

Detektör Tasarımı ve Benzetimi - II

Doç. Dr. Sinan KUDAY

*Ankara Üniversitesi Fizik Bölümü
skuday@ankara.edu.tr*

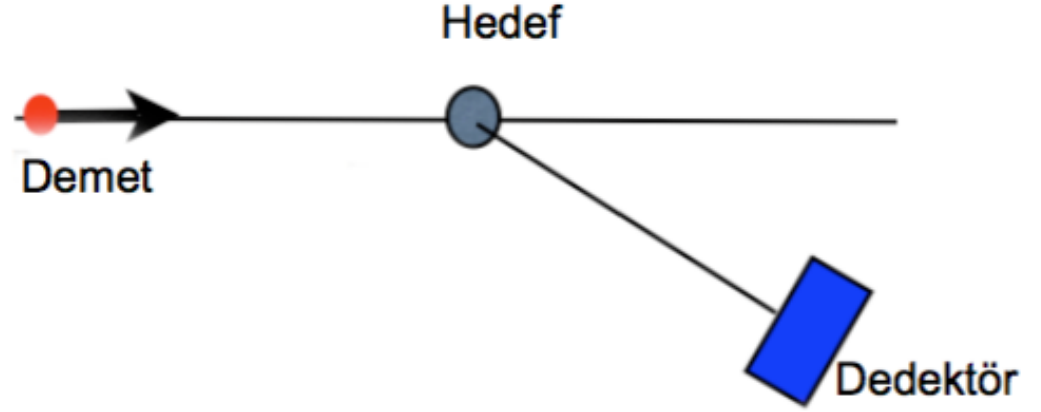
1. Geant4 Kod Yapısı
2. Geant4 Geometrileri Oluřturma
3. Materyal Nesneleri ve Yönetimi
4. Geant4 Örnekleri (Projelere Başlangıç)
5. Uygulamalar

Kısım 1:

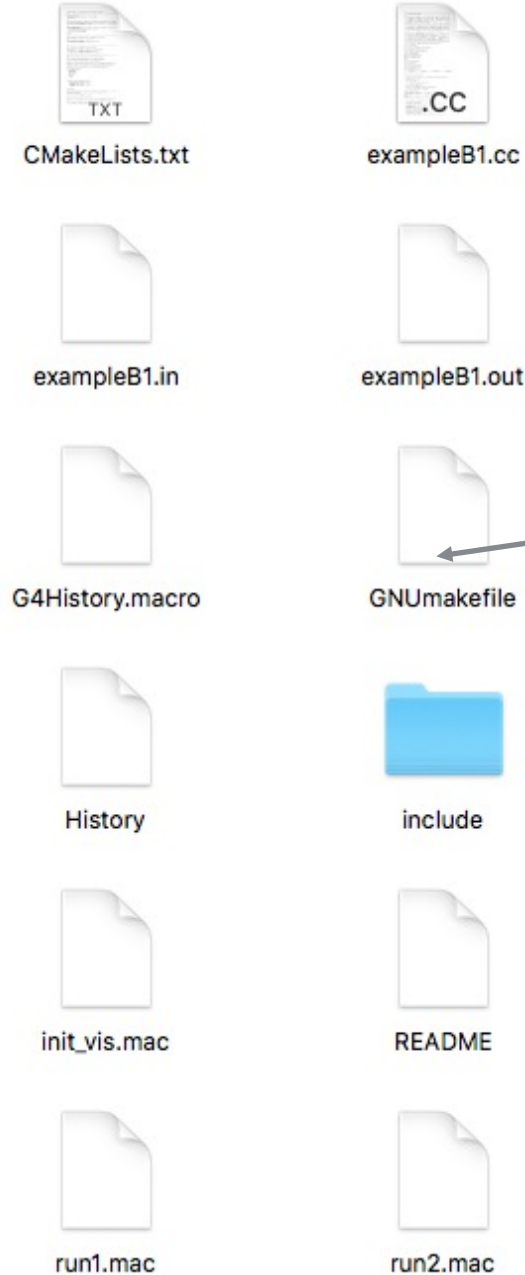
Geant4 Kod Yapısı

Basit bir nkleer Fizik deneyinin tanımlanması için gerekli nesnelere:

- Demet
- Hedef
- Dedektr
- Etkileşim Alanları
- Fizik Etkileşimleri ve Algoritmaları
- Analiz Yntemleri



ExampleB1 Dosyaları



```
# $Id: GNUmakefile 68058 2013-03-13 14:47:43Z gcosmo $
# -----
# GNUmakefile for examples module.  Gabriele Cosmo, 06/04/98.
# -----

name := exampleB1
G4TARGET := $(name)
G4EXLIB := true

ifndef G4INSTALL
  G4INSTALL = ../../..
endif

.PHONY: all
all: lib bin

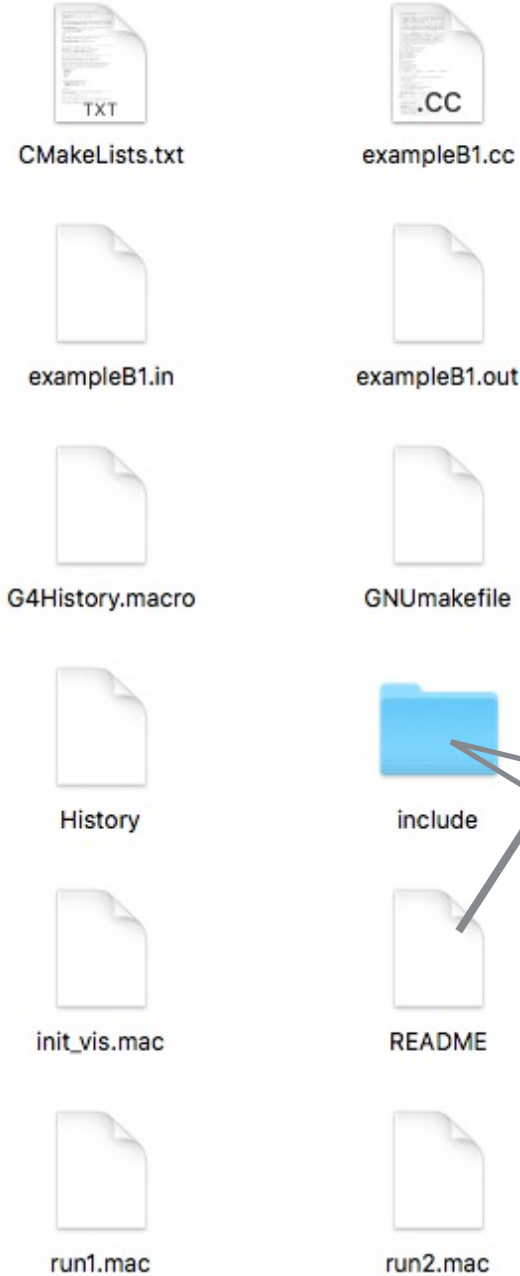
include $(G4INSTALL)/config/binmake.gmk

visclean:
  rm -f g4*.prim g4*.eps g4*.wrl
  rm -f .DAWN_*
```

Derlenmesi (cmake yöntemi):

Derlenmesi (make yöntemi):

ExampleB1 Dosyaları



C- HOW TO RUN

- Execute exampleB1 in the 'interactive mode' with visualization:

```
% ./exampleB1
```

and type in the commands from run1.mac line by line:

```
Idle> /control/verbose 2
```

```
Idle> /tracking/verbose 1
```

```
Idle> /run/beamOn 10
```

```
Idle> ...
```

```
Idle> exit
```

or

```
Idle> /control/execute run1.mac
```

```
....
```

```
Idle> exit
```

- Execute exampleB1 in the 'batch' mode from macro files (without visualization)

```
% ./exampleB1 run2.mac
```

```
% ./exampleB1 exampleB1.in > exampleB1.out
```

include:

h B1ActionInitialization.hh

h B1DetectorConstruction.hh

h B1EventAction.hh

h B1PrimaryGeneratorAction.hh

h B1RunAction.hh

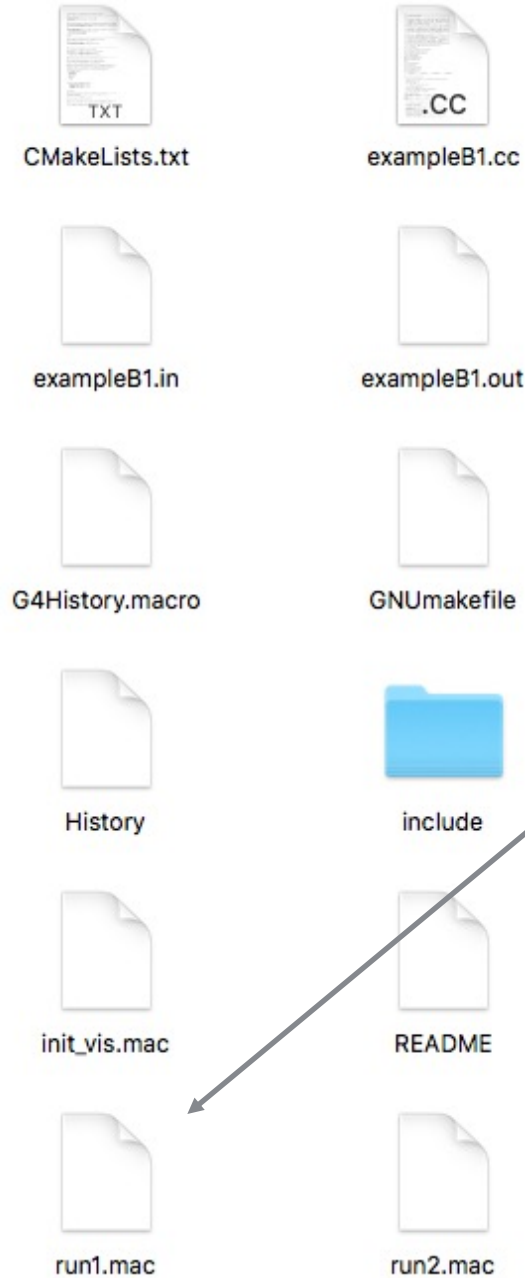
h B1SteppingAction.hh

src:

B1ActionInitialization.cc

B1DetectorConstruction.cc

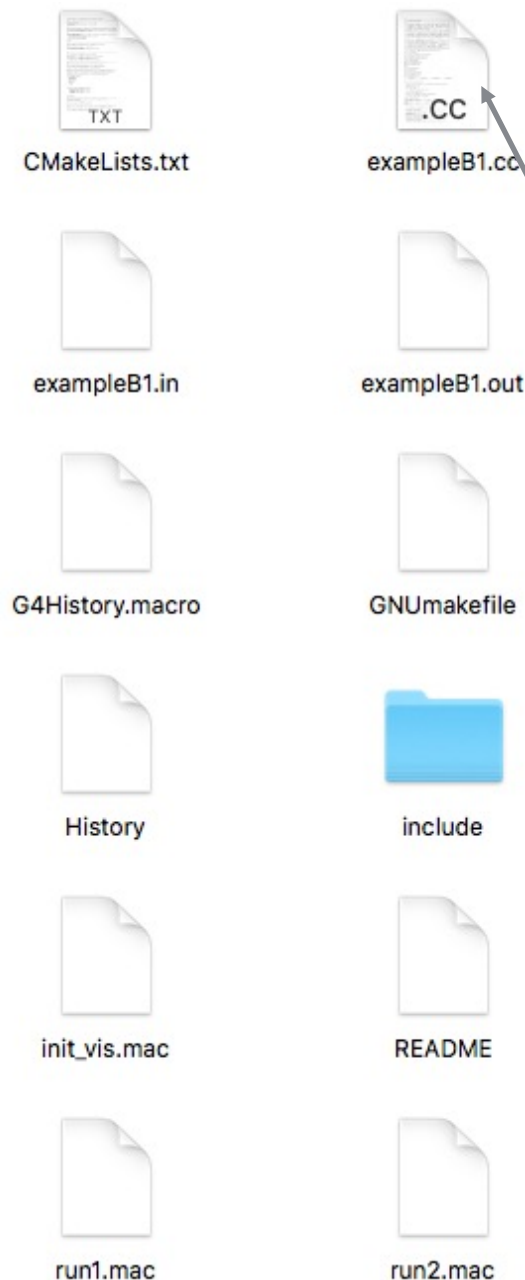
ExampleB1 Dosyaları



A screenshot of a text editor window titled 'run1.mac'. The window shows the following macro file content:

```
1 # Macro file for example B1
2 #
3 # Can be run in batch, without graphic
4 # or interactively: Idle> /control/execute run1.mac
5 #
6 # Change the default number of workers (in multi-threading mode)
7 #/run/numberOfWorkers 4
8 #
9 # Initialize kernel
10 /run/initialize
11 #
12 /control/verbose 2
13 /run/verbose 2
14 /event/verbose 0
15 /tracking/verbose 1
16 #
17 # gamma 6 MeV to the direction (0.,0.,1.)
18 #
19 /gun/particle gamma
20 /gun/energy 6 MeV
21 #
22 /run/beamOn 5
23 #
24 # proton 210 MeV to the direction (0.,0.,1.)
25 #
26 /gun/particle proton
27 /gun/energy 210 MeV
28 /tracking/verbose 2
29 #
30 /run/beamOn 1
```

ExampleB1 Dosyaları



```
// Choose the Random engine
G4Random::setTheEngine(new CLHEP::RanecuEngine);

// Construct the default run manager
//
#ifdef G4MULTITHREADED
  G4MTRunManager* runManager = new G4MTRunManager;
#else
  G4RunManager* runManager = new G4RunManager;
#endif

// Set mandatory initialization classes
// Detector construction
runManager->SetUserInitialization(new B1DetectorConstruction());
// Physics list
G4VModularPhysicsList* physicsList = new QBBC;
physicsList->SetVerboseLevel(1);
runManager->SetUserInitialization(physicsList);
// User action initialization
runManager->SetUserInitialization(new B1ActionInitialization());
// Initialize visualization
G4VisManager* visManager = new G4VisExecutive;
// G4VisExecutive can take a verbosity argument - see /vis/verbose guidance.
// G4VisManager* visManager = new G4VisExecutive("Quiet");
visManager->Initialize();
// Get the pointer to the User Interface manager
G4UImanager* UImanager = G4UImanager::GetUIpointer();
// Process macro or start UI session
//
if ( ! ui ) {
  // batch mode
  G4String command = "/control/execute ";
  G4String fileName = argv[1];
  UImanager->ApplyCommand(command+fileName);
}
else {
  // interactive mode
  UImanager->ApplyCommand("/control/execute init_vis.mac");
  ui->SessionStart();
  delete ui;
}

// Job termination
// Free the store: user actions, physics_list and detector_description are
// owned and deleted by the run manager, so they should not be deleted
```


- Her örnek için README açıklamaları okunup anlaşılmalıdır ,
- **Derleme işlemi için GNUMakefile veya [CMakeLists.txt](#) okunmalıdır,**
- Derlemenin ardından çalıştırılacak **macro dosyasının** anlaşılması gerekir
- Macro dosyasında ilgili modifikasyonlar yapıldıktan sonra çalıştırılmalıdır.

G4 kodlaması, C++ dilinde nesne yönelimli sınıf yapısı (object oriented) kullanılarak oluşturulmuştur !!

Object Oriented Class Methodology bilinmelidir !

Kısım II: Geometri Oluřturma

Geometri Oluřturma

- Geometriler, oluřturacađımız SANAL DENEYlerin bařlangıcıdır.
- Geometri oluřturma; deneyde kullanılacak **materyallerin ve yerlerinin** tanımlanması iřlemidir.
- Benzetimin **her adımında izlenen yol üzerindeki** materyaller (parçacık) kontrol edilir / ötelenebilir / etkileřime girebilir.
- Basit geometriler (kutu, silindir, küre) iřleme süresini kısaltır.
- Ayrıntılı benzetimlerde karmařık/geliřmiř geometriler kullanılabilir.
- İlk geometri tanımları DETEKTÖR inřaasında karřımıza çıkar.

Not: Geliřmiř Geometrilere sonraki sunumda bahsedilecektir.

Geometri Oluşturma

- Tanımlara geometrik şeklin boyutlarını belirleyerek başlayın:

Box 5m x 2m x 10m

Sphere r=10m

Cylinder r = 5m, h = 10 m ...

- Hangi özelliklere sahip olmalı?

Material

Any B- or E-fields?

Is it sensitive, i.e. used for read-out?

- Hangi Konumda ?

Single placement?

Or repeatedly as described by some function?

Solids



Logical Volumes



Physical Volumes

Geometri Oluřturma

- Solid (katı) geometrilerden oluřan tanımlamalardır

Kutu, küre, silindir tanımlamalarından karmařık geometriler de elde edilebilir !!

- Hazır fonksiyonlardır: G4Box, G4Cone, ...vs.
- Boyut argümanlarına ihtiyaç duyarlar

Boyutlar birimleri (cm, mm ?) ile birlikte verilmelidir !!

Solids



Logical Volumes



Physical Volumes

Geometri Oluřturma

Ařağıdaki sıra ile tanımlanırlar:

- Katı geometriye bir materyal eklenir
- Katı geometriye E veya B alanları eklenir
- Katı geometri hassas (sensitive) yapılır
- Kullanıcı limitleri belirlenir
- Fiziksel kız hacimler belirlenir.
(physical daughter volumes)

Solids



Logical Volumes



Physical Volumes

Geometri Oluşturma

Aşağıdaki sıra ile tanımlanırlar:

- Hacmin konumu ve rotasyonu belirtilir.
- Aynı hacmin replikası veya tekrarı belirtilir.

Solids



Logical Volumes



Physical Volumes

Örnek:

```
G4Box* boxSolid = new G4Box("aBox", 1.0*m, 1.0*m, 1.0*m);
```

```
G4LogicalVolume* boxLogic = new G4LogicalVolume(BoxSolid, BoxMaterial,  
"Box1Logic")
```

```
G4ThreeVector pos(1*m, 1*cm, 1*mm);  
G4RotationMatrix* rot = 0;  
G4VPhysicalVolume* boxPhys = new G4PVPlacement(rot, 0, BoxLogic, "Box1Placed",  
World, pMany = 0, copyNo = 0, surfChk = true);
```

- Tüm katı şekiller G4VSolid sınıfıyla türetilir / oluşturulur.
- Tanımlanması bağlantılı bazı fonksiyonları kullanır hale getirir

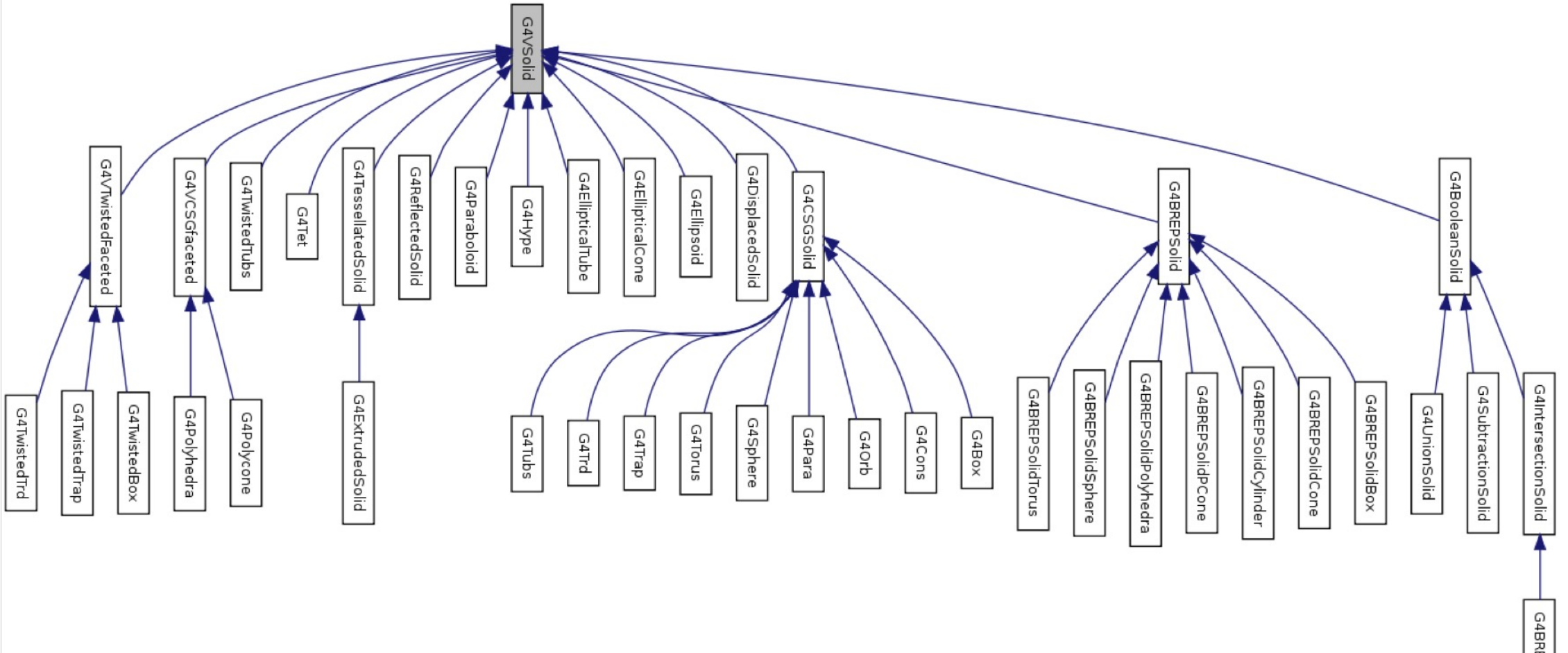
bool Inside(G4ThreeVector)

G4double DistanceToIn(G4ThreeVector)

G4double DistanceToOut(G4ThreeVector)

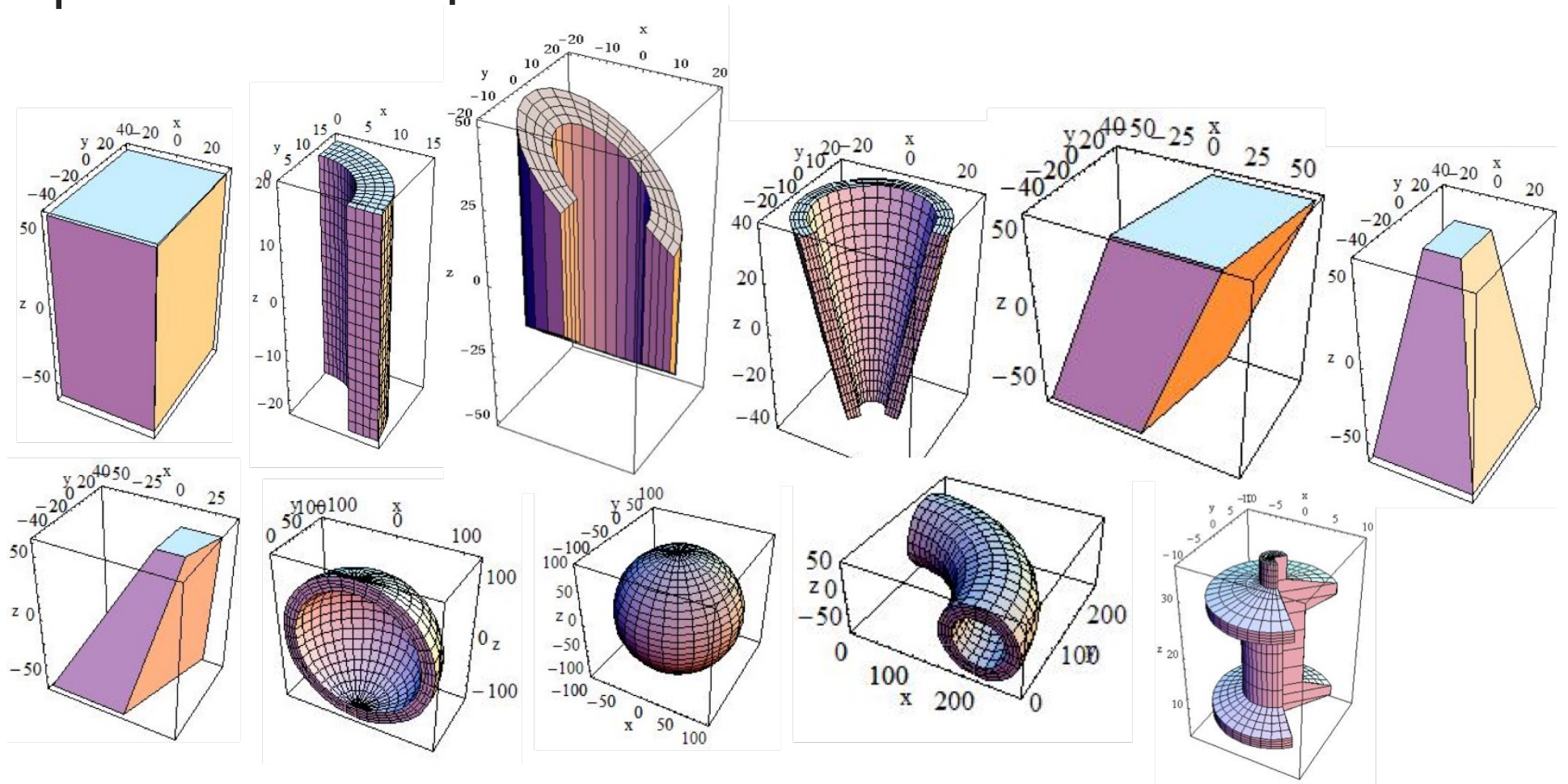
G4ThreeVector SurfaceNormal(G4ThreeVector)

Katı Geometriler



- Constructed Solid Geometries (CSG)
- Boundary Represented Shapes (BREPs)
- Complex shapes by parameterization
- Boolean operations also possible

Katı Geometriler



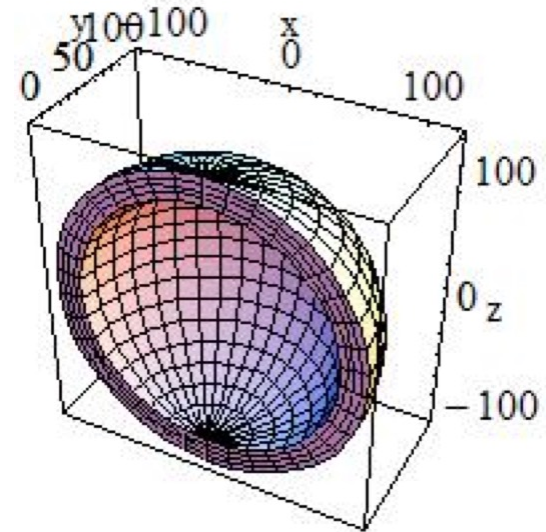
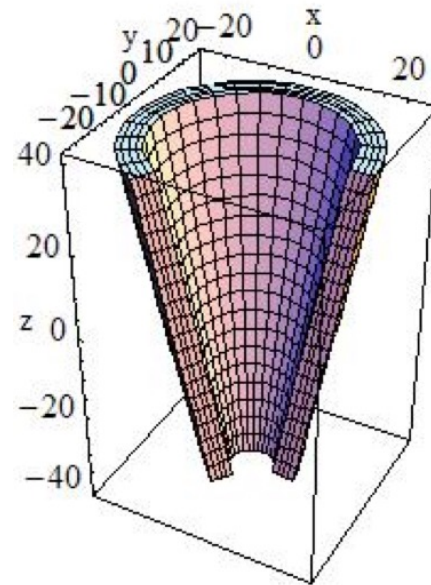
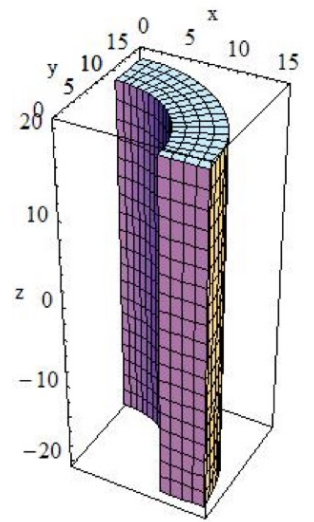
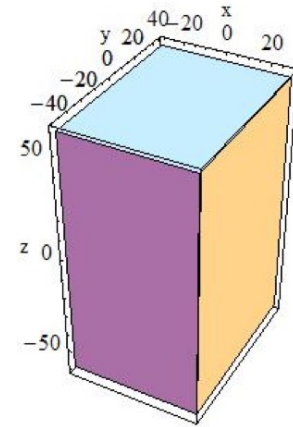
Katı Geometriler (Örnekler)

```
G4VSolid* boxSolid = new G4Box("aBox", 1.0*m, 1.0*m,  
1.0*m);
```

```
G4VSolid* tubeSolid = new G4Tubs("aTube",  
1.0*m, //inner radius (0 possible)  
2.0*m, //outer radius  
4.0*m // half - height  
0.*deg, 360.*deg); //segment angles
```

```
G4VSolid* coneSolid = new G4Cons("aCone",  
1.0*m, 1.5*m, //inner/outer radius 1  
2.0*m, 2.5*m, //inner/outer radius 2  
4.0*m // half - height  
0.*deg, 360.*deg); //segment angles
```

```
G4VSolid* sphereSolid = new G4Sphere("aSphere",  
1.0*m, 1.5*m //inner/outer radius  
0.*deg, 360.*deg, // phi  
0.*deg, 180.*deg); // theta
```



Mantıksal Hacimler (Logical Volumes)

- Katı Geometriler (solids) malzemelerin sadece boyutunu ve şeklini belirtir. Oysa bir deneyde sadece bu parametreleri bilmek yetmez.
- Hacimlere aşağıdaki özellikler verilebilir;
 - i. Materyal cinsi
 - ii. E / B Alanlar
 - iii. Görselleştirme Atıfları
 - iv. Hassasiyet
 - v. Kız hacimlerin pozisyonları
 - vi. Bölgeler / Lezyonlar
- Mantıksal Hacimler pozisyon ve rotasyonları tanımlamaz (Fiziksel nicelikler;; fiziksel hacim içinde tanımlanır.)
- Mantıksal Hacimler aynı tipteki fiziksel hacimler arasında paylaşılabilir.

Mantıksal Hacimler (Logical Volumes)

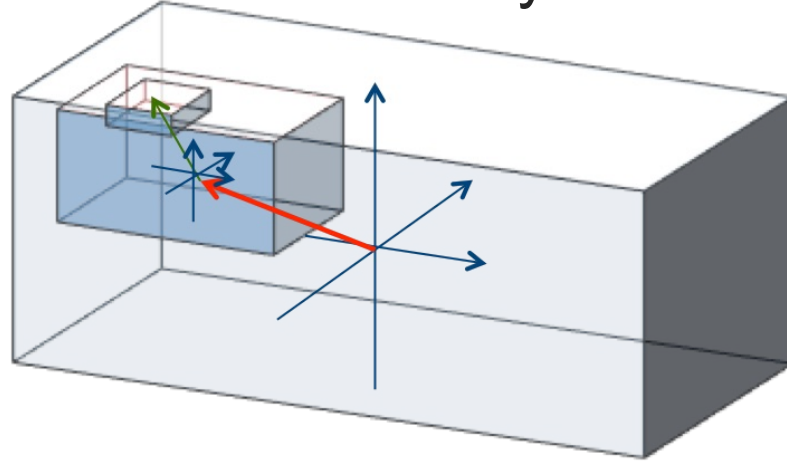
- Tanımlandıktan sonra mantıksal hacimlerin kaydı yapılmış olur.
- Kullanılması için başlatılması, fiziksel / katı bir hacime paylaştırılması gereklidir.
- Kullanılmazlarsa tanımlanmaları hatalı olsa bile program çalışır.

Örnek:

```
G4LogicalVolume(G4VSolid* pSolid,          //pointer to solid, must to be 0
                G4Material* pMat,          //pointer to material, must not be 0
                const G4String& name,      //name of this logical volume
                G4FieldManager* pFieldMgr = 0, //pointer to field manager
                G4VSensitiveDetector* pSensDet = 0, //sensitive detector
                G4UserLimits* pUserLimits = 0, // user limits
                G4bool optimise = true);    // optimization on
```

Fiziksel Hacimler (Physical Volumes)

- Oluşturulan katı ve mantıksal hacimlerin yerleşimi veya hareketi fiziksel hacim ile tanımlanır.
- Geant4 benzetimleri hacim geometrilerinin hiyerarşik yapısıyla oluşturulur:
 - En üst: Kök Hacim (veya World olarak da isimlendirilir)
 - İçeride: Ana hacimler (Tüm diğer kız hacimler Ana hacmin koordinat eksenini baz alırlar)
 - Daha içeride: Kız hacimler (Aynı ana hacme bağlı birden fazla olabilir)
- Fiziksel Hacimler **G4VPhysicalVolume** sınıfıyla üretilirler.
- Diğer hacimler gerçek fiziksel hacmi temsil ediyorsa **G4PVPlacement** kullanılır.



Fiziksel Hacimler (Örnek)

```
G4VSolid* boxSolid= new G4Box("aBox", 1.0*m, 1.0*m, 1.0*m);

G4LogicalVolume* boxLogic = new G4LogicalVolume(boxSolid, boxMat, "logicBox");

G4RotationMatrix* xRot = new G4RotationMatrix();
xRot->rotateX(M_PI/4*rad);

G4ThreeVector yTrans(0., 1.*m, 0.);

//constructor 1
G4VPhysicalVolume* boxPhys = new G4PVPlacement(xRot, yTrans, boxLogic,
"physicBox", motherLog, 0, copyNo, true);

//constructor 2
G4VPhysicalVolume* boxPhys = new G4PVPlacement(G4Transform3D(xRot, yTrans),
boxLogic, "physicBox", motherLog, 0, copyNo, true);

//constructor 3
G4VPhysicalVolume* boxPhys = new G4PVPlacement(xRot, yTrans, boxLogic,
"physicBox", motherPhys, 0, copyNo, true);
```

```
G4PVPlacement(G4RotationMatrix* pRot,    // rotation w.r.t. to mother volume
               const G4ThreeVector& trans, // translation w.r.t. mother
               G4LogicalVolume* pLog,    // solid logical volume
               const G4String& name,
               G4LogicalVolume* pMLog,   // mother logical volume
               G4bool pMany,              // not used
               G4int copyNo,              // set to 0 for first volume of type
               G4bool surfChk);           // check for overlaps
```

Fiziksel Hacimler (Örnek)

B1 Örneği için:

B1DetectorConstruction.cc

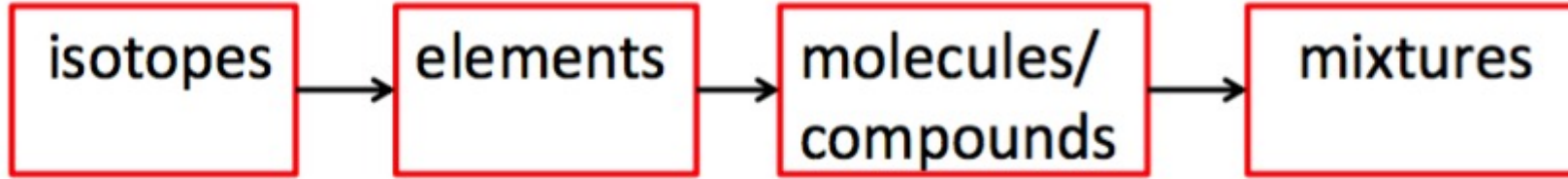
```
120 // Shape 1
121 //
122 G4Material* shape1_mat = nist->FindOrBuildMaterial("G4_A-150_TISSUE");
123 G4ThreeVector pos1 = G4ThreeVector(0, 2*cm, -7*cm);
124
125 // Conical section shape
126 G4double shape1_rmina = 0.*cm, shape1_rmaxa = 2.*cm;
127 G4double shape1_rminb = 0.*cm, shape1_rmaxb = 4.*cm;
128 G4double shape1_hz = 3.*cm;
129 G4double shape1_phimin = 0.*deg, shape1_phimax = 360.*deg;
130 G4Cons* solidShape1 =
131     new G4Cons("Shape1",
132               shape1_rmina, shape1_rmaxa, shape1_rminb, shape1_rmaxb, shape1_hz,
133               shape1_phimin, shape1_phimax);
134
135 G4LogicalVolume* logicShape1 =
136     new G4LogicalVolume(solidShape1,           //its solid
137                       shape1_mat,           //its material
138                       "Shape1");           //its name
139
140     new G4PVPlacement(0,                       //no rotation
141                     pos1,                     //at position
142                     logicShape1,             //its logical volume
143                     "Shape1",               //its name
144                     logicEnv,               //its mother volume
145                     false,                  //no boolean operation
146                     0,                       //copy number
147                     checkOverlaps);         //overlaps checking
148
```

Kısım 3:

Materyal Nesneleri ve Yönetimi

Materyal Nesneleri ve Yönetimi

- G4 materyalleri tanım olarak şunlar olabilir: İzotoplar, Elementler, Bileşikler (Compounds-Molecules), Alaşımlar (Alloys)



- Fiziksel durumları: Katı, sıvı veya gaz olabilir. (Plazma yok!)
- Çeşitli değerlerdeki basınç, sıcaklık veya yoğunluk durumlarında olabilirler.
- Şekildeki sırasıyla **G4Isotope**, **G4Element** ve **G4Material** sınıfları kullanılır.
- Karışımlar / Alaşımlar bu 3 sınıfın nesnelere kullanılarak oluşturulabilir.

Materyal Nesneleri ve Yönetimi

- Kullanıcılar kendi element (tümü olduğu halde) veya materyallerini yaratabilirler.
- G4 girilen bir materyal için sadece **yoğunluk (density)** bilgisini zorunlu tutmaktadır.
- Geri kalan bilgiler opsiyoneldir: fiziksel durum, sıcaklık, basınç (Aksi belirtilmediği sürece Standart koşullar geçerlidir.)
- Bazı anormal durumlar tanımlıdır: Standart Koşul dışındaki gazlar, yüksek basınçta katılar, çok düşük yoğunluklu sıvılar ...vs.
- NIST Materyal veritabanı kullanılabilir. Önemli bir kolaylıktır!
- Tüm tanımlı materyal listesi: <http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/ForApplicationDeveloper/html/Appendix/materialNames.html>

Materyal Nesneleri ve Yönetimi

G4 ile elementlerin kod içi tanımlanması:

- Materyalin atom sayısı, kütle numarası, yoğunluğu verilmelidir.

```
G4double density = 4.506*g/cm3;
```

```
G4double a = 47.867*g/mole;
```

```
G4Material* ti = new G4Material("pureTitanium", z=22, a, density);
```

- Vakum tanımlanması veya herhangi bir gazın ortama yerleştirilmesi:

```
G4NISTManager* manager = G4NISTManager::GetPointer;
```

```
G4Material* vacuum = manager->FindOrBuildMaterial("G4_Galactic");
```

Materyal Nesneleri ve Yönetimi

G4 ile moleküllerin kod içi tanımlanması:

- Molekül atom sayısı, kütle numarası, yoğunluğu ve yapısı verilmelidir.

```
G4double a = 1.01*g/mole;
```

```
G4Element* elH = new G4Element("Hydrogen", "H", z=1, a);
```

```
a = 16.00*g/mole;
```

```
G4Element* elO = new G4Element("Oxygen", "O", z=8, a);
```

```
G4double density = 1.0*g/cm3;
```

```
G4int ncomp = 2;
```

```
G4Material* H2O = new G4Material("Water", density, ncomp);
```

```
G4int nAtoms;
```

```
H2O->AddElement(elH, nAtoms=2);
```

```
H2O->AddElement(elO, nAtoms=1);
```

Materyal Nesneleri ve Yönetimi

G4 ile alaşım / karışımların kod içi tanımlanması:

- Alaşımdaki elementlerin atom sayısı, kütle numarası, yoğunluğu ve alaşımın yapısı verilmelidir.

```
G4Element* elC = ... ;    // define carbon
G4Material* H2O = ... ;    // define molecule (previous page)
G4Material* SiO2 = ... ;   // define another molecule

G4double density = 0.20*g/cm3;
G4int ncomp = 3;
G4double fracMass;
G4Material* Aerog = new G4Material("Aerogel", density, ncomp);

Aerog->AddMaterial(SiO2, fracMass = 62.5*perCent);
Aerog->AddMaterial(H2O, fracMass = 37.4*perCent);
Aerog->AddElement(elC, fracMass = 0.1*perCent);
```

Materyal Nesneleri ve Yönetimi

G4 ile izotopların kod içi tanımlanması:

- Aslında tanımlanan elementler de doğal birer izotoptur. Yoğunluk değerleri farklı girilse bile doğal çokluğuyla (abundance) kabul edilir.
- Doğal olmayan elementler ve farklı değerlerin girilmesi;

G4int z; G4int a;

G4Isotope* u235 = new G4Isotope("U235", z=92, a=235.,
235.044*g/mole);

G4Isotope* u238 = new G4Isotope("U238", z=92, a=238.,
238.051*g/mole);

G4Isotope sınıfıyla hiç olmadıkları kabul edilerek türetildiler !

Materyal Nesneleri ve Yönetimi

G4 ile bileşiklerin kod içi tanımlanması:

- Örnek: Nükleer reaktör için uranyum yakıtı (fuel)

```
G4int ncomp;
```

```
G4double abundance;
```

```
G4Element* enrichedU = new G4Element("enrichedU", ncomp=2);
```

```
enrichedU->AddIsotope(u235, abundance=5.0*perCent);
```

```
enrichedU->AddIsotope(u238, abundance=95.0*perCent);
```

```
G4Element* eIF = new G4Element("Fluorine", "F", 9., 18.998*g/mole);
```

```
G4double density;
```

```
G4Material* fuel = new G4Material("NuclearFuel", density = 5.09*g/cm3,  
                                ncomp=2, kStateGas, 640*kelvin, 1.5e7*pascal) ;
```

```
fuel->AddElement(eIF, 6);
```

```
fuel->AddElement(enrichedU, 1);
```

Kısım 4: Geant4 Örnekleri ve Dokümanları

- Geant4 kullanımı için önerilen ilk başvuru kaynağı;

www.geant4.org

Overview

Geant4 is a toolkit for the simulation of the passage of particles through matter. Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science. The three main reference papers for Geant4 are published in Nuclear Instruments and Methods in Physics Research A 506 (2003) 250-303 [↗](#), IEEE Transactions on Nuclear Science 53 No. 1 (2006) 270-278 [↗](#) and Nuclear Instruments and Methods in Physics Research A 835 (2016) 186-225 [↗](#).

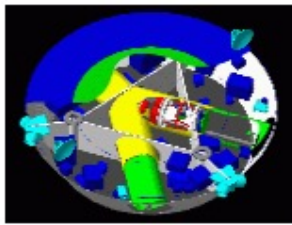
Applications



A sampling of applications, technology transfer and other uses of Geant4

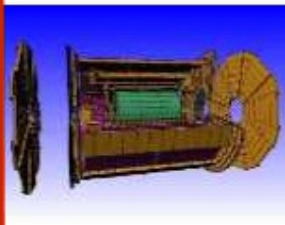
[printer-friendly version](#)

User Support



Getting started, guides and information for users and developers

Publications



Validation of Geant4, results from experiments and publications

Collaboration



Who we are: collaborating institutions, members, organization and legal information

News

- 7 Dec 2018
Release 10.5 is available from the [Download](#) area.
- 25 May 2018
Patch-02 to release 10.4 is available from the [source archive](#) [↗](#) area.
- 20 Oct 2017
Patch-03 to release 10.3 is available from the [source archive](#) [↗](#) area.

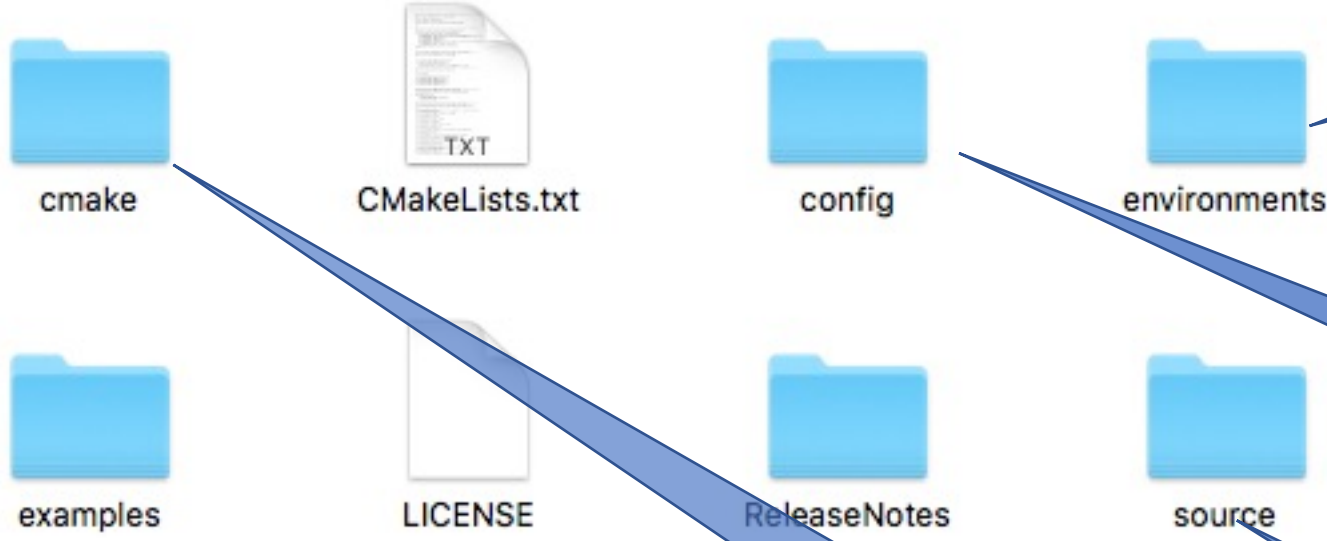
Events

- [First Geant4 Tutorial Workshop in São Paulo](#) [↗](#), São Paulo (Brazil), **4-8 February 2019**.
- [Geant4 Advanced Course @ CERN](#) [↗](#), CERN (Geneva), **26-27 March 2019**.
- [ENSAR2 workshop: Geant4 in nuclear physics](#) [↗](#), at CIEMAT, Madrid (Spain), **24-26 April 2019**.
- [Geant4 Course at the 16th Seminar on Software for Nuclear, Sub-nuclear and Applied Physics](#) [↗](#), Porto Conte, Alghero (Italy), **26-31 May 2019**.

Geant4 Örnekleri ve Dokümanları

- [User Documentation](#): User's Guides kısmında yer alan kitapçıklar
 1. **Installation Guide**
 2. **User's Guide: For Application Developers**
 3. **User's Guide: For Toolkit Developers**
 4. **Physics Reference Manuals**
 5. **Physics List Guide**

G4 kurulum klasörü: geant4.10.02.p02



G4Py (Python arayüzü) ve G4 MOMO (Java) arayüzü içerir.

Tüm .gmk uzantılı dosyalar (makefile için gerekli)

G4 dosyalarının cmake ile kurulumları için gereken dosyalar

G4 kodu kaynak dosyaları (Tüm modüller)

ÖRNEKLER BURADA !!!

● EXAMPLES:

1. **BASIC**
2. **EXTENDED**
3. **ADVANCED**

G4 yazılımı bu kısımda sunulan 50'den fazla örnek içerir.

README açıklamaları !!

ExampleB2a / B2b

- Simplified tracker geometry with global constant magnetic field
- Geometry with placements (G4PVPlacement) and parameterisation (G4PVParameterisation)
- Scoring within tracker via G4 sensitive detector and hits
- Geant4 physics list (FTFP_BERT) with step limiter

ExampleB3

- Schematic Positron Emitted Tomography system
- Geometry with simple placements with rotation (G4PVPlacement)
- Radioactive source - Scoring within Crystals via G4 scorers
- Modular physics list built via builders provided in Geant4

ExampleB4

- Simplified calorimeter with layers of two materials - Geometry with replica (G4PVReplica)
- Scoring within layers in four ways: via user actions (a), via user own object (b), via G4 sensitive detector and hits (c) and via scorers (d)
- Geant4 physics list (FTFP_BERT)
- Histograms (1D) and ntuple saved in the output file

ExampleB5

- A double-arm spectrometer with wire chambers, hodoscopes and calorimeters with a local constant magnetic field
- Geometry with placements with rotation, replicas and parameterisation
- Scoring within wire chambers, hodoscopes and calorimeters via G4 sensitive detector and hits - Geant4 physics list (FTFP_BERT) with step limiter
- UI commands defined using G4GenericMessenger
- Histograms (1D, 2D) and ntuple saved in the output file

● EXAMPLES:

1. **BASIC**
2. **EXTENDED**
3. **ADVANCED**

- analysis/
- biasing/
- common/
- electromagnetic/
- errorpropagation/
- eventgenerator/
- exoticphysics/
- field/
- g3tog4/
- geometry/
- hadronic/
- medical/
- optical/
- parallel/
- parameterisations/
- persisitency/
- polarisation/
- radioactivedecay/
- runAndEvent/
- visualization/

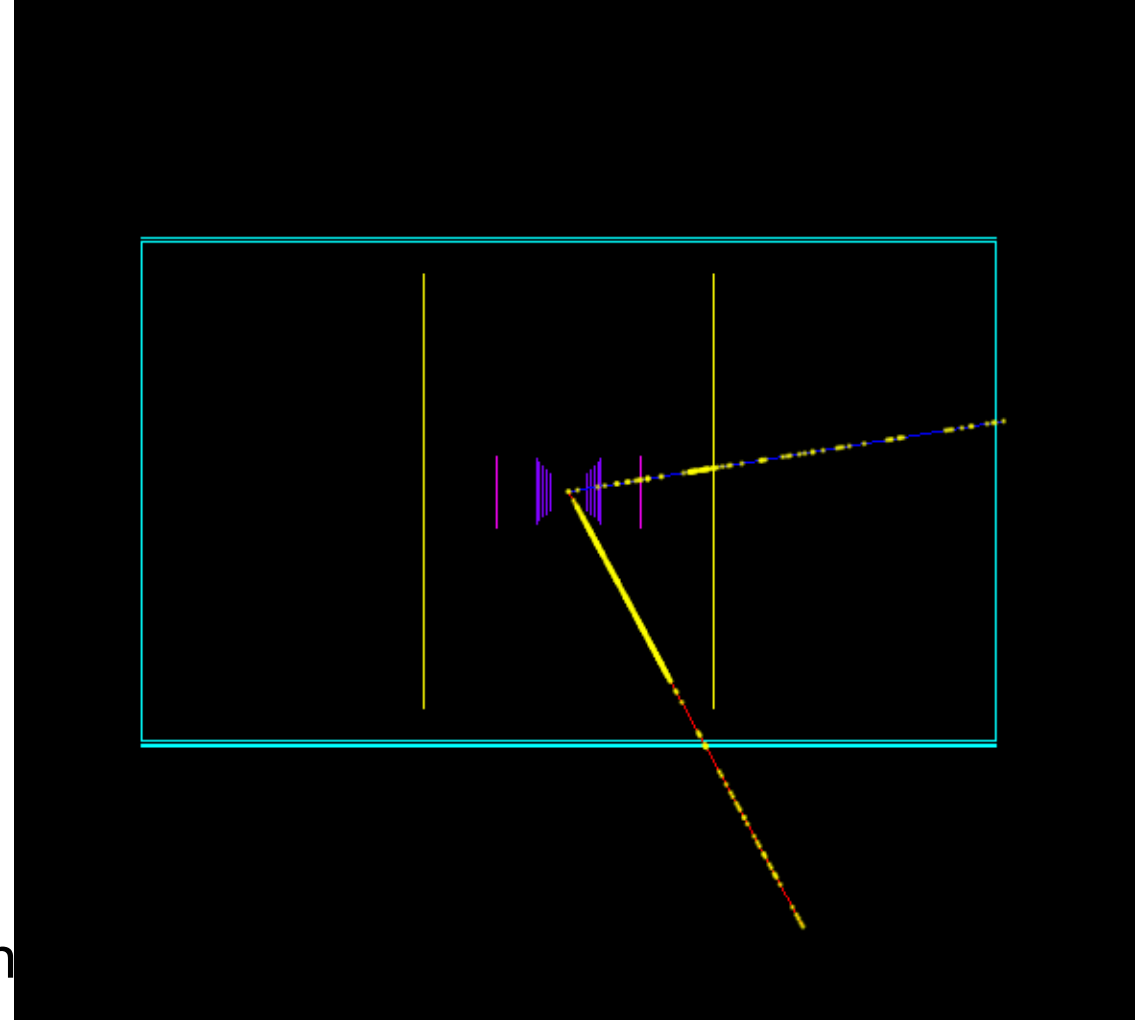
- EXAMPLES: G4 yazılımı bu kısımda sunulan 50'den fazla örnek içerir.

1. **BASIC**
2. **EXTENDED**
3. **ADVANCED**

- amsEcal/
- Brachytherapy/
- ChargeExchangeMC/
- Composite calorimeter/
- Dnaphysics/
- Dnageometry/
- Gamma-knife/
- Hadrontherapy/
- iort_therapy/
- Medical linac/
- Microbeam/
- Microdosimetry/
- Microelectronics/
- Nanobeam/
- Purging magnet/
- Radioprotection/
- RICH/
- Underground physics/
- X-ray fluorescence/
- X-ray telescope/

Geant4 Seçilmiş Örnekler

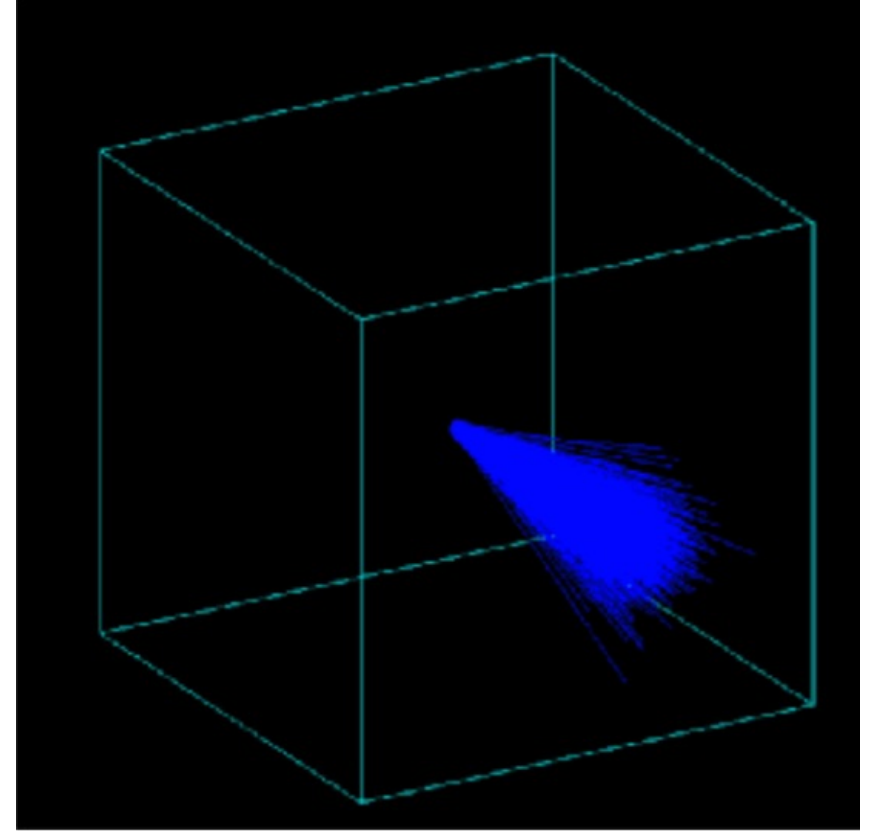
- EXTENDED > HADRONIC (01 -06)
Herhangi bir çekirdeğin orijin noktasında hedefe konularak hadronlarla bombardıman edilmesinin benzetimi yapılmaktadır.
- Tüm temel parçacıklar / iyonlar ve referans fizik listeleri kullanılabilir.
- Çıktı olarak root formatında tesir kesiti, ikincil parçacıkların enerji spektrumu, pozisyonu ve ışınımların dağılımları elde edilir.
- Neutron-hp için Fission Fragment modeli de eklenmiştir. (Su bazlı geometride fisyon ürünü içerme kabiliyetini göstermektedir.)
- Geometrisi basit kutu içinde hedef çekirdekler yerleştirilerek sağlanır.
- Detektör (örneğe bağlı) hedef etrafına yerleştirilen tüp şeklinde 0.5 cm ayırılarda yüzeylerdir. (SD Materyal eklenmemiş) .



● EXTENDED > RADIOACTIVE DECAY R01

Herhangi bir çekirdeğin radyoaktif bozunumunun temel çıktılarını (enerji spektrum, zaman, aktivite) veren örnektir.

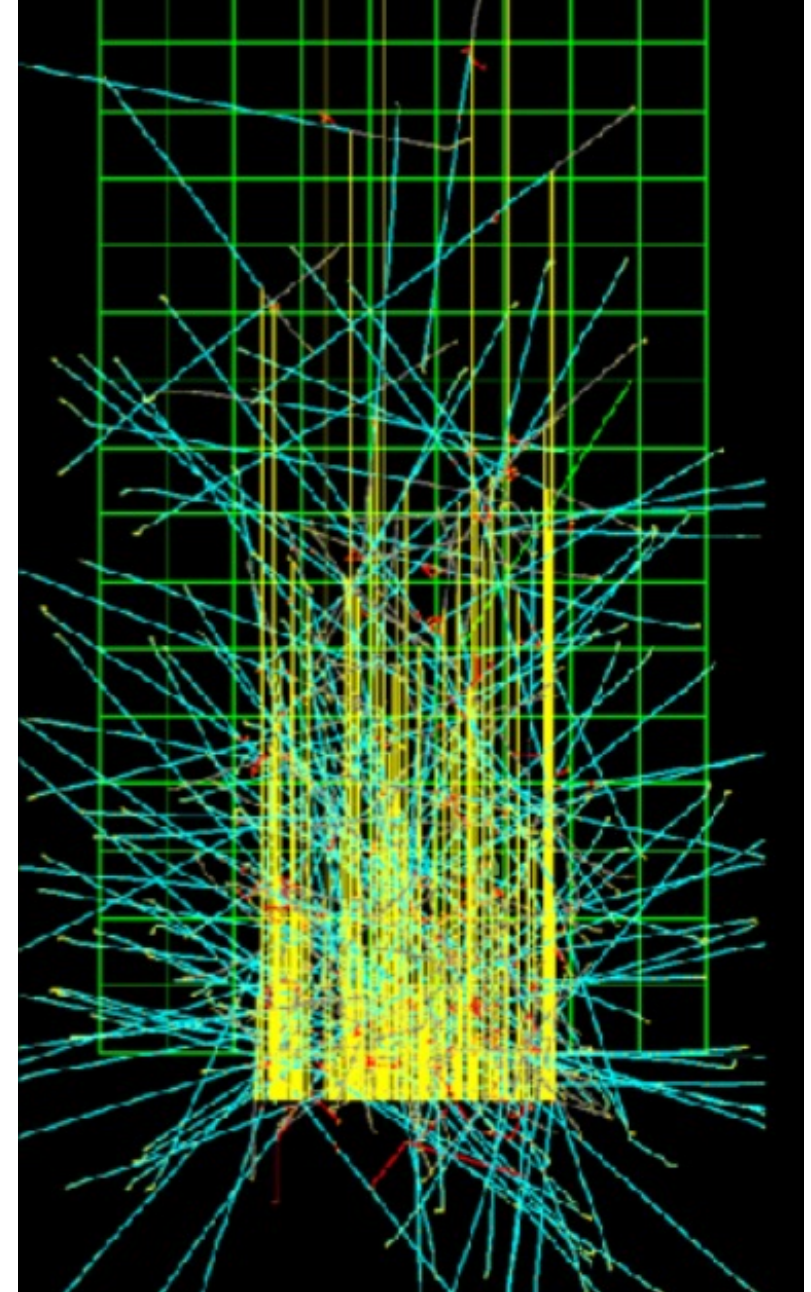
- Tüm parçacıklar ve iyonlar geantinodur. (enerji kaybı yaşanmaz).
- PhysicsList.cc içinde **G4RadioactiveDecay**, **G4Transportation** tanımlamaları parçacıkları ve süreçleri tanımlar.
- `/rdecay01/fullchain [True/False]` ile bozunum kanalı açar/kapanır. Tekil bozunum sağlar.
- Geometrisi basit kutu içinde sonsuz homojen ortam ile sağlanır.



● EXTENDED > RADIOACTIVE DECAY R02

Herhangi bir çekirdeğin radyoaktif bozunumunda seçilen bozunum kanallarını, zaman penceresi, bias ve varyans düşürme tekniği kullanarak verir.

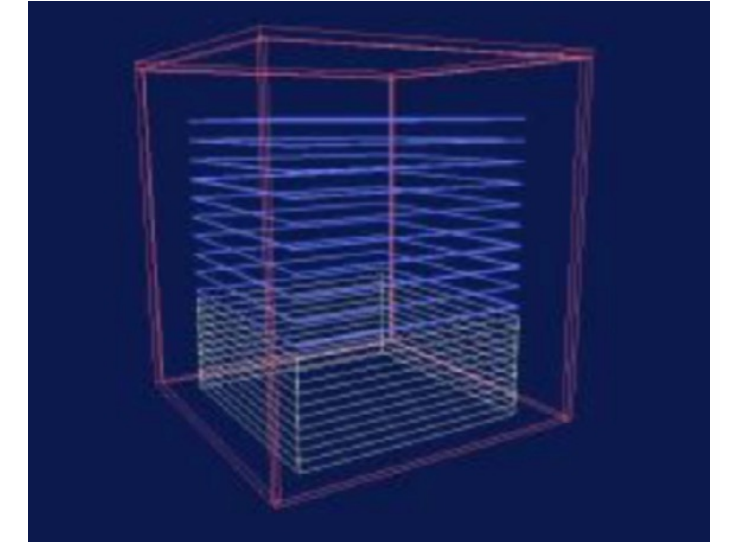
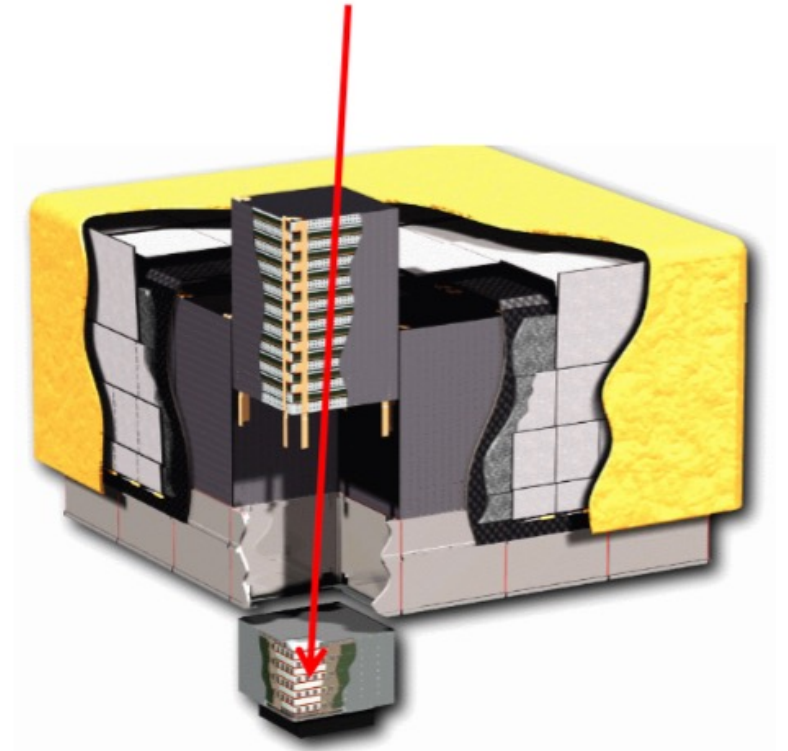
- Tüm parçacıklar ve ionlar geantinodur. (enerji kaybı yaşanmaz.)
- PhysicsList.cc içinde **G4RadioactiveDecay** ile hem radyoaktif izotopların hem de etkileşimlerde oluşan çekirdeklerin bozunumları ele alınır.
- `/grdm/selectVolume Target` ile kullanıcı hedeflenen (bozunum) hacimi sınırlandırabilir.
- Etkileşimlere EM ve Hadronik süreçler de eklenmiştir.
- Geniş bir Csl silindirik kutu içinde merkeze yerleştirilmiş küçük tüp şeklinde 2 cm x 5 cm bir Ge detektörü vardır.



● ADVANCED > GAMMA RAY TELESCOPE

G4 ile Uzay Uygulamasına örnek olabilecek çalışmadır. (Cirrone, PhD - INFN-LNS (Italy))

- Detektör; yüksek performanslı veri alabilen silikon düzlemsel izsürücü (tracker) şeritler ile tasarlanmıştır
- Detektör tracker dışında, CsI kalorimetreye ve anti-çakışma sistemine sahiptir.
- Enerji, 3B pozisyon ve zaman hitleri ASCII formatında alınabilmektedir.
- Histogramlama test aşamasındaki Lizard sistemi kullanılarak yapılmış.
- NASA tarafından uzaya 2008'de fırlatılan Fermi Gamma Ray teleskopu super-nova, pulsarlar, karadelikler hakkında veri ulaştırmaktadır.

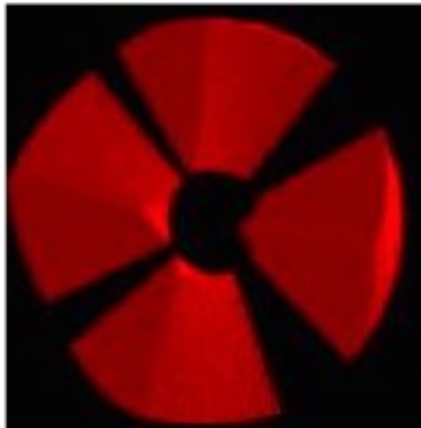


• ADVANCED > HADRON THERAPY (CATANA)

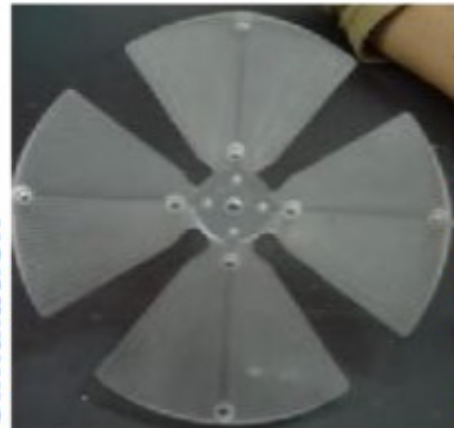
Proton demet hattı üzerinde kurulmuş göz tedavisine yönelik sistemin benzetimidir.

- IAEA (voxelized) standart geometri ile su fantomu içinde detektör tasarımı.
- Referans fizik listelerini ve iyon-iyon etkileşimleri için yerel (özel) fizik listelerini kullanmaktadır.
- Doz, LET ve RBE çıktıları alınabilir.
- INFN laboratuvarları (LNS ve TIFPA) tarafından oluşturulmuş/kullanılmaktadır.

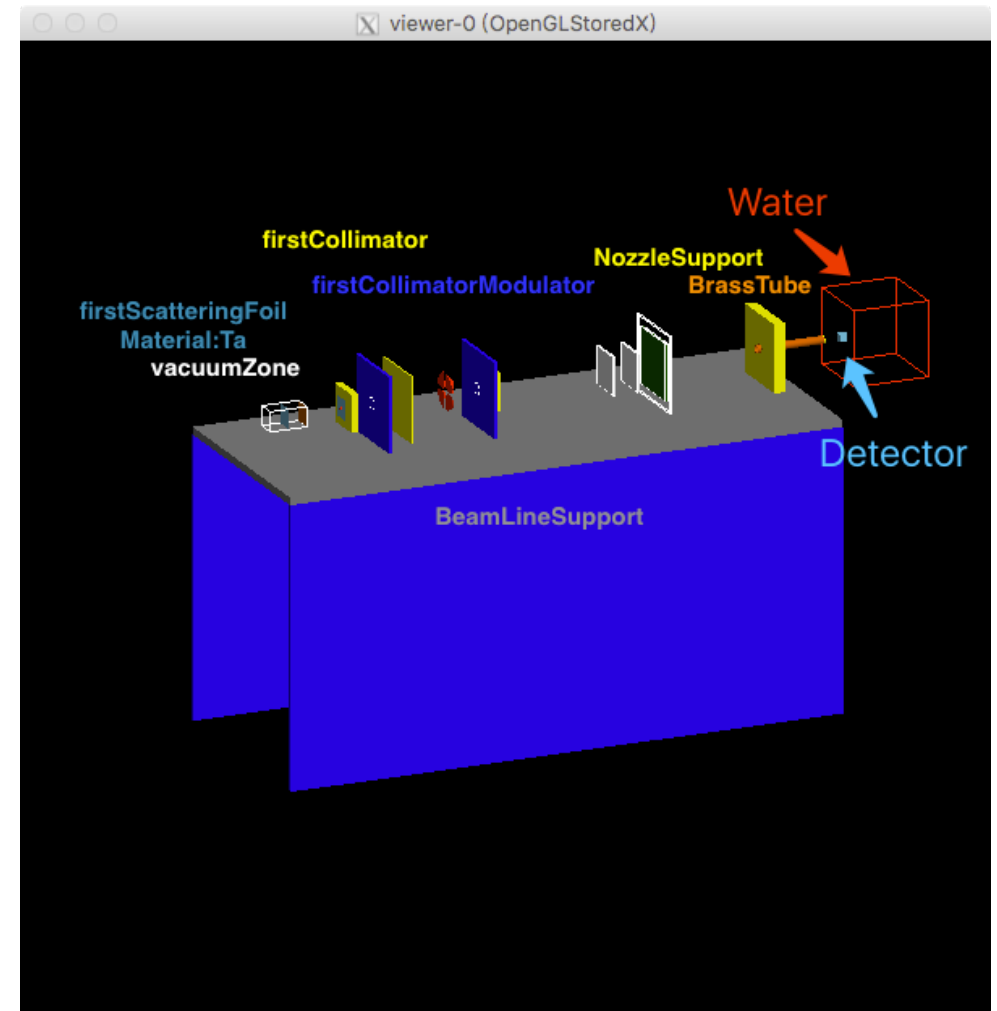
A dedicated class to the modulation system



**Monte Carlo
Simulation**



Real Modulator

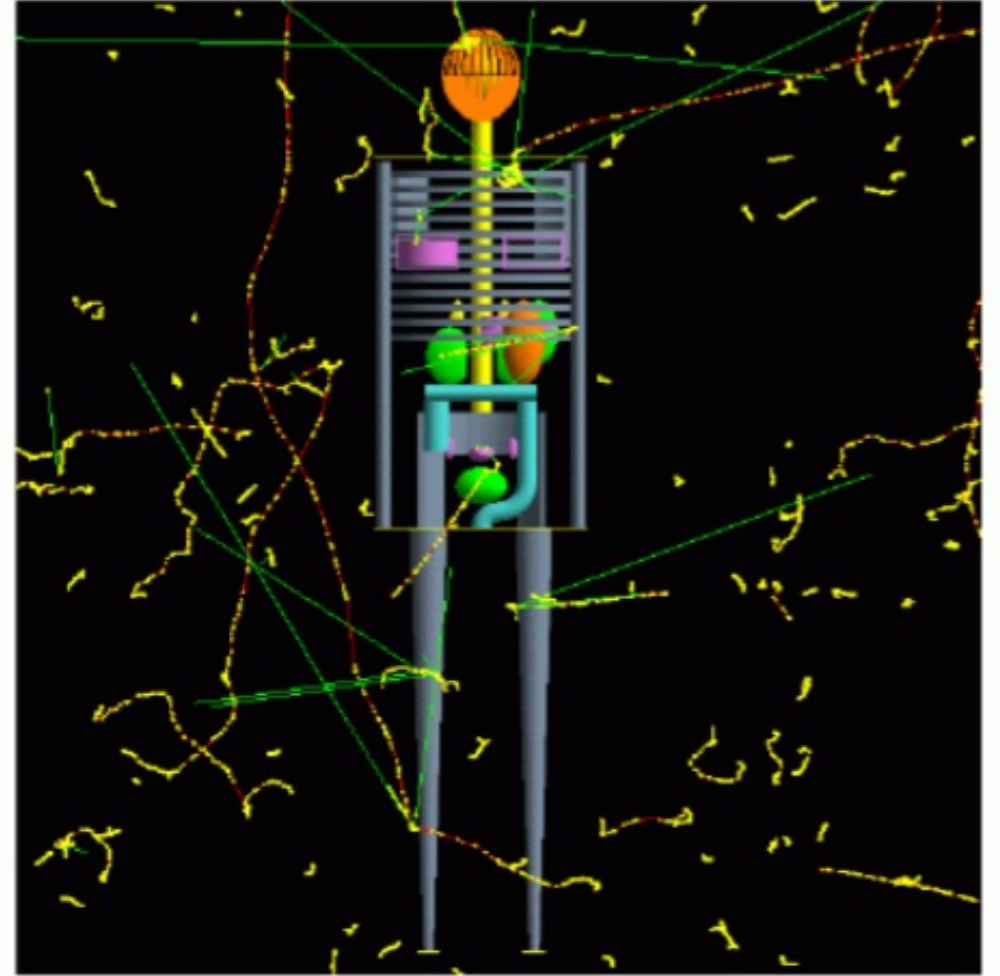


- ADVANCED > HUMAN PHANTOM

G4 için Antropomorfik kadın ve erkek fantom benzetimi yapılmıştır.

- Yaklaşımlara dayalı MIRD ve ORNL modelleri için ayrı benzetimler.
- GDML yoluyla geometriler implemante edilmiştir.
- Standart fizik süreçleri EM ile kullanılmış
- Yüklü parçacıklar, gamm ve geantino dahil edilmiştir.
- INFN - LNG laboratuvarı tarafından kullanılmak üzere üretilmiştir.

MIRD Female Phantom with particle tracks



- [G4 HyperNews Forumları: \(hypernews.slac.stanford.edu\)](http://hypernews.slac.stanford.edu) G4 ile ilgili tüm çıkmış problemler ve çözümleri için en yetkin adrestir.

GEANT4 at hypernews.slac.stanford.edu Forum List by Category



Forums by Category Forums by Time Order Request a New Forum	Recent Postings Search in Forums Subscribe to Forums	Member Info Members List New Member	Overview Contact Admin
---	--	---	---

Category: Applications

[Educational Applications](#) [Industrial instruments](#) [Medical Applications](#) [Space Applications](#)

Category: Control of runs, events, tracks, particles

[Event and Track Management](#) [Multithreading](#) [Particles](#) [Run Management](#)

Category: Experimental Setup

[Biasing and Scoring](#) [Fields: Magnetic and Otherwise](#) [Geometry](#) [Hits, Digitization and Pileup](#)

Category: General matters

[Documentation and Examples](#) [HyperNews System Announcements](#) [Hypernews Testing](#) [Installation and Configuration](#) [User Requirements](#)

Category: Interfaces

[\(Graphical\) User Interfaces](#) [Analysis](#) [Persistency](#) [Visualization](#)

Category: Physics

[Biasing and Scoring](#) [DNA/Very Low Energy](#) [Electromagnetic Processes](#) [Fast Simulation, Transportation & Others](#) [Hadronic Processes](#)
[Processes Involving Optical Photons](#)

Teşekkürler!

Kısım 5: Uygulama: Geant4 ile bir örneğın materyal ve geometrilerini deęiştirerek çalıştırınız.