COMPUTING METHODS IN HIGH ENERGY PHYSICS

S. Lehti

Helsinki Institute of Physics

Spring 2024

Detector simulations

Introduction

In high energy physics, the method to simulate experimental data is done in two stages, event generation and detector simulation. Event generators describe the particle reactions and produce the momentum vectors of the generated particles. The output of an event generator is used as input for a detector simulation program.

The response of a detector to the passage of the scattered particles involves random processes such as ionization, multiple Coulomb scattering etc. Therefore the transport of particles through an experimental setup can be effectively implemented using the Monte Carlo technique. Programming packages, such as GEANT can be used to describe complicated detector configurations. Many GEANT based simulation packages has been developed to offer central facilities for Monte Carlo studies for different experiments.

GEANT4

Geant4 is a free software package composed of tools which can be used to accurately simulate the passage of particles through matter. The simulation processes included in the toolkit are

- geometry of the system
- materials involved
- particles

Detector simulations

- generation of events
- tracking through matter and EM fields
- response of sensitive detector components
- generation of event data
- storage of events and tracks
- visualization
- analysis of data

Users may construct stand-alone applications, or applications built upon another object-oriented framework.

The Geant4 physics models handle the interactions of particles with matter across a very wide energy range. It incorporates a large part of all that is known about particle interactions.

The Geant4 web page url is
http://geant4.cern.ch

For the detector simulation one needs a detector description. The detector definition requires the representation of its geometrical elements, their materials and electronics properties. The geometrical representation of detector elements focuses on the definition of solid models and their spatial position.

Geant4 is capable of describing and propagating in variety of electromagnetic fields. Magnetic

Detector simulations

fields, electric fields and electromagnetic uniform and non-uniform fields can be specified. In order to propagate a particle inside a field, the equation of motion of the particle in the field is integrated in the toolkit.

A physical interaction of a particle in the sensitive region of a detector is called a hit. Hits simulate the physical signal in the detector element produced by the particle-matter interaction.

The physical signal in the sensitive detector element, the detector response, is transformed with the help of electronics into a digital signal, which can be stored on tape or hard drive. The simulation of this step is called digitization, and the digitized data, the raw data ("digis"). Typical usages of the digitizer are

▶ Simulation of ADC and/or TDC

▶ Simulation of readout scheme

▶ Generation of raw data

▶ Simulation of trigger logics

▶ Simulation of pile-up

At high luminosities several separate events are likely to occur at the same bunch crossing. These so called minimum bias events or pile-up add a non-negligible level of background to the response of the detector system. In the tracker the pile-up events generate a number of hits which can affect the track finding efficiency and fit accuracy.

Detector simulations

In the calorimeters these events add pile-up noise affecting calorimeter and missing energy resolution.

Physics processes describe how particles interact with the material. Seven major categories are provided by Geant4:

- electromagnetic
- hadronic
- decay
- photolepton-hadron
- optical
- parametrization
- transportation

Electromagnetic physics processes include

- Photon processes, like Compton scattering, gamma conversion, photo-electric effect and muon pair production
- Electron/positron processes, like ionization and delta rays, bremsstrahlung and positron annihilation
- Muon processes, like ionization and delta rays, bremsstrahlung and e+e- pair production
- Hadron/ion processes, like ionization
- Multiple scattering
- Low energy electromagnetic processes, like Rayleigh scattering

Detector simulations

Hadronic interactions include

- lepton-hadron interactions
- photonuclear and electronuclear reactions
- nucleus-nucleus reactions
- elastic scattering
- interactions of stopping particles
- nuclear cascades
- fission, evaporation, break-up models
- low energy neutron interactions
- radioactive decay

The optical effects include processes like Cerenkov radiation, scintillation, wavelenght shifting and absorption.

Probability of having an interaction within L due to physics process i is

$$p_i(L) = 1 - exp(-L/\lambda_i) \quad (1)$$

($\lambda_i =$ mean free path). The Monte Carlo algoritm:

1. Set properties for incident particle (momentum..)
2. Get values for λ_i for all relevant processes
3. Sample the free path length L from the distributions of all participating processes
4. Select the smallest path length
5. Move the particle by this step
6. Simulate the interaction
7. if particle still exists, goto 1

Detector simulations

Free path lengths sampling produces steps.

A step is always created at the material boundaries.

Physics processes limit the step size according to the sample's free path length.

E.g. zick-zack path due to elastic Coulomb scattering.

E.g. Energy loss and possible creation of other particles

The physics process is a microscopic description of the particle interaction with another particle of the material or the external field, accoring to quantum theory.

The Detector Description must be as realistic as possible for the physics to be studied. Particles tracked through the detector must see and feel a realistic environment in order to deliver realistic simulation results.

The Detector Description need lots of experience and expertise, one has to know which details of the detector are essential and which can be safely ignored. Often the detector cannot be modelled 1:1, e.g. the geometry must fit into the computer memory..

Usage of Geant4

Since the contents of the program varies according to the needs of a

Detector simulations

given simulation, the Geant4 toolkit does not provide the main program. The main must be written by the user. Examples of simple main programs are, however, provided. The main() method is implemented by two toolkit classes

- ▶ G4RunManager
- ▶ G4UImanager

The first thing main() must do is to create an instance of the G4RunManager class. It controls the flow of the program and manages the event loops within a run. The run manager is responsible for initialization procedures, and it must be given all the information necessary to build and run the simulation. This includes

- ▶ detector description
- ▶ physics processes
- ▶ how the primary particles are produced
- ▶ any additional requirements

Example 1:

```
G4RunManager* runManager = new
G4RunManager;
runManager->SetUserInitialization(new
      ExN01DetectorConstruction);
runManager->SetUserInitialization(new
      ExN01PhysicsList);

runManager->SetUserAction(new
      ExN01PrimaryGeneratorAction);

runManager->initialize();
```

Detector simulations

```
G4UImanager* UI =
G4UImanager::GetUIpointer();
UI->ApplyCommand("/run/verbose
1");
UI-
>ApplyCommand("/event/verbose
1");
UI-
>ApplyCommand("/tracking/verbose
1");

int numberOfEvents = 3;
runManager-
>BeamOn(numberOfEvents);


delete runManager;
```

The detector description specifies the detector geometry, used materials, defines sensitive regions and readout schemes.

The geometry defines the size and shape, like Boxes, Tubes and their sections, Cones and their sections, Spheres, Wedges, and Toruses.

```
G4Tubs(const G4String& pName,
        G4double pRMin,
        G4double pRMax,
        G4double pDz,
        G4double pSPhi,
        G4double pDPhi)
```

```
G4Material* Al = new
G4Material("Aluminum",
z= 13., a= 26.98*g/mole, density=
2.7*g/cm3);
```

Every quantity such as mass, energy, density,.. is represented by

Detector simulations

multiplying the numerical value with the unit. For reading returned quantities, the numerical value should be divided by the unit:
double speed = getSpeed()/(km/h);
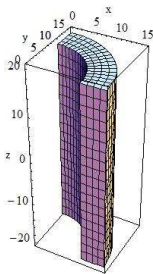
Figure: pRMin = 10, pRMax = 15, pDz = 20 pSPhi = 0*Degree, pDPhi = 90*Degree

The Logical Volume manages the information associated with detector elements represented by a given Solid and Material, independently from its physical position in the detector.

Physical volumes represent the spatial positioning of the volumes describing the detector elements. Several techniques can be used. They range from the simple placement of a single copy to the repeated positioning using either a simple linear formula or a user specified function.

SensitiveDetector represents the parts of the detector that will do the actual measurement. Sensitivity is handled via logical volumes, and the

Detector simulations

principal mandate of a sensitive detector is the construction of hit objects using information from steps along a particle track.

Readout geometry is a virtual, parallel geometry for obtaining the channel number. It associates ReadoutGeometry object with a sensitive detector.

The primary particle production can be done with particle guns, or one can use interface to external event generators. For Geant4 it was decided not to link any FORTRAN code in the program. Instead, Geant4 provides an ASCII file interface for such event generators.

In the example 1 the PhysicsList

defines a geantino, and assigns transportation process. The PrimaryGeneratorAction constructs a particle gun and generates events by shooting geantinos with the gun. Geantino is a non-physical test particle which simulates particle transportation, it does not interact with material.

Geant4 is designed to transport long lived particles along macroscopic distances. We have to tell Geant4 the initial conditions of the particles we want to have simulated. The short lived particles like Z will not be tracked.

User has to instantiate the Primary Vertex.

Detector simulations

The G4UImanager is used for giving optional UI (or GUI) commands via the member function ApplyCommand. The commands are listed in the user guide (for app.dev) section 7.1. The command

/run/verbose 1

is selected to display main topics, the command

/event/verbose 1

to give stacking information, and command

/tracking/verbose 1

gives minimum information of each Step. These all have default values 0, which give silent operation.

Summary:

The various stages during a simulation run are

▶ Initialization
  - detector description
  - physics processes selection and configuration
▶ Run
  - contains several Events under the same conditions
▶ Event
  - generation of primary particles
  - tracking of all particles

We do not go deeper into the usage of Geant4, as normally one does not have to use Geant directly, but via an auxiliary program where the geometry etc is already defined and the end user only provides the primary event for the detector

Detector simulations

simulation to get a hit collection. Very often experiments do not use Geant4 digitization, but implement their own digitization machineries, like in CMS.

For further reading, Geant4 provides extensive documentation like

- ► User's guide for application developer (basic usage of Geant4)
- ► Physics reference manual
- ► Software reference manual

and most of all working examples for beginners, extended examples and examples for advanced users.