Geant4 Simulation of Physical Processes part 2: An extended example

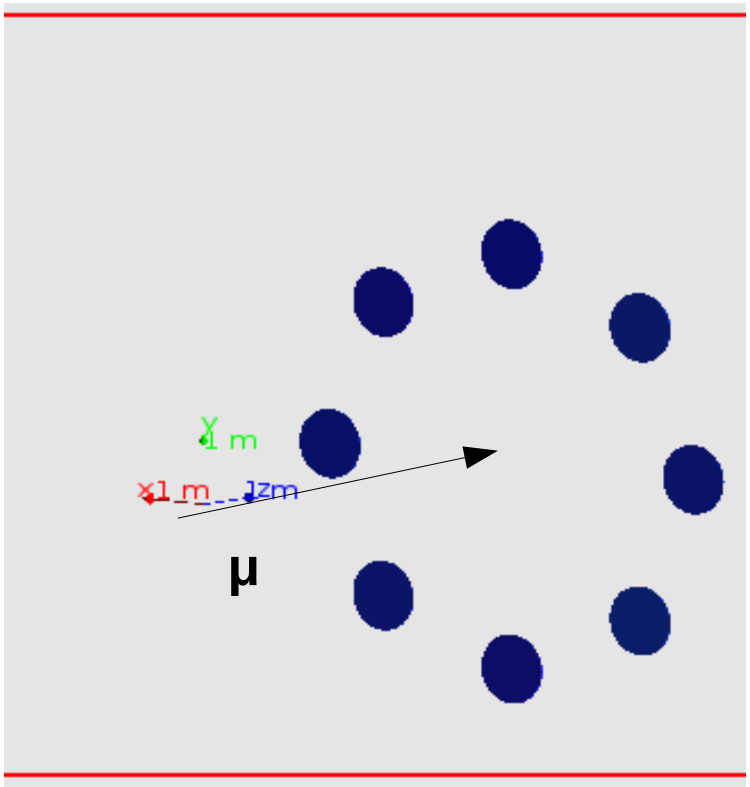
Алексей Жемчугов, Михаил Демичев
ОИЯИ
E-mail: zhemchugov@jinr.ru
mdemichev@jinr.ru

Большие Коты

4-12 июня 2015

An extended example

- SimpleHE.zip на http://geant4.jinr.ru
- Opltical photons included with their processes (Cherenkov light, scintillation, absorbtion, scattering)
- Geometry: water tank
- Inside water 8 disks - «sensitive detectors»
- Muon gun (100 MeV, orthogonal to detectors plane)
- If optical photon hits the «detector» ROOT histos are filled with detector and photon information. Other particles are ignored.
- At each muon step energy deposit is printed
- Either batch or graphical interface with visualisation



Simulation of detector response

- Every logical volume can be set up as a counting volume or so called «sensitive detector». While the particle crosses this volume on every step a dedicated user function is called for special treatment of such a case.

- There can be multiple sensitive volumes. They can treat particle crossings ach in its own way.

- Here you are free to further simulate signal digitisation and detector response:

Setting up the SensitiveDetector(I)

- User creates a class inherited from G4VSensitiveDetector
- User implements methods:
  - **Initialize**() is called by RunManager at the beginning of each event. Here we can clear out variables for counters and accumulators.
  - **ProcessHits**() is called at each step of the particle crossing the sensitive detector. Allows us to access all information about particle properties and its interaction with material. Here we can create hits.
  - **EndOfEvent**() is called at the end of event.

    A right place to either store it or reject the event.

Setting up the SensitiveDetector(II)

• Initialise (or just get pointer to) the manager of sensitive detectors G4SDManager

G4SDManager fSDM = G4SDManager::GetSDMpointer();*

• Register the pointer to object of user`s sensitive detector inside the manager

fSDM->AddNewDetector(G4VSensitiveDetector)*

• Associate logical volume with sensitive detector

logVolume->SetSensitiveDetector(G4VSensitiveDetector)*

How to get information from the G4Step*

```
G4int pdgcode =  step->GetTrack()->GetDefinition()->GetPDGEncoding();

G4double energy =  step->GetTrack()->GetKineticEnergy();

G4double time =  step->GetPostStepPoint()->GetGlobalTime();

G4int id = step->GetPreStepPoint()->GetTouchableHandle()-
>GetCopyNumber();

G4String vol_name = step->GetPreStepPoint()->GetTouchableHandle()-
>GetVolume()->GetName();

G4double edep = step->GetTotalEnergyDeposit();

G4double trackid = step->GetTrack()->GetTrackID();
```

User defined actions

- Классы-наследники G4RunAction, G4EventAction, G4UserTrackingAction, G4UserSteppingAction
- G4RunAction: для каждого сеанса (**BeginOfRun, EndOfRun**)
- G4EventAction: вызывются на каждом событии (**BeginOfEvent, EndOfEvent**)
- **G4UserTrackingAction**: для каждого трека (до и после трекинга частицы) вызываются методы соответственно:
    - ***virtual void PreUserTrackingAction(const G4Track*)***
    - ***virtual void PostUserTrackingAction(const G4Track*)***
- G4UserSteppingAction: на каждом шаге вызывается метод
    - ***virtual void UserSteppingAction(const G4Step*)***

User interface to G4 simulation

- Пакетный режим
- Пакетный режим, управляемый сценарием
- Интерактивный режим с командной строкой
- Интерактивный режим с графическим интерфейсом

Управление программой моделирования

- Geant4 имеет встроенный набор команд, управляющих моделированием
  - уровень диагностики
  - изменение геометрических параметров модели
  - изменение списка учитываемых процессов
  - ...
- Этот набор может быть расширен пользователем
- Команды могут использоваться в интерактивном режиме, считываться из файла сценария или быть запрограммированы в коде

G4 commands

/control	управление интерфейсом
/units	система единиц
/geometry	навигация и проверка перекрытий объемов
/tracking	управление объектами TrackingManager и SteppingManager
/event	управление объектом EventManager
/cuts	управление порогами рождения частиц
/run	управление сеансом
/random	управление генератором случайных чисел
/material	просмотр списка и свойств материалов
/particle	изменение списков и свойств частиц
/process	изменение списка активных процессов
/vis	визуализация
/gun	управление генератором первичной вершины
/hits	управление детектирующими объемами
/score	работа со счетчиками

/control/execute

Выполнение сценария:

/control/execute имя_файла_сценария

Например

/control/execute ~/run.mac

/tracking/verbose

/tracking/verbose level

Verbosity level — amount of messages in the output

0 : Quiet — no messages

1 : Minimal information for each step

2 : Level1 + secondaries informatiom

3 : Extra information about the state in the beginning and the end of the step.

4 : 3+ info about the processes

5 : 4+ processes report their expected step length for the current step.

/run/beamOn

Start the simulation.

Example:

/run/beamOn 1000 — start simulating a run of 100 events

/run/beamOn 1000 event.mac 100

Start simulating a run of 1000 events. During the run after each 100 events start script «event.mac»

/random

Management of the initial state of random numbers generator

/random/setDirectoryName [fileName]

/random/setSavingFlag [flag]

/random/saveThisRun

/random/saveThisEvent

/random/resetEngineFrom [fileName]

The most important commands

Idle> help
Command directory path : /
 Sub-directories :
 1) /control/ UI control commands.
 2) /units/ Available units.
.....
Type the number (0:end, -n:n level back) :
0
Exit from HELP.

Idle>exit

Saving your simulation. Geant4 to ROOT.

•**Geant4 не имеет встроенных средств сохранения результатов на диск.**

•Пользователь может использовать любые способы сохранения результатов моделирования, используя методы Geant4 для доступа к этим результатам

- ASCII файл
- ROOT
- JAS
- HBOOK
- g4tools (AIDA)
- .....

Simple usage of ROOT

Simple case — using ROOT methods in sensitive detector class

- SD Constructor - Initialise trees and histograms
- Fill the trees and histograms inside ProcessHits() and/or EndOfEvent() methods
- SD Destructor — Save trees and histograms into ROOT file

Using ROOT in more elaborate way (I)

Create a singleton class MyROOTManager in main():

```
class MyROOTManager
{
public:
static MyROOTManager* GetPointer()
{
    if (theInstance == 0) theInstance = new  MyROOTManager();
    return theInstance;
}
private:
MyROOTManager(){}; // создание объекта только методом GetPointer()
static MyROOTManager* theInstance;
};

MyROOTManager::theInstance = 0;
```

It is accessible from everywhere,
just include the header and get pointer!

Using ROOT in more elaborate way (II)

- Class MyROOTManager stores all the trees and histograms and also manages writing into TFile

- In every place of your simulation code MyROOTManager can be accessed by calling

```
MyROOTManager* fRM = MyROOTManager::GetPointer();
```

- Then you fill your histograms through fRM pointer.

Compiling Geant4 and ROOT: fixes in the CMakeLists.txt

```
....
project(SimpleHE)
set(CMAKE_MODULE_PATH
      ${Geant4_DIR}/Modules/
      ${CMAKE_MODULE_PATH})
option(WITH_GEANT4_UIVIS "Build example with Geant4 UI and Vis drivers" ON)
if(WITH_GEANT4_UIVIS)
  find_package(Geant4 REQUIRED ui_all vis_all)
else()
  find_package(Geant4 REQUIRED)
endif()
#include (FindROOT.cmake)
find_package(ROOT REQUIRED)
include(${Geant4_USE_FILE})
include_directories(${PROJECT_SOURCE_DIR}/include)
#include(FindROOT.cmake)
set(INCLUDE_DIRECTORIES   ${ROOT_INCLUDE_DIR} )
include_directories( ${INCLUDE_DIRECTORIES})
set(LINK_DIRECTORIES      ${ROOT_LIBRARY_DIR}  )
file(GLOB sources ${PROJECT_SOURCE_DIR}/src/*.cc)
file(GLOB headers ${PROJECT_SOURCE_DIR}/include/*.hh)
add_executable(simpleHE.exe simpleHE.cc ${sources} ${headers})
target_link_libraries(simpleHE.exe ${Geant4_LIBRARIES} ${ROOT_LIBRARIES})
....
```

Optical photons

Регистрация конструктора оптических процессов в дополнение к стандартному набору физ. процессов:

```
// Physics list
G4VModularPhysicsList* physicsList = new QGSP_BERT;
physicsList->SetVerboseLevel(1);
physicsList->RegisterPhysics(new G4OpticalPhysics);
runManager->SetUserInitialization(physicsList);
```

Optical properties of materials

- В материалах, оптические свойства которых не заданы, оптические фотоны не моделируются
- Необходимо создать объект G4MaterialPropertiesTable, заполнить в нем таблицы оптических свойств и передать его в соотв. объект класса G4Material

```
G4MaterialPropertiesTable* MPT1 = new G4MaterialPropertiesTable();

Water->SetMaterialPropertiesTable(MPT1);
```

- Основные таблицы оптических свойств:
  - RINDEX - показатель преломления (в зависимости от энергии фотона)
  - ABSLENGTH - Длина поглощения (в зависимости от энергии фотона)
- Дополнительные свойства для сцинтилляции, Ми-рассеяния.

Have fun with Example1 and Example2! :-)

1. Using Ex1 as a starting point add graphical interface and visualisation to it. Hint: have a look at Ex2.

2. Add calculation of total energy deposit for the brick of water in Ex1.

3. Add saving energy deposit to a ROOT histogram

in Ex1.