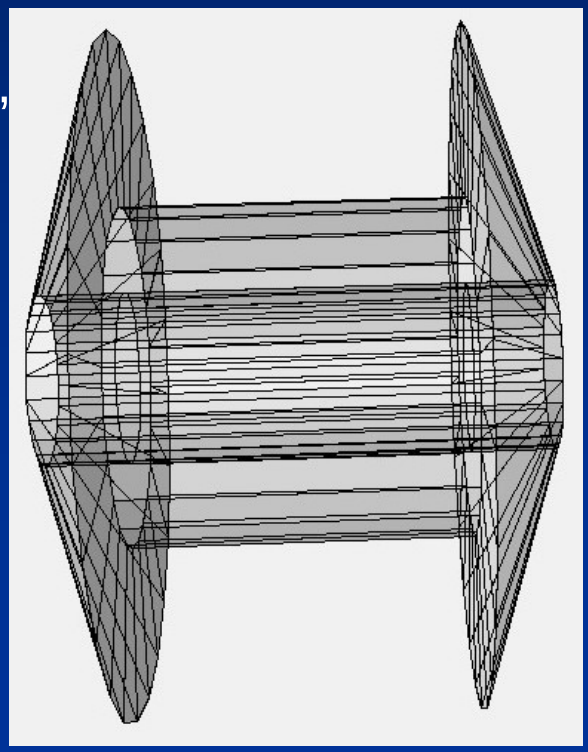
G4TwistedTrd

全ての利用可能な形状に関しては
Section 4.1.2 of Geant4 Application Developers Guide を参照

BREP Solids

- *BREP = Boundary REPresented Solid*
- Solidを特徴づけるすべての境界面を列挙
  - 例、 立方体の6面
  - 平面, 二次あるいは高次の曲面
    - elementary BREPS
  - Splines(スプライン), B-Splines(ベジエ-スプライン), NURBS (Non-Uniform B-Splines)
    - advanced BREPS
- Advanced BREPSはCADで設計したモデルを取り込むことを想定している

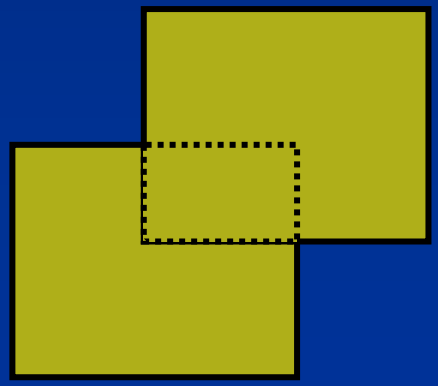




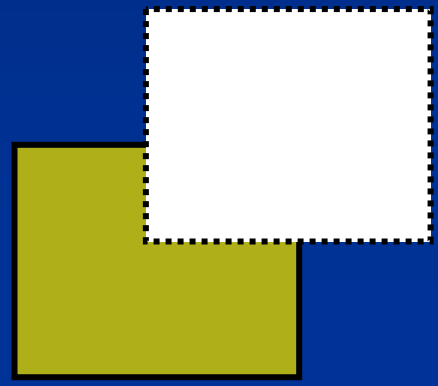
Boolean Solids

- Solid はブール演算を使って結合できる:
  - **G4UnionSolid, G4SubtractionSolid, G4IntersectionSolid**
  - 必要なもの: 2つのsolid, 1つのブール演算, および 2番目のsolid の変換 (optional)
  - 2番目のsolid は1番目のsolid の座標系に対して配置される
  - ブール演算の結果がsolidとなる。このようにして3番目のsolidも最初の演算によって作られたsolidに結合できる。
- 結合されるSolidはCSGでも他のBoolean solidでもよい。
- 注意: Boolean solid中のCSG solidの数に比例してトラッキングのための計算時間が増える

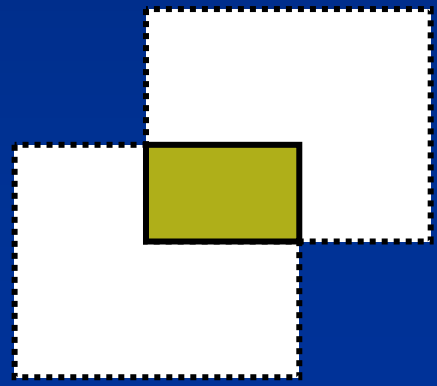
G4UnionSolid



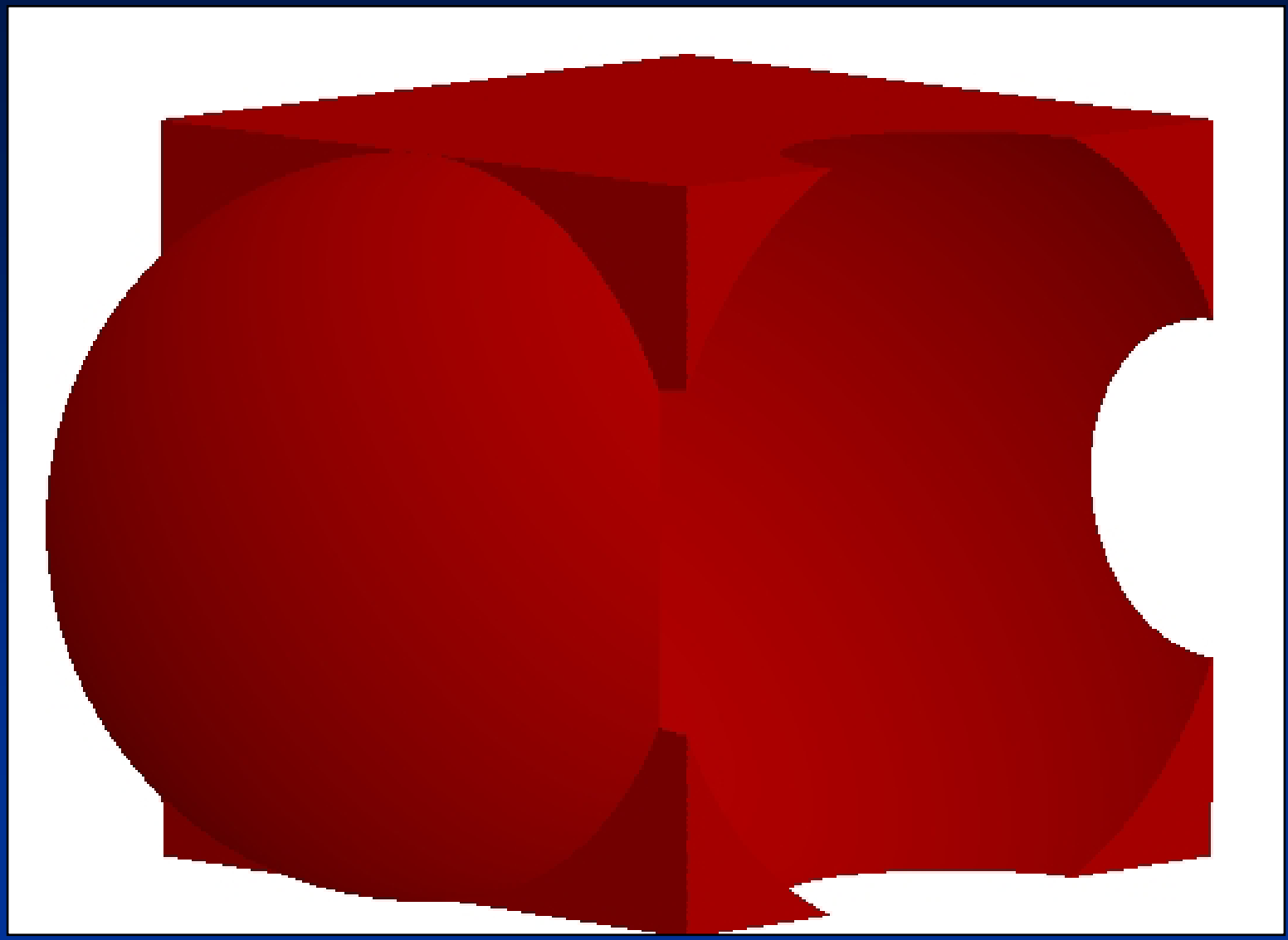
G4SubtractionSolid



G4IntersectionSolid



Boolean solid



OpenGL, VRMLなどでは見えない
RayTracerを使って描く

Boolean Solids – 例

```
G4VSolid* box = new G4Box(“Box",50*cm,60*cm,40*cm);
G4VSolid* cylinder
 = new G4Tubs(“Cylinder”,0.,50.*cm,50.*cm,0.,2*M_PI*rad);
G4VSolid* union
 = new G4UnionSolid("Box+Cylinder", box, cylinder);
G4VSolid* subtract
 = new G4SubtractionSolid("Box-Cylinder", box, cylinder,
       0, G4ThreeVector(30.*cm,0.,0.));
G4RotationMatrix* rm = new G4RotationMatrix();
rm->RotateX(30.*deg);
G4VSolid* intersect
   = new G4IntersectionSolid("Box&&Cylinder",
       box, cylinder, rm, G4ThreeVector(0.,0.,0.));
```

G4LogicalVolume

G4LogicalVolume

```
G4LogicalVolume(G4VSolid *pSolid,
                G4Material *pMaterial,
                const G4String &name,
                G4FieldManager *pFieldMgr=0,
                G4VSensitiveDetector *pSDetector=0,
                G4UserLimits *pULimits=0);
```

- 位置と回転以外のvolumeに関するすべての情報を含む
  インストール前の測定器のようなもの
  - 形状と寸法 (G4VSolid)
  - 物質、感度属性、可視化(visualization)属性
  - daughter volumeの位置
  - 磁場, User limits,
  - Region
- Solidへのポインタはnullではいけない。
- トラッキングするためのジオメトリーではmaterialへのポインタはnullではいけない。
- 基底クラスとしては機能しない

体積と質量の計算

- Solidの体積を計算できる:

```
G4double GetCubicVolume();
```

  - ほとんどのCSG solidについては正確な体積が計算される、それ以外のsolidにはMonte Carlo 積分による評価がされる。

- ジオメトリー全体(またはその一部)の質量はlogical volumeから計算できる:

```
G4double GetMass(G4bool forced=false)
```

G4VPhysicalVolume

LogicalVolumeを配置する

最も単純なやり方
G4PVPlacementを使う　：physical volumeのコンストラクタの一つ
　　娘LogicalVolumeを親Volumeの座標系の中の指定した位置に置く。

```
G4PVPlacement( G4RotationMatrix* pRot,
               const G4ThreeVector& tlate,
                     G4LogicalVolume* pCurrentLogical,
               const G4String& pName,
                     G4LogicalVolume* pMotherLogical,
                     G4bool pMany, G4int pCopyNo,
                     G4bool pSurfChk=false )
```

pRotはRotationMatrix
このように与えた場合、置かれるLogicalVolumeではなく座標系の方が回転する。
普通に物を回転させるセンズでMatrixを作ると、逆方向に回転するので注意。
引数として与えるRotationMatrixは予め定義しておく必要がある。
tlateは平行移動を与えるベクトル

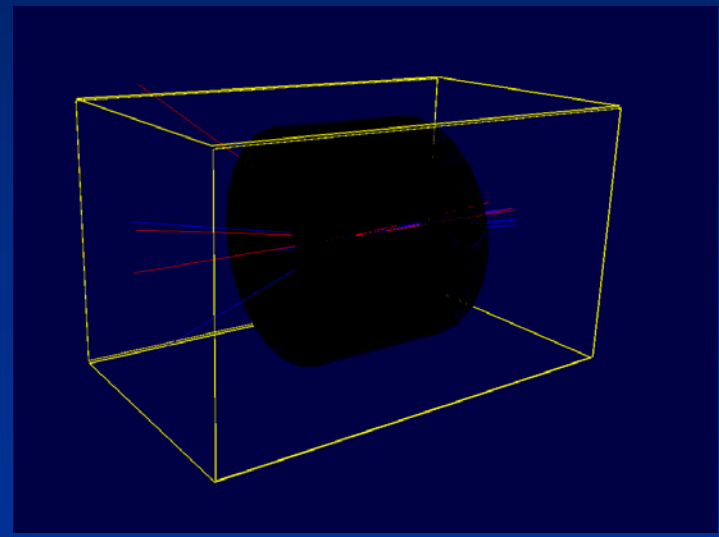
pCopyNoは、同じLogicalVolumeを使っていくつもPhysicalVolumeを
置きたい場合につけるコピーナンバー。
pManyは今は使っていない引数なので常にFalseでよい。
G4PVPlacementで作る1つのインスタンスは一つの検出器（部品）しか代表しないので、
置きたい数だけnewで作成する。

普通のセンスで回転マトリクスを定義したい場合にはG4Transform3Dを使う。
コンストラクタの第１引数に物が回転する方向の回転マトリクス、
第２引数に平行移動のベクトルを与えて作る。

- Motherとdaughterの原点、座標軸を合わせる
- Daughterを回す
- Daughterを置く

例) /examples/novice/N04/src/ExN04DetectorConstruction.cc

```
//------------------------------ muon counters
// As an example of CSG volumes with rotation
  G4VSolid * muoncounter_box
    = new G4Box("muoncounter_box",muBox_width,muBox_thick,
                                  muBox_length);
  G4LogicalVolume * muoncounter_log
    = new G4LogicalVolume(muoncounter_box,Scinti,"mucounter_L",0,0,0);
  G4VPhysicalVolume * muoncounter_phys;
  for(int i=0; i<nomucounter ; i++)
  {
    G4double phi, x, y, z;
    phi = 360.*deg/nomucounter*i;
    x = muBox_radius*std::sin(phi);
    y = muBox_radius*std::cos(phi);
    z = 0.*cm;
    G4RotationMatrix rm;
    rm.rotateZ(phi);                          //Z軸まわりの回転
    muoncounter_phys
      = new G4PVPlacement(G4Transform3D(rm,G4ThreeVector(x,y,z)),
                          muoncounter_log, "muoncounter_P",
                          experimentalHall_log,false,i);
  }
```



World volumeを定義

- Mother volume = 0とする
- 回転なし
- 原点に置く

```
G4VPhysicalVolume* experimentalHall_phys
 = new G4PVPlacement( 0,
                      G4ThreeVector(0., 0., 0.),
                      experimentalHall_log, “expHall”,0,
                      false, 0);
      …
      …

return experimentalHall_phys;  //world volumeへのポインタを返す
```

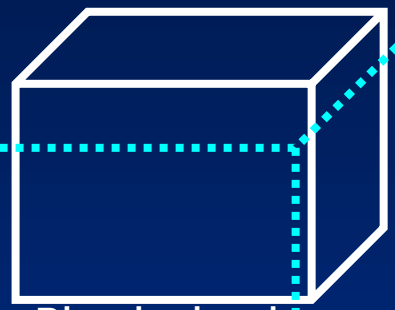
おさらい

検出器のジオメトリーを定義する

- 基本的な方針

```
G4VSolid* pBoxSolid =
  new G4Box(“aBoxSolid”,
        1.*m, 2.*m, 3.*m);
G4LogicalVolume* pBoxLog =
  new G4LogicalVolume( pBoxSolid,
    pBoxMaterial, “aBoxLog”, 0, 0, 0);
G4VPhysicalVolume* aBoxPhys =
  new G4PVPlacement( pRotation,
    G4ThreeVector(posX, posY, posZ),
    pBoxLog, “aBoxPhys”, pMotherLog,
    0, copyNo);
```

Logical volume :
+ material, sensitivity, etc.
solid : shape and size

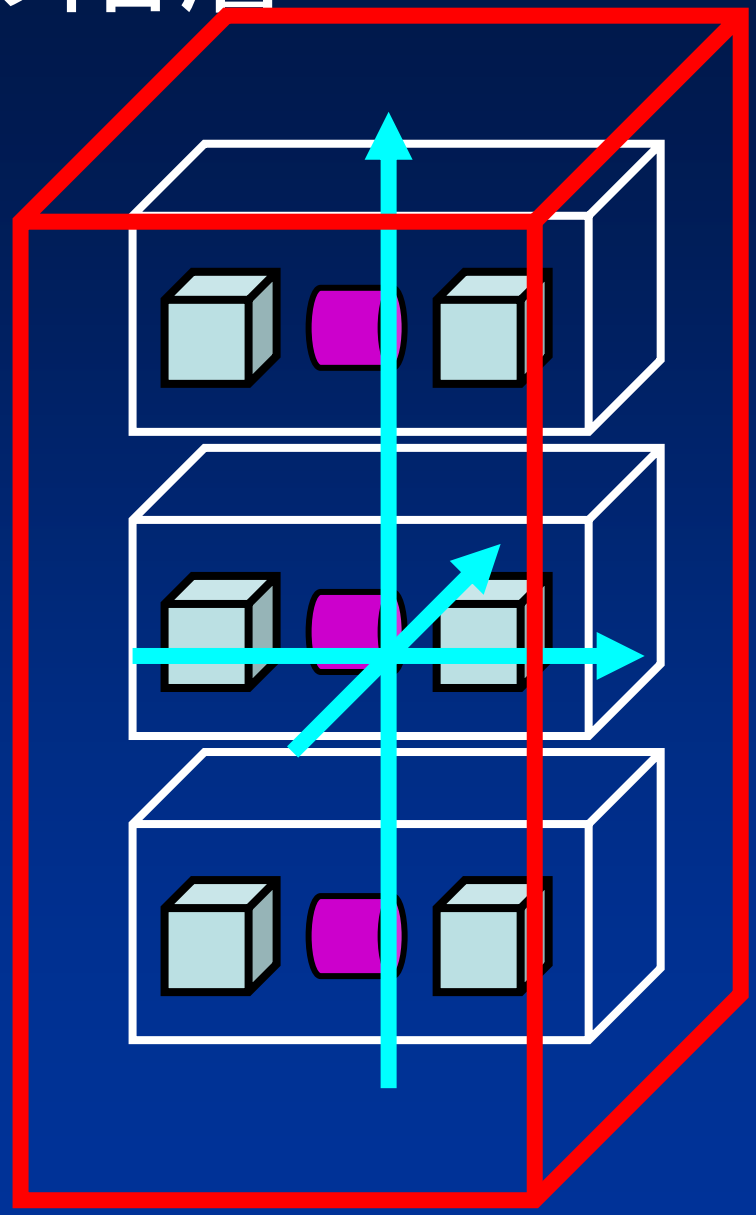


Physical volume :
+ rotation and position

- ある volumeはそのmother volumeの内部に置かれる。
  daughter volumeの位置と回転はそのmother volume
  　でのローカル座標系で記述する。
  mother volumeの中心がそのローカル座標系の原点となる。
  - Daughter volumeはmother volumeからはみ出してはいけない

ジオメトリーの階層

- 一つのlogical volumeを一回以上配置することができる。一つあるいはそれ以上のvolumeをあるmother volumeに置くことができる
- mother-daughterの関係は G4LogicalVolumeが持つ情報であることに注意
  - もしmother volumeを1回以上配置すると、すべてのdaughterは定義により、すべてのmother physical volumeに現れることになる
- world volumeは他のすべてのvolumeを完全に含むユニークなphysical volumeでなければならない。
  - world volumeはグローバル座標系を定義する。world volumeの中心がグローバル座標系の原点となる
  - トラックの位置はグローバル座標系で与えられる



Geometry collision detection

- Volume同士のcollisionは許されない
  →まともなtrackingがされない

```
G4PVPlacement( G4RotationMatrix* pRot,
               const G4ThreeVector& tlate,
                     G4LogicalVolume* pCurrentLogical,
               const G4String& pName,
                     G4LogicalVolume* pMotherLogical,
                     G4bool pMany, G4int pCopyNo,
                     G4bool pSurfChk=true)
```

衝突チェックがなされる

問題なければ
Checking overlaps for volume muoncounter_P ... OK!

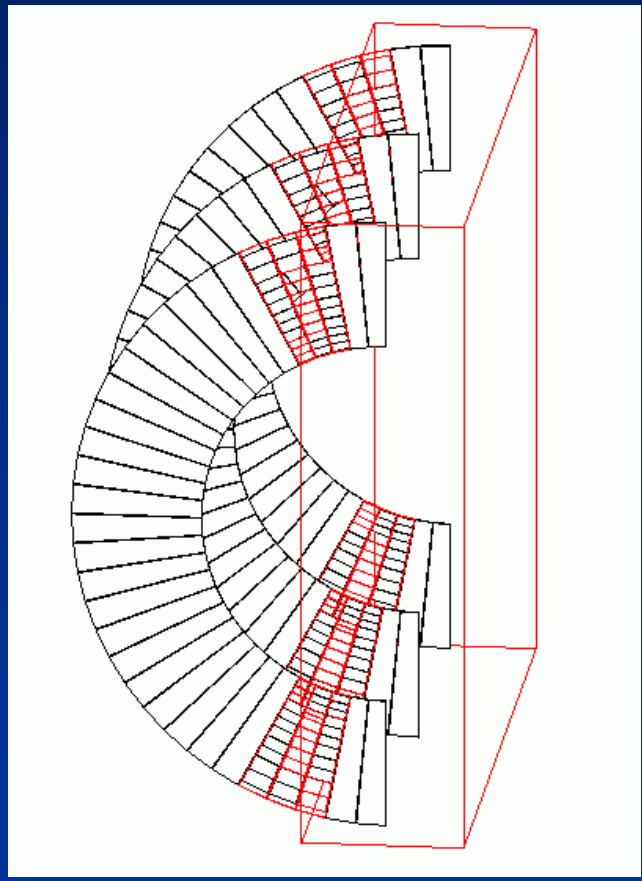
衝突が検出されると警告が出る

```
Checking overlaps for volume muoncounter_P ...
WARNING - G4PVPlacement::CheckOverlaps()
        Overlap is detected for volume muoncounter_P
        with caloM_P volume's
        local point (-401.016,-1213.5,-10), overlapping by at least: 77.804 cm

*** G4Exception : InvalidSetup
      issued by : G4PVPlacement::CheckOverlaps()
Overlap with volume already placed !
*** This is just a warning message.
```

時間がかかるので、デバッグ時だけに使うことをおすすめします。

- DAVIDを使ってVolumeの衝突を検出することができる
  - 色が着く(デフォルトは赤)
  - http://www-geant4.kek.jp/vis/



まとめ

- 物質
  - 同位体、元素をもとに初めから書く
  - あるいは、あらかじめ定義された物質を使う
    - NIST material data base
- 検出器などの構造体　(ジオメトリー)
  - 3つの階層構造
    - Solid/shape
    - Logical volume
    - Physical volume
      - G4PVPlacementクラスを使う最も単純な方法を解説した

Why / Where materials ?

- Geant4で追跡している粒子が起こす物理過程は、そこがどのような物質であるかに依存する

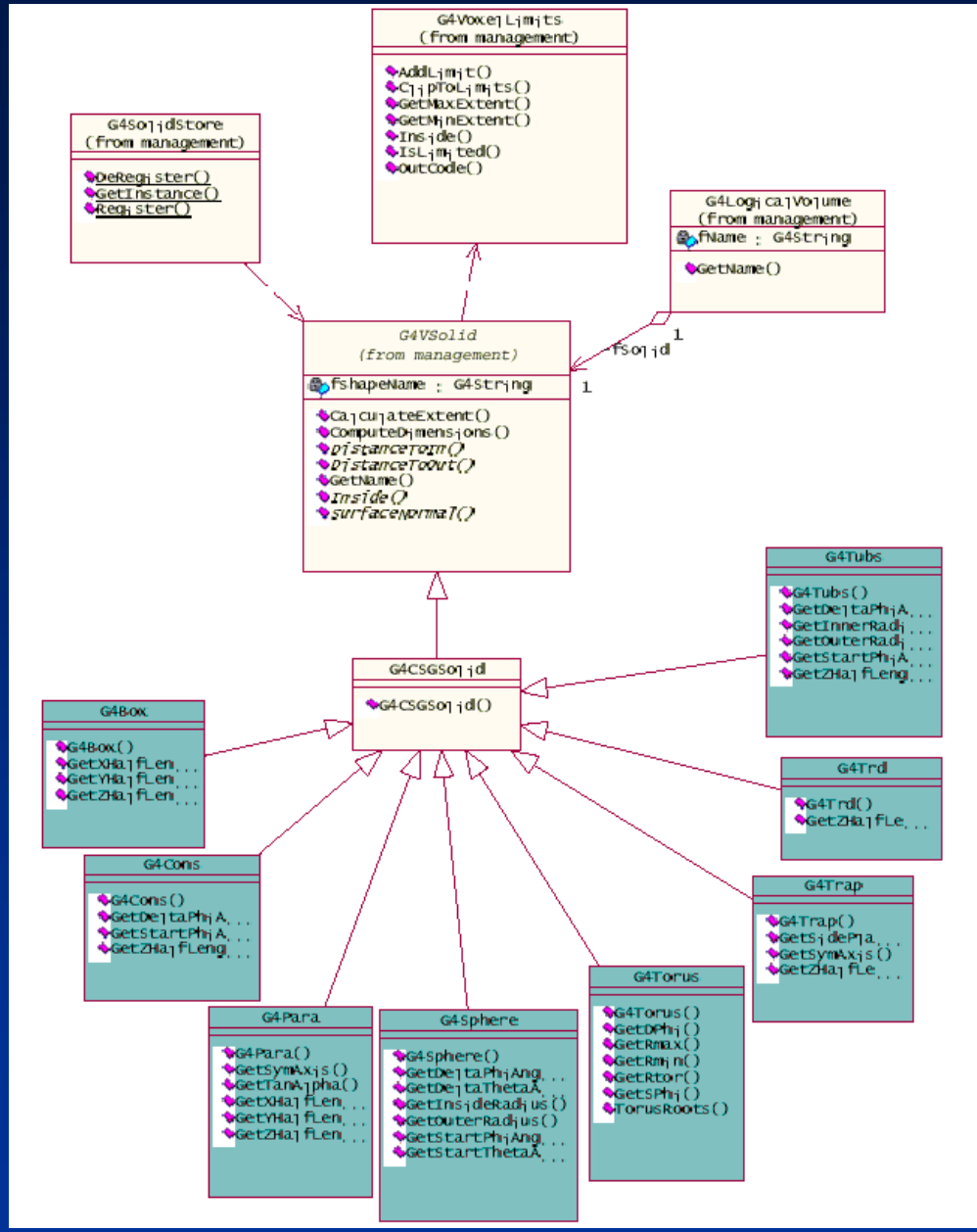
  G4eIonisation → G4VEnergyLossProcess::BuildDEDXTable(…)

- ジオメトリー(幾何学的構造)を定義するときに、ユーザーはlogical volumeに物質を関係つけることが要求される

```
G4LogicalVolume::G4LogicalVolume ( G4VSolid* solid,
G4Material* material, const G4String name, … )
```

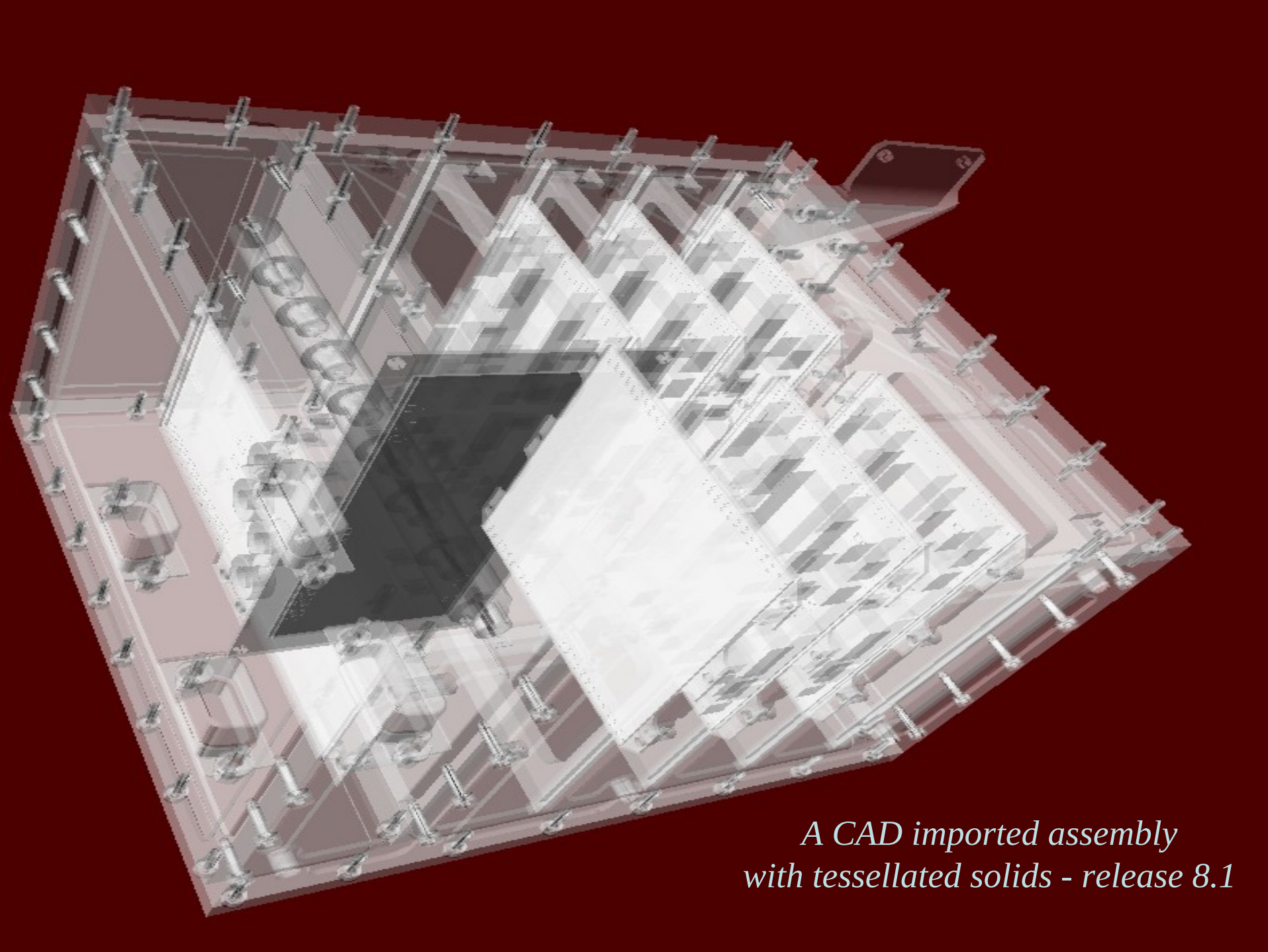
G4VSolid

- 抽象クラス。Geant4でのすべてのsolidはそこから継承される。
- It defines but does not implement all functions required to:
  - compute distances between the shape and a given point
  - check whether a point is inside the shape
  - compute the extent of the shape
  - compute the surface normal to the shape at a given point
- ユーザーは自分自身でsolid classを作ることができる



Tessellated (モザイク式の) solids

- **`G4TessellatedSolid`**
  - いくつかの面で定義された一般的なsolid (**`G4VFacet`**)
    - 面は三角形 (**`G4TriangularFacet`**) あるいは四角形 (**`G4QuadrangularFacet`**)
  - CADから取り入れられた複雑な幾何学的形状を変換するために重要な構造物
  - 明示的に定義することもできる:
    - By providing the vertices of the facets in *anti-clock wise* order, in *absolute* or *relative* reference frame
  - GDML binding



A CAD imported assembly
with tessellated solids - release 8.1

G4ExtrudedSolid (押し出された)

- G4ExtrudedSolid is a specific case of tessellated solid.
- G4ExtrudedSolid is a solid which represents the extrusion of an arbitrary polygon with fixed outline in the defined Z sections.
- The z-sides of the solid are the scaled versions of the same polygon.
- The solid is implemented as a specification of G4TessellatedSolid.