

Hands-on on Geant4-DNA **physics**

geant4-dna.org

The « dnaphysics » extended example

Batmunkh MUNKHBAATAR

NUM & JINR, Ulaanbaatar, Mongolia & Dubna, Russia
batmunkh@jinr.ru

Sébastien INCERTI

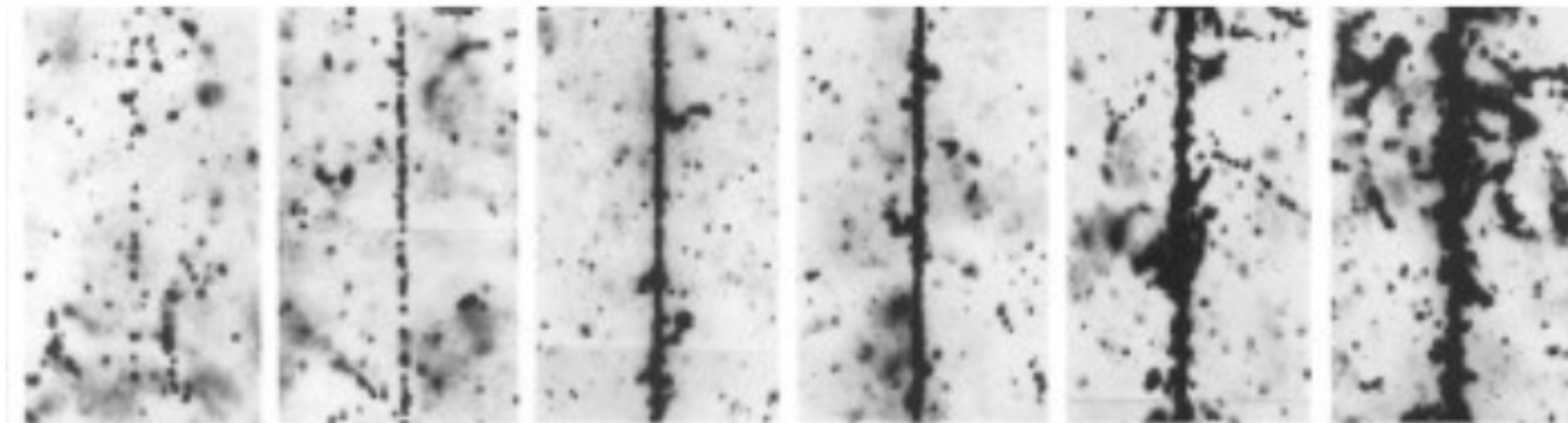
LP2i, CNRS / IN2P3, Bordeaux, France
sebastien.incerti@cnrs.fr

Geant4-DNA tutorial
Pohang Accelerator Laboratory, Republic of Korea
07/02/2025

Geant4 version 11.3
Released in December 2024

Tracks of charged particles in nuclear emulsion

- **Physical information on track structure** (spatial distribution of energy deposition, interaction types, etc.) is used in many domains involving ionising radiation
 - microdosimetry, radiation biology, biophysical modeling, clinical radiotherapy, space radiation, environmental science, solid state physics, accelerator design...



H¹⁺

He²⁺

C⁶⁺

O⁸⁺

Si¹⁴⁺

Ca²⁰⁺

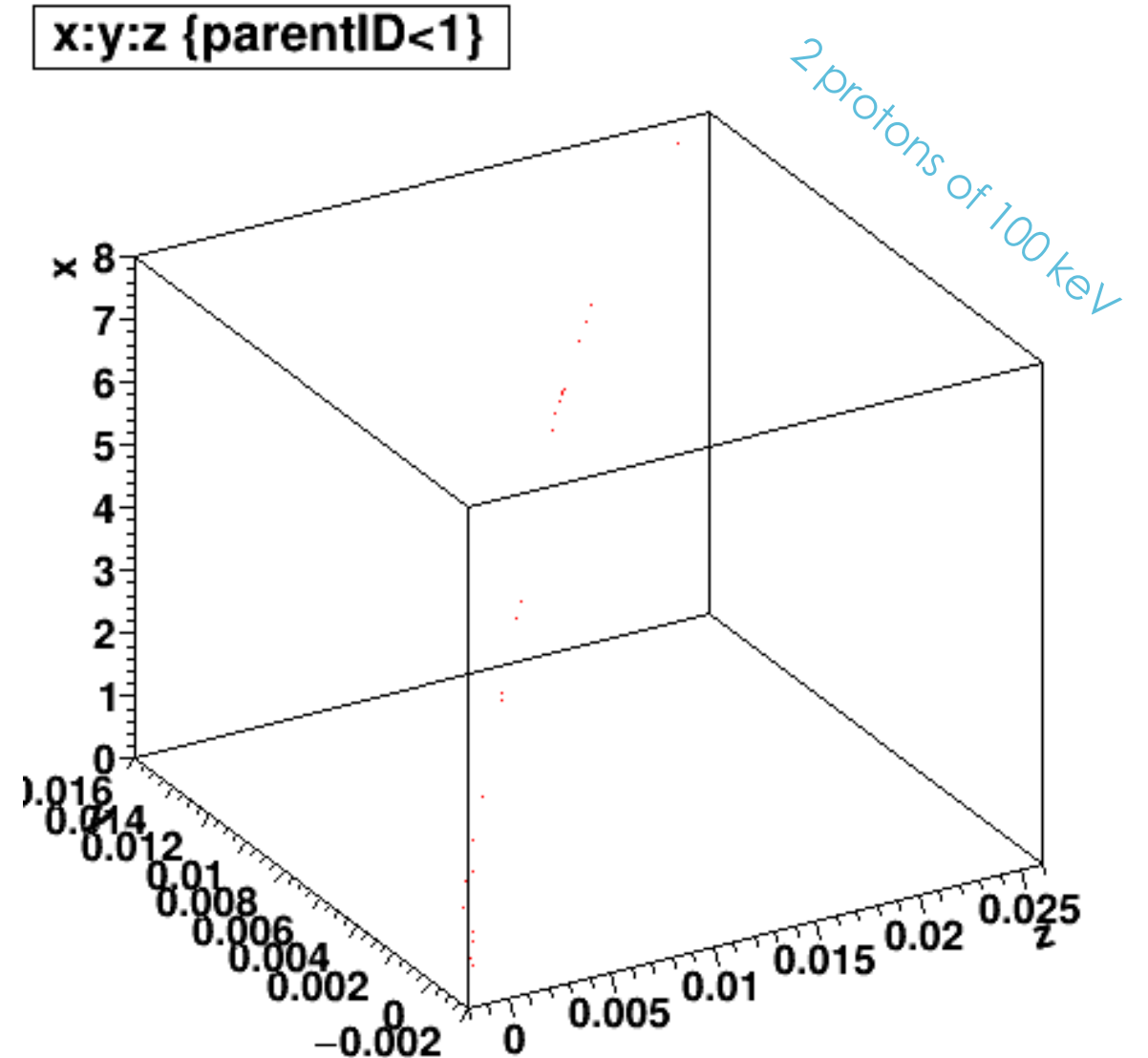
« Study of Elementary Particles by the Photographic Method »
C.F. Powell *et al.* (1959)

The « dnaphysics » extended example

- We are going to try the « **dnaphysics** » extended example which makes use of **Geant4-DNA Physics processes and models**
 - This is the « **simplest** » Geant4–DNA example
- This is a Geant4 « extended » example and it is located in
`$G4INSTALL/examples/extended/medical/dna`
- Remember that all Geant4-DNA examples are listed at:
<http://geant4-dna.org>
https://geant4-userdoc.web.cern.ch/Doxygen/examples_doc/html/Examples_dna.html

The « dnaphysics » extended example

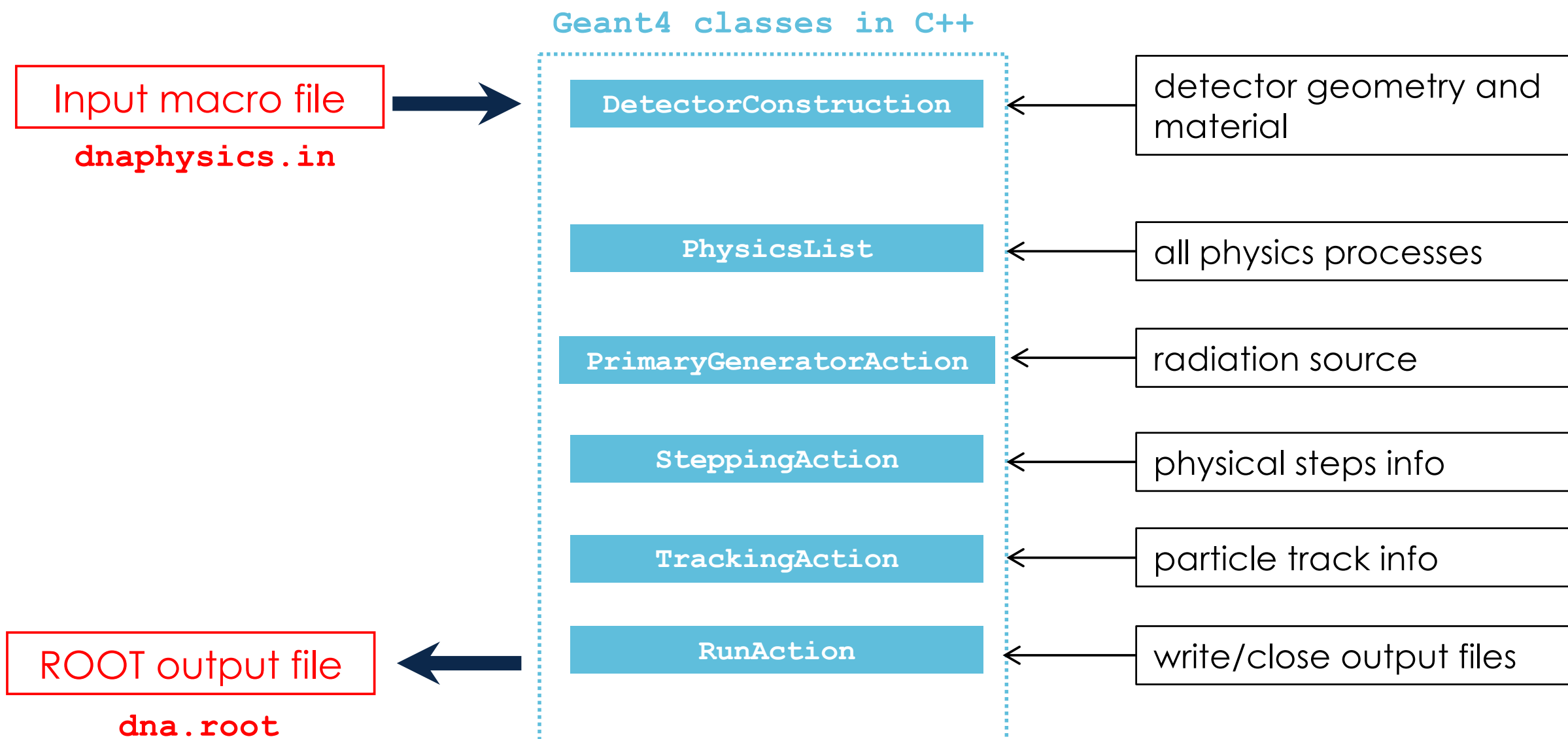
- Users can simulate experimentally-observed 3D **track structure** of different particles
- They can learn how to **extract all physics information** from this track structure, for example:
 - Type of **particle**
 - Type of **physics process** at each step
 - **Coordinates** in space
 - Energy **deposit**
 - **Step length** and number
 - **Kinetic energy**
 - **Scattering angle**
 - ID of **event, track, parent track**
 - ...



The « dnaphysics » extended example

- This example aims to demonstrate how to **use the Geant4-DNA physics processes and models in a liquid water volume**
 - All particle interactions are simulated on a **step-by-step basis** (« **discrete** » processes)
 - The PhysicsList class uses the **Geant4-DNA Physics constructor** called `G4EmDNAPhysics_option2`, so you do not need to code the physics yourself
 - You can easily switch from one constructor to another (e.g. `option4`, `option6`)
- It also explains how to **easily change the density of the target material** (liquid water): this is the « **variable density material** » feature of Geant4
 - Users can investigate density change effects
 - E.g. `1.06 g/cm3` average density of cell nucleus (cf. PARTRAC code)
- Produces a **ROOT ntuple** to analyze & visualize track structures

Structure of the « dnaphysics » extended example



« dnaphysics » extended example: input macro commands (UI)

```
# Verbosity
/tracking/verbose 0
/run/verbose 2
/control/verbose 2
#
# MT
/run/numberOfThreads 2
#
# Material
/dna/test/setMat G4_WATER
# or alternatively
#/dna/test/setMatDens G4_WATER_MODIFIED 1.200 g/cm3
#
# Size of World volume
/dna/test/setSize 100 um
#
# Atomic deexcitation
/process/em/fluor true
/process/em/augetrue true
/process/em/augetrueCascade true
/process/em/deexcitationIgnoreCut true
#
# Physics
# - To use Geant4-DNA constructor X, X=0, 2, 4, or 6 (recommended)
#/dna/test/addPhysics DNA_Opt0
/dna/test/addPhysics DNA_Opt2
#/dna/test/addPhysics DNA_Opt4
#/dna/test/addPhysics DNA_Opt6
#
# - To add radioactive decay physics
/dna/test/addPhysics raddecay
#
# Heavy ions tracking cut
#/dna/test/addIonsTrackingCut false
#
# Run initialization
/run/initialize
#
# Visualization
#/control/execute vis.mac
#
# Incident particle type
#/gun/particle e-
/gun/particle proton
#/gun/particle hydrogen
#/gun/particle alpha
#/gun/particle alpha+
#/gun/particle helium
#/gun/particle ion
#/gun/ion 14 28
#
# Incident particle energy
/gun/energy 100 keV
#
# Beam on
/run/beamOn 2
```

Control of verbosity

Select **number of threads**

Select the target material

Set World size

Activate fluorescence, Auger electron production and Auger cascade, without production cuts

Select Geant4-DNA **Physics constructor**

Add radioactive decay physics

Tracking cut for heavy ions

Activate visualization

Choose incident particle

Set incident **particle energy**

Shoot particles

« dnaphysics » extended example: **DetectorConstruction** class

■ Detector **material**:

- The « World » volume contains **liquid water**, defined as « **G4_WATER** » (= Geant4 NIST material* for liquid water). Density is **1 g/cm³**

```
// Water is defined from NIST material database
G4NistManager * man = G4NistManager::Instance();

G4Material * H2O = man->FindOrBuildMaterial("G4_WATER");

/*
If one wishes to test other density value for water material,
one should use instead:
G4Material * H2O = man->BuildMaterialWithNewDensity("G4_WATER_MODIFIED",
"G4_WATER",1.100*g/cm3);
```

Users can change density value!

See source file
[dnaphysics/src/DetectorConstruction.cc](#)

« dnaphysics » extended example: **DetectorConstruction** class

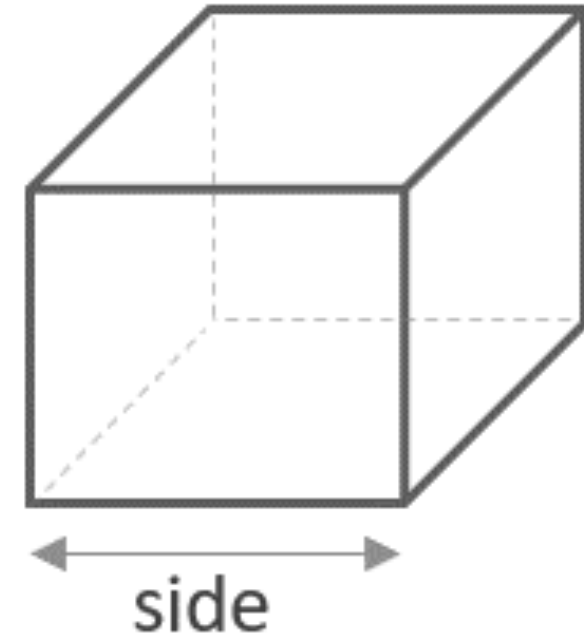
- Detector **geometry**:
 - The « World » volume is a **cube** (side size is 100 micrometer, see constructor)

```
// World volume
G4double worldSizeX = fWorldSize;
G4double worldSizeY = worldSizeX;
G4double worldSizeZ = worldSizeX;

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

fLogicWorld = new G4LogicalVolume(solidWorld, // its solid
                                  fpWaterMaterial, // its material
                                  "World"); // its name

fPhysiWorld = new G4PVPlacement(0, // no rotation
                                G4ThreeVector(), // at (0,0,0)
                                "World", // its name
                                fLogicWorld, // its logical volume
                                0, // its mother volume
                                false, // no boolean operation
                                0); // copy number
```



See source file
[dnaphysics/src/DetectorConstruction.cc](https://github.com/GEANT4/dnaphysics/blob/master/src/DetectorConstruction.cc)

« dnaphysics » extended example: **PhysicsList** class

- **Physics constructor** and cut-off parameters:
 - The PhysicsList class uses the default **G4EmDNAPhysics_option2** Physics constructor

G4EmDNAPhysics_option2 constructor

```
PhysicsList::PhysicsList() : G4VModularPhysicsList()
{
    SetDefaultCutValue(1.0 * micrometer);
    SetVerboseLevel(1);

    fEmPhysics = "DNA_Opt2";
    fEmPhysicsList = new G4EmDNAPhysics_option2();
    fDecayPhysicsList = new G4DecayPhysics();

    G4ProductionCutsTable::GetProductionCutsTable()->SetEnergyRange(100 * eV, 1 * GeV);
    G4EmParameters* param = G4EmParameters::Instance();
    param->SetMinEnergy(100 * eV);
    param->SetMaxEnergy(1 * GeV);
}
```

See source file [dnaphysics/src/PhysicsList.cc](#)

Note

- For energies above than the maximum energy covered by Geant4-DNA models, Geant4 standard electromagnetic models are used (beta)

« dnaphysics » extended example: PrimaryGeneratorAction class

■ Radiation source:

- This class describes the source of incident particles:
 - type, kinetic energy, position, angular distribution...

```
#include "PrimaryGeneratorAction.hh"

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo...

PrimaryGeneratorAction::PrimaryGeneratorAction()
: G4VUserPrimaryGeneratorAction(),
  fpParticleGun(0)
{
  G4int n_particle = 1;
  fpParticleGun = new G4ParticleGun(n_particle);
}
```

See source file

[dnaphysics/src/PrimaryGeneratorAction.cc](#)

The macro file ([dnaphysics.in](#)) contains UI* commands:

```
/gun/particle proton
/gun/energy 100 keV
/gun/position 0 0 0 um
/gun/direction 0 0 1
/run/beamOn 1
```

* See e.g. http://www.hep.ph.ic.ac.uk/~yoshiu/COMET/comet_g4HTMLdoc/_gun_.html

« dnaphysics » extended example: **SteppingAction** class

■ **Physical steps** of particle track:

- This class records useful information about each physical interaction: particle type, kinetic energy, position, energy deposition, etc.

```
if (partDef == G4Electron::ElectronDefinition())  
    flagParticle = 1;
```

Users can **flag a particle**

```
G4int procID = postStep->GetProcessDefinedStep()->GetProcessSubType();  
  
else if (flagParticle == 1) {  
    if (procID == 58)  
        flagProcess = 10;
```

Users can **flag a process**
(see **README** for numbering)

(Note: string comparison of process name is possible but is not recommended)

```
G4AnalysisManager* analysisManager = G4AnalysisManager::Instance();  
  
// fill ntuple  
analysisManager->FillNtupleDColumn(0, flagParticle);  
analysisManager->FillNtupleDColumn(1, flagProcess);  
analysisManager->FillNtupleDColumn(2, xp);  
analysisManager->FillNtupleDColumn(3, yp);  
analysisManager->FillNtupleDColumn(4, zp);  
analysisManager->FillNtupleDColumn(5, step->GetTotalEnergyDeposit() / eV);
```

G4AnalysisManager

Users can **extract step information** about primary/secondary particles, position of energy depositions... and **fill ROOT ntuple(s) or histograms** with such information

« dnaphysics » extended example: **TrackingAction** class

- **Physical properties** of particle **track**:
 - This class records useful information about each physical track

```
// Call analysis manager
G4AnalysisManager* analysisManager = G4AnalysisManager::Instance();

// Fill track information ntuple
analysisManager->FillNtupleDColumn(1, 0, flagParticle);
analysisManager->FillNtupleDColumn(1, 1, x);
analysisManager->FillNtupleDColumn(1, 2, y);
analysisManager->FillNtupleDColumn(1, 3, z);
analysisManager->FillNtupleDColumn(1, 4, dirx);
analysisManager->FillNtupleDColumn(1, 5, diry);
analysisManager->FillNtupleDColumn(1, 6, dirz);
analysisManager->FillNtupleDColumn(1, 7, aTrack->GetKineticEnergy() / eV);
analysisManager->FillNtupleIColumn(1, 8, aTrack->GetTrackID());
analysisManager->FillNtupleIColumn(1, 9, aTrack->GetParentID());
analysisManager->AddNtupleRow(1);
```

G4AnalysisManager

Users can extract **track information** such as: type of particle for the current track, track position, momentum direction, kinetic energy, ID and parent track ID

« dnaphysics » extended example: RunAction class

■ Write and close simulation results:

- This class is used to create ROOT ntuple(s) and histograms, and to save them in a ROOT file (`dna.root`) at the end of the run
- The singleton `G4AnalysisManager` is the entity in charge of collecting and accumulating and saving all information

```
void RunAction::EndOfRunAction(const G4Run* aRun)
{
    G4int nofEvents = aRun->GetNumberOfEvent();
    if (nofEvents == 0) return;

    // Print histogram statistics
    auto analysisManager = G4AnalysisManager::Instance();

    // Save histograms
    analysisManager->Write();
    analysisManager->CloseFile();
}
```

G4AnalysisManager

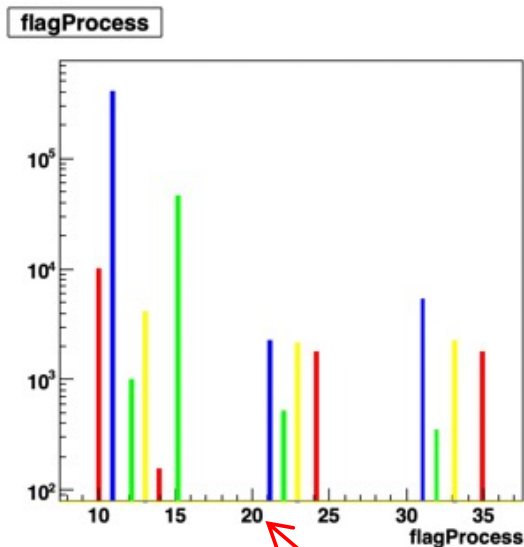
Note

- Save histograms in method `EndOfRunAction()`
- In MT mode, each thread creates and fills a local ROOT file; all files are then merged automatically at the end of the simulation

« dnaphysics » extended example: output

- Results are saved in the `dna.root` file and can be analyzed using ROOT*
 - Type `root plot.C`

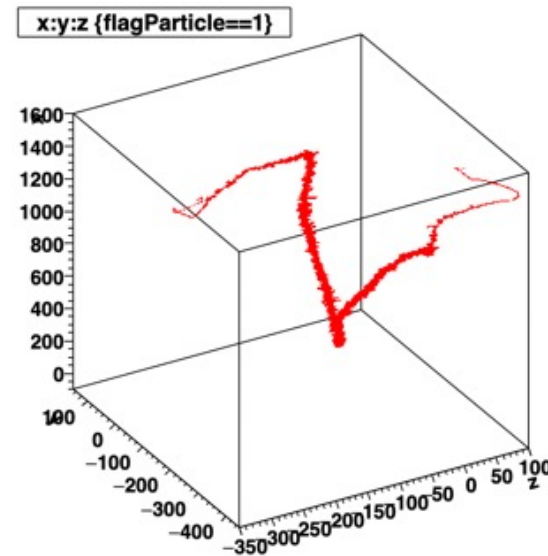
Number of occurrences of physical processes



(See [README](#) for process numbers)

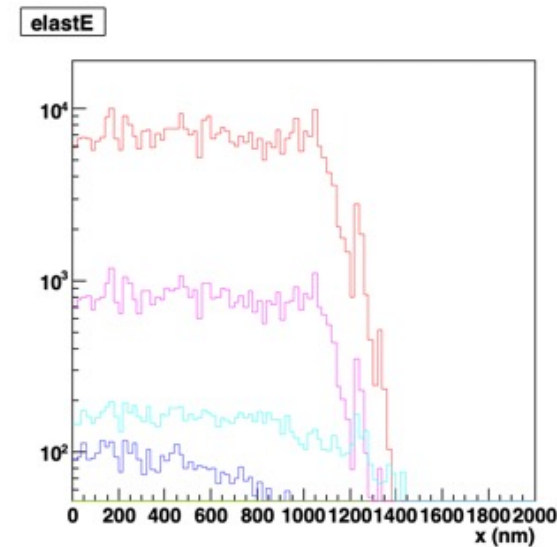
```
proton_G4DNAElastic: 21
proton_G4DNAExcitation: 22
proton_G4DNAIonisation: 23
proton_G4DNAChargeDecrease: 24
```

3D visualization of 2 proton tracks



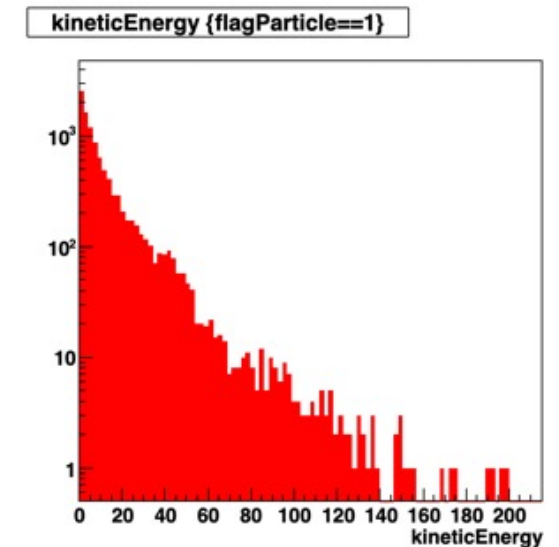
2 protons
(100 keV)

Occurring processes along x-axis



Elastic
Vib. excitation
Solvation
Ionisation
...

Energy distribution of electrons



Note: We also provide the `deexcitation.in` and `plotDeexcitation.C` macros to illustrate atomic deexcitation (fluorescence photons and Auger electrons)

Hands-on practice with the « dnaphysics » extended example

Hands-on practice with the « dnaphysics » extended example

- **Copy** the **dnaphysics** extended example to your local directory, **create** your build directory and **compile** dnaphysics

```
cd
```

```
cp -r $G4EXAMPLES/extended/medical/dna/dnaphysics .
```

```
cd dnaphysics
```

```
mkdir build-dnaphysics
```

```
cd build-dnaphysics
```

```
cmake ../
```

```
make
```

or `make -jN` if you have **N** cores

```
localhost.localdomain:/build-dnaphysics < 80 >make -j2
[ 22%] Building CXX object CMakeFiles/dnaphysics.dir/src/ActionInitialization.cc.o
[ 22%] Building CXX object CMakeFiles/dnaphysics.dir/dnaphysics.cc.o
[ 33%] Building CXX object CMakeFiles/dnaphysics.dir/src/DetectorConstruction.cc.o
[ 44%] Building CXX object CMakeFiles/dnaphysics.dir/src/DetectorMessenger.cc.o
[ 55%] Building CXX object CMakeFiles/dnaphysics.dir/src/PhysicsList.cc.o
[ 66%] Building CXX object CMakeFiles/dnaphysics.dir/src/PrimaryGeneratorAction.cc.o
[ 77%] Building CXX object CMakeFiles/dnaphysics.dir/src/RunAction.cc.o
[ 88%] Building CXX object CMakeFiles/dnaphysics.dir/src/SteppingAction.cc.o
[100%] Linking CXX executable dnaphysics
[100%] Built target dnaphysics
```

Hands-on practice with the « dnaphysics » extended example

- **Run** dnaphysics using the provided **macro file**:

`./dnaphysics dnaphysics.in`

```
# Verbosity
/tracking/verbose 0
/run/verbose 2
/control/verbose 2
#
# MT
/run/numberOfThreads 2
#
# Material
/dna/test/setMat G4_WATER
# or alternatively
#/dna/test/setMatDens G4_WATER_MODIFIED 1.200 g/cm3
#
# Size of World volume
/dna/test/setSize 100 um
#
# Atomic deexcitation
/process/em/fluo true
/process/em/augetrue true
/process/em/augetrueCascade true
/process/em/deexcitationIgnoreCut true
#
# Physics
# - To use Geant4-DNA constructor X, X=0, 2, 4, or 6 (recommended)
#/dna/test/addPhysics DNA_Opt0
#/dna/test/addPhysics DNA_Opt2
#/dna/test/addPhysics DNA_Opt4
#/dna/test/addPhysics DNA_Opt6
#
# - To add radioactive radioactive decay
/dna/test/addPhysics raddecay
#
# Heavy ions tracking cut
#/dna/test/addIonsTrackingCut false
#
# Run initialization
/run/initialize
#
# Visualization
#/control/execute vis.mac
#
# Incident particle type
#/gun/particle e-
#/gun/particle proton
#/gun/particle hydrogen
#/gun/particle alpha
#/gun/particle alpha+
#/gun/particle helium
#/gun/particle ion
#/gun/ion 14 28
#
# Incident particle energy
#/gun/energy 100 keV
#
# Beam on
/run/beamOn 2
```

Geant4-DNA physics **option 2**

Incident particle is **proton**

2 protons of **100 keV** are shot

Hands-on practice with the « dnaphysics » extended example

- Results are saved in `dna.root` and can be analyzed using ROOT (`plot.C`)

`root plot.C`

```
// All
ntuple->SetFillStyle(1001);
ntuple->SetFillColor(2);
ntuple->Draw("flagProcess","", "B");

// Excitation
ntuple->SetFillStyle(1001);
ntuple->SetFillColor(3);
ntuple->Draw("flagProcess","flagProcess==12 || flagProcess==15 || flagProcess==22|

// Elastic
ntuple->SetFillStyle(1001);
ntuple->SetFillColor(4);
ntuple->Draw("flagProcess","flagProcess==11 || flagProcess==21 || flagProcess==31|

// Ionisation
ntuple->SetFillStyle(1001);
ntuple->SetFillColor(5);
ntuple->Draw("flagProcess","flagProcess==13 || flagProcess==23 || flagProcess==33|
```

Plot occurring physical processes

e-_G4DNAElectronSolvation	10
e-_G4DNAElastic	11
e-_G4DNAExcitation	12
e-_G4DNAIonisation	13
e-_G4DNAAttachment	14
e-_G4DNAVibExcitation	15
proton_G4DNAElastic	21
proton_G4DNAExcitation	22
proton_G4DNAIonisation	23
proton_G4DNAChargeDecrease	24

(See [README](#) for
process numbers)

```
c1->cd(2);

ntuple->SetMarkerColor(2);

ntuple->Draw("x:y:z","flagParticle==1");

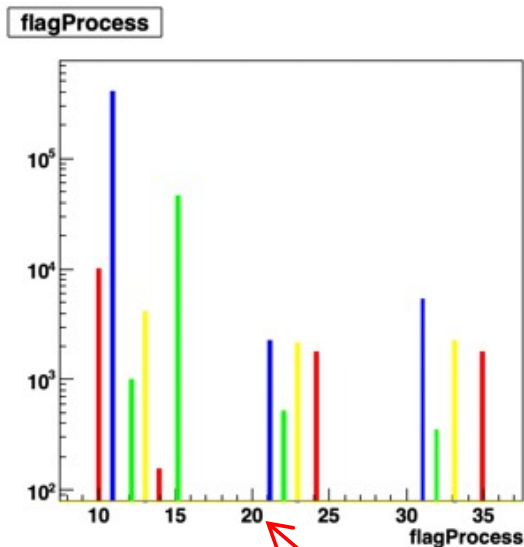
//ntuple->SetMarkerColor(4);
//ntuple->SetMarkerSize(4);
//ntuple->Draw("x:y:z/1000","flagParticle==4 || flagParticle==5 || flagParticle==6","same");
```

Draw processes in track

« dnaphysics » extended example: output

- Results are saved in the `dna.root` file and can be analyzed using ROOT*
 - Type `root plot.C`

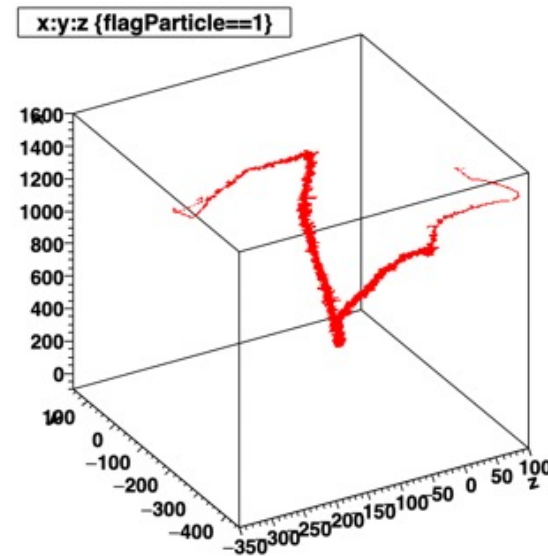
Number of occurrences of physical processes



(See [README](#) for process numbers)

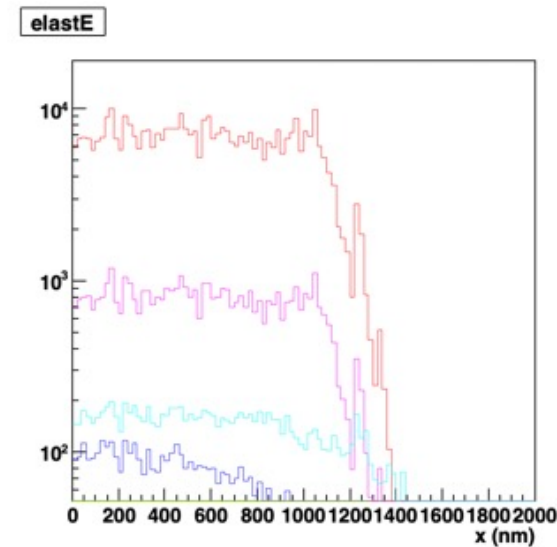
```
proton_G4DNAElastic: 21
proton_G4DNAExcitation: 22
proton_G4DNAIonisation: 23
proton_G4DNAChargeDecrease: 24
```

3D visualization of 2 proton tracks



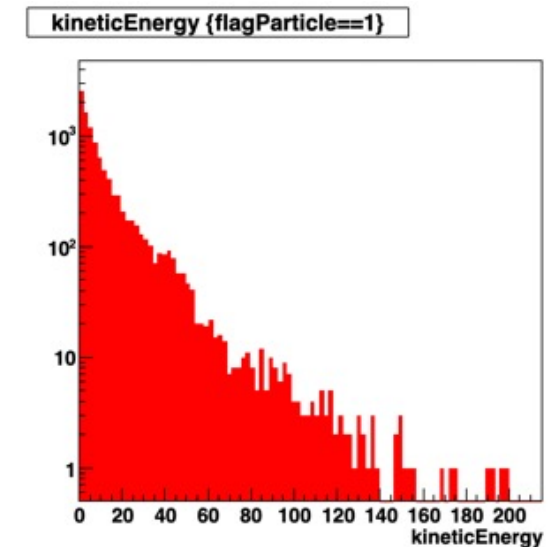
2 protons
(100 keV)

Occurring processes along x-axis



Elastic
Vib. excitation
Solvation
Ionisation
...

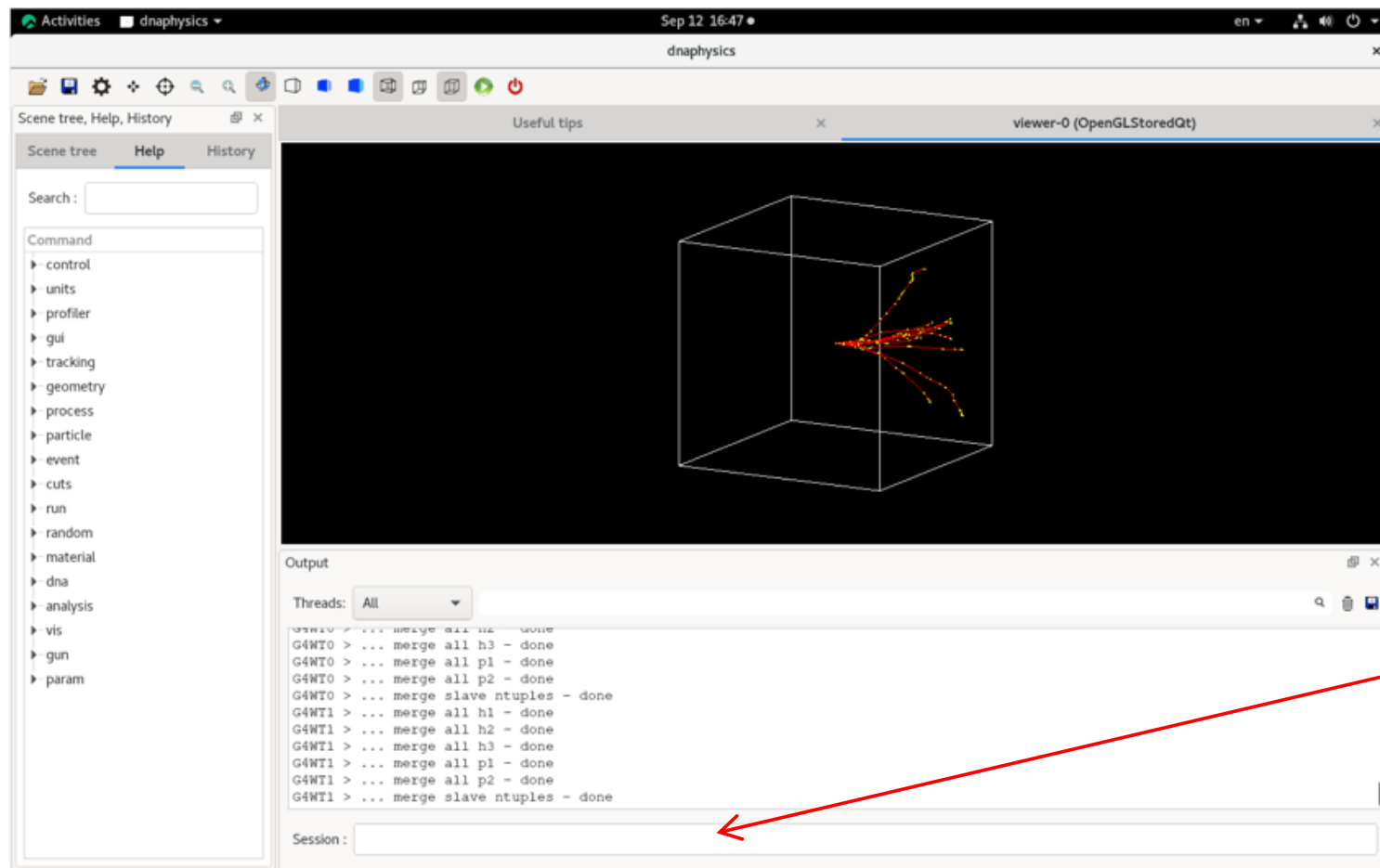
Energy distribution of electrons



Note: We also provide the `deexcitation.in` and `plotDeexcitation.C` macros to illustrate atomic deexcitation (fluorescence photons and Auger electrons)

Hands-on practice with the « dnaphysics » extended example

- Run dnaphysics in **interactive mode** (allows GUI commands)
`./dnaphysics`



You can enter your commands in the **GUI session**:

```
/gun/particle e-  
/gun/energy 100 keV  
/run/beamOn 10
```


Type of particles and physical processes (see [README](#))

Particles

-particles

```
gamma: 0
e-: 1
proton: 2
hydrogen: 3
alpha: 4
alpha+: 5
helium: 6
```

Processes

-processes

Capture: 1

```
e-_G4DNAElectronSolvation: 10
e-_G4DNAElastic: 11
e-_G4DNAExcitation: 12
e-_G4DNAIonisation: 13
e-_G4DNAAttachment: 14
e-_G4DNAVibExcitation: 15
msc: 110
CoulombScat: 120
eIoni: 130
```

```
proton_G4DNAElastic: 21
proton_G4DNAExcitation: 22
proton_G4DNAIonisation: 23
proton_G4DNAChargeDecrease: 24
msc: 210
CoulombScat: 220
hIoni: 230
nuclearStopping: 240
```

```
hydrogen_G4DNAElastic: 31
hydrogen_G4DNAExcitation: 32
hydrogen_G4DNAIonisation: 33
hydrogen_G4DNAChargeIncrease: 35
```

```
alpha_G4DNAElastic: 41
alpha_G4DNAExcitation: 42
alpha_G4DNAIonisation: 43
alpha_G4DNAChargeDecrease: 44
msc: 410
CoulombScat: 420
ionIoni: 430
nuclearStopping: 440
```

```
alpha+_G4DNAElastic: 51
alpha+_G4DNAExcitation: 52
alpha+_G4DNAIonisation: 53
alpha+_G4DNAChargeDecrease: 54
alpha+_G4DNAChargeIncrease: 55
msc: 510
CoulombScat: 520
hIoni: 530
nuclearStopping: 540
```

```
helium_G4DNAElastic: 61
helium_G4DNAExcitation: 62
helium_G4DNAIonisation: 63
helium_G4DNAChargeIncrease: 65
```

```
GenericIon_G4DNAIonisation: 73
msc: 710
CoulombSca: 720
ionIoni: 730
nuclearStopping: 740
```

```
phot: 81
compt: 82
conv: 83
Rayl: 84
```

Try to change the Physics constructor...

- Can be selected in `dnaphysics.in` ...

```
#  
# Physics  
# - To use Geant4-DNA constructor X, X=0, 2, 4, or 6 (recommended)  
#/dna/test/addPhysics DNA_Opt0  
/dna/test/addPhysics DNA_Opt2  
#/dna/test/addPhysics DNA_Opt4  
#/dna/test/addPhysics DNA_Opt6  
#
```

- Run the simulation and display ROOT results

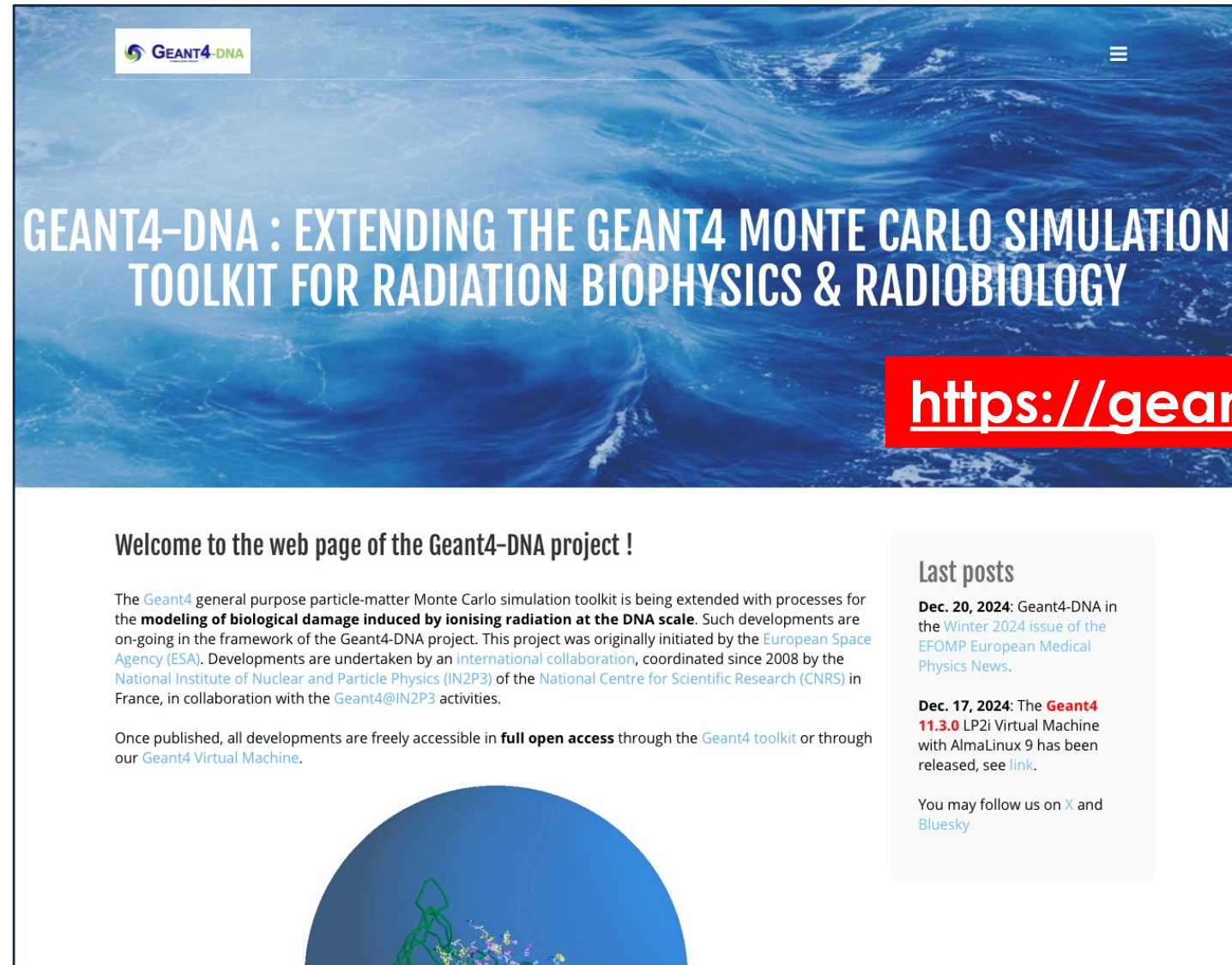
Try to change the density value of the target material

- Can be selected in [dnaphysics.in](#) ...

```
#  
# Material  
/dna/test/setMat G4_WATER  
# or alternatively  
#/dna/test/setMatDens G4_WATER_MODIFIED 1.200 g/cm3  
#
```

- Run the simulation and display ROOT results

Thank you for your attention



GEANT4-DNA : EXTENDING THE GEANT4 MONTE CARLO SIMULATION TOOLKIT FOR RADIATION BIOPHYSICS & RADIOBIOLOGY

<https://geant4-dna.org/>

Welcome to the web page of the Geant4-DNA project !

The [Geant4](#) general purpose particle-matter Monte Carlo simulation toolkit is being extended with processes for the **modeling of biological damage induced by ionising radiation at the DNA scale**. Such developments are on-going in the framework of the Geant4-DNA project. This project was originally initiated by the [European Space Agency \(ESA\)](#). Developments are undertaken by an [international collaboration](#), coordinated since 2008 by the [National Institute of Nuclear and Particle Physics \(IN2P3\)](#) of the [National Centre for Scientific Research \(CNRS\)](#) in France, in collaboration with the [Geant4@IN2P3](#) activities.

Once published, all developments are freely accessible in **full open access** through the [Geant4 toolkit](#) or through our [Geant4 Virtual Machine](#).

Last posts

Dec. 20, 2024: Geant4-DNA in the [Winter 2024 issue of the EFOMP European Medical Physics News](#).

Dec. 17, 2024: The [Geant4 11.3.0](#) LP2i Virtual Machine with AlmaLinux 9 has been released, see [link](#).

You may follow us on [X](#) and [Bluesky](#)