Monte Carlo programs

Alexander Khanov

PHYS6260: Experimental Methods is HEP
Oklahoma State University

November 15, 2017
Simulation steps: event generator

- **Input** = “data cards” (program options)
  - this is the beginning of the chain, so no real input

- **Output** = event records (list of particles and “vertices” – points where particles decayed into other particles)

- If produced in a standard form (e.g. HEPMC), the result of this step can be moved between different experiments
  - ex.: ATLAS passes to CMS the top pair production MC event records to facilitate combination of top mass measurements
  - ex.: the same samples can be probed with different future detector models to see which one has the best sensitivity to new physics

- Sometimes there is no real generator – use “particle gun”
  - particle gun = an MC generator which produces single particles with random momentum and direction
Simulation steps: particle propagation

- Input = event records
- Output = hits
  - this step is usually done by a GEANT based program
  - GEANT takes care of “swimming” in magnetic field, producing secondaries, and recording energy deposits
- Particles lose (“deposit”) their energy everywhere, but we are interested only in energy deposits in “sensitive volumes” (detector elements)
  - A hit is characterized by position of entry into the sensitive volume (can be inside the volume if the particle was created there), exit out of the sensitive volume (or the point where the particle decayed or went below the energy cut-off), and total energy deposited inside the volume
- Sometimes there is no real material simulation – use “geantino”
  - geantino = an artificial GEANT particle which is not involved in any material interactions, usually used to verify geometry
Simulation steps: digitization

- **Input** = hits
- **Output** = digits
  - this step is highly detector specific
  - for each hit, we need to determine the list of detector elements which are sensitive to this hit and convert the energy deposited in the sensitive volume to the signals similar to those recorded in the real experiment
  - ex.: in the silicon tracking detector, we convert the deposited energy to equivalent electric charge, determine which strips see the charge (taking into account Lorentz shifts etc.), find the fraction of charge deposited at each strips, and then simulate the operation of analog-to-digital convertor

- After digitization, the data obtained in a real experiment and those produced by MC follow the same reconstruction chain and can be directly compared at each step
Simulation steps: clusterization

- Input = digits
- Output = clusters
- Information from nearby digital signals is combined to create spatial points which determine position of original particles that crossed the detector
  - clusters are similar to hits (they are also characterized by position and deposited energy) but these characteristics are results of a measurement and have their uncertainties
- It is possible to skip the previous step and convert hits to clusters directly using parameterized detector response – the procedure known as “smearing”
Simulation steps: reconstruction

- Input = clusters
- Output = physics objects
  - Examples of physics objects: trigger objects, muons, electrons, \( \tau \)-leptons, jets, missing transverse energy

This step is the end of the simulation chain, the physics analysis takes it from here
  - the objects made at this stage are “raw” and have to be “cooked” (fixed) through a procedure known as calibration
GEANT (GEometry ANd Tracking)

- GEANT components (from http://geant4.cern.ch/):
  - the geometry of the system,
  - the materials involved,
  - the fundamental particles of interest,
  - the generation of primary events,
  - the tracking of particles through materials and electromagnetic fields,
  - the physics processes governing particle interactions,
  - the response of sensitive detector components,
  - the generation of event data,
  - the storage of events and tracks,
  - the visualization of the detector and particle trajectories, and
  - the capture and analysis of simulation data at different levels of detail and refinement.

- Created in 1950’s, originally written in fortran
  - used CERN developed packages IOPACK, ZEBRA, PAW

- geant4 (C++ implementation of GEANT using OO approach) started in 1994 (RD44), first production release in 1998

- De facto standard for all modern HEP experiments
GEANT: technicalities

- GEANT can be downloaded and built quite easily, it’s a standalone program that in its minimum version does not require anything else (except `cmake`)
  - it’s easy to run but not so easy to get something useful out of it
- For complex detectors, it may take months to build complete detector description
- GEANT experts are special people, don’t question their ways
- A minimum GEANT configuration requires three classes (PrimaryGenerator, Geometry, Physics) and a `main` to run
- To run GEANT, one can hardcode the commands in `main`, run it with commands from the command prompt, run it with commands written in a macro, or use GUI
  - GEANT experts don’t use GUI
- You will need to define your materials, detector geometry, describe the way you get your data in, and physics which you want to simulate
GEANT hits

- Sensitive volume: part of the detector where deposited energy gets recorded
- Only sensitive volumes can hold hits
- `geant4` provides an alternative to sensitive volumes called scoring
  - can’t tell you the details
ATLAS GEANT simulation