Experimental Techniques in Particle Physics

Geant4: Particle Sources

Andreas Nowack
nowack@physik.rwth-aachen.de
RWTH Aachen University

WS 2020/21

Quick Intro to Geant 4

Actions in Geant4

User classes

Initialization classes

Invoked at the initialization

G4VUserDetectorConstruction → Material and Geometry

G4VUserPhysicsList → Particles and Processes

Action classes

Invoked during an event loop

G4VUserPrimaryGeneratorAction → Primary Particles

G4UserRunAction

G4UserEventAction

G4UserSteppingAction

} Topic of Last Lesson

G4UserStackingAction

G4UserTrackingAction

main()

Geant4 does not provide main().

Note: classes written in Red are mandatory.

G4VUserPrimaryGeneratorAction

minimalistic primary generator action using G4ParticleGun:

```
DetectorPhysPrimaryGeneratorAction::DetectorPhysPrimaryGeneratorAction(
  DetectorPhysDetectorConstruction* DetectorPhysDC) : DetectorPhysDetector(DetectorPhysDC) {
  G4int n_particle = 1;
  particleGun  = new G4ParticleGun(n_particle);
  // default particle
  G4ParticleTable* particleTable = G4ParticleTable::GetParticleTable();
  G4String particleName;
  G4ParticleDefinition* particle = particleTable->FindParticle(particleName="gamma");
  // gun settings
  particleGun->SetParticleDefinition(particle);
  particleGun->SetParticleMomentumDirection(G4ThreeVector(0.,0.,1.));
  particleGun->SetParticleEnergy(10.*MeV);
  particleGun->SetParticlePosition(G4ThreeVector(0*cm,0.*cm,100.*cm));
}
```

constructor:
initialize particle source

```
DetectorPhysPrimaryGeneratorAction::~DetectorPhysPrimaryGeneratorAction() {
  delete particleGun;
}
```

destructor

```
void DetectorPhysPrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent) {
  particleGun->GeneratePrimaryVertex(anEvent);
}
```

start of each event

Particle Gun

- Properties of G4ParticleGun can be modified event by event. Use the following methods in GeneratePrimaries():

```
void SetParticleDefinition(G4ParticleDefinition*)
void SetParticleMomentum(G4ParticleMomentum)
void SetParticleMomentumDirection(G4ThreeVector)
void SetParticleEnergy(G4double)
void SetParticleTime(G4double)
void SetParticlePosition(G4ThreeVector)
void SetParticlePolarization(G4ThreeVector)
void SetNumberOfParticles(G4int)
```

- GeneratePrimaryVertex() can be invoked more than once to generate additional particle tracks. Different particle properties are possible.
- More than one G4ParicleGun can be used in the primary generator action.

- Complex particle sources are possible.

Particle Gun

- user interface commands:

Command	Arguments	Description and Restrictions
/gun/List		List available incident particles
/gun/particle	name	Defines the particle type [default *geantino*], using Geant4 naming convention.
/gun/direction	ex ey ez	Set the momentum direction of generated particles. Does not need to be a unit vector.
/gun/energy	E unit	Sets the energy. The units can be eV, keV, MeV, GeV, TeV or PeV.
/gun/position	X Y Z unit	Sets starting position of the particle. The units can be micron, mm, cm, m or km.
/gun/ion	Z A Q E	After /gun/particle ion, sets the properties (atomic number Z, atomic mass A, ionic charge Q, excitation energy E in keV) of the ion.
/gun/ionLvl	Z A Q lvl	After /gun/particle ion, sets the properties (atomic number Z, atomic mass A, ionic charge Q, Number of metastable state excitation level (0-9) of the ion.
/gun/time	t0 unit	Sets the primary particle (event) time [default 0 ns]. The units can be ps, ns, us, ms, or s.
/gun/polarization	Px Py Pz	Sets the polarization vector of the source, which does not need to be a unit vector.
/gun/number	N	Sets the number of particles [default 1] to simulate on each event.

Other Particle Generators

- G4GeneralParticleSource
  - it is used the same way as G4ParticleGun (globally replace G4ParticleGun with G4GeneralParticleSource)
  - configuration via methods and via command line/macro
  - specification of spectral, spatial, and angular distributions of the primary source particles
    - spectrum: mono-energetic, linear, exponential, power-law, Gaussian, bremsstrahlung, blackbody, cosmic diffuse gamma ray, or piece-wise fits to data
    - spatial sampling:
      - point source
      - planar sources: circles, annuli, ellipses, squares, or rectangles
      - 1D or 2D beam spots
      - surface or volume sources: sphere, ellipsoid, cylinder, or parallelepipedon
    - angular distribution: unidirectional, isotropic, cosine-law, beam, or arbitrary (user defined)
    - multiple sources: multiple independent sources can be used in the same run

- G4HEPEvtInterface
  - almost all HEP (High Energy Physics) event generators can store the generated events in ASCII files
  - Geant4 can read these ASCII files and can produce G4PrimaryParticle objects associated with a G4PrimaryVertex object.

General Particle Source:
Basic Commands equivalent to Particle Gun

Command	Arguments	Description and Restrictions
/gps/List		List available incident particles
/gps/particle	name	Defines the particle type [default *geantino*], using Geant4 naming convention.
/gps/direction	Px Py Pz	Set the momentum direction [default (1,0,0)] of generated particles using direction cosines.
/gps/energy	E unit	Sets the energy [default 1 MeV] for mono-energetic sources. The units can be eV, keV, MeV, GeV, TeV or PeV. (NB: it is recommended to use /gps/ene/mono instead.)
/gps/position	X Y Z unit	Sets the centre co-ordinates (X,Y,Z) of the source [default (0,0,0) cm]. The units can be micron, mm, cm, m or km. (NB: it is reccomended to use /gps/pos/centre instead.)
/gps/ion	Z A Q E	After /gps/particle ion, sets the properties (atomic number Z, atomic mass A, ionic charge Q, excitation energy E in keV) of the ion.
/gps/ionLvl	Z A Q lvl	After /gps/particle ion, sets the properties (atomic number Z, atomic mass A, ionic charge Q, Number of metastable state excitation level (0-9) of the ion.
/gps/time	t0 unit	Sets the primary particle (event) time [default 0 ns]. The units can be ps, ns, us, ms, or s.
/gps/polarization	Px Py Pz	Sets the polarization vector of the source, which does not need to be a unit vector.
/gps/number	N	Sets the number of particles [default 1] to simulate on each event.
/gps/verbose	level	Control the amount of information printed out by the GPS code. Larger values produce more detailed output.

General Particle Source: Source Position and Shape

Command	Arguments	Description and Restrictions
/gps/pos/type	dist	Sets the source positional distribution type: *Point* [default], *Plane, Beam, Surface, Volume*.
/gps/pos/shape	shape	Sets the source shape type, after /gps/pos/type has been used. For a Plane this can be *Circle, Annulus, Ellipse, Square, Rectangle*. For both Surface or Volume sources this can be *Sphere, Ellipsoid, Cylinder, Para* (parallelpiped).
/gps/pos/centre	X Y Z unit	Sets the centre co-ordinates (X,Y,Z) of the source [default (0,0,0) cm].
/gps/pos/rot1	$R1_x$ $R1_y$ $R1_z$	Defines the first (x' direction) vector R1 [default (1,0,0)], which does not need to be a unit vector, and is used together with /gps/pos/rot2 to create the rotation matrix of the shape defined with /gps/shape.
/gps/pos/rot2	$R2_x$ $R2_y$ $R2_z$	Defines the second vector R2 in the xy plane [default (0,1,0)], which does not need to be a unit vector, and is used tohgether with /gps/pos/rot1 to create the rotation matrix of the shape defined with /gps/shape.
/gps/pos/halfx	len unit	Sets the half-length in x [default 0 cm] of the source.
/gps/pos/halfy	len unit	Sets the half-length in y [default 0 cm] of the source.
/gps/pos/halfz	len unit	Sets the half-length in z [default 0 cm] of the source.
/gps/pos/radius	len unit	Sets the radius [default 0 cm] of the source or the outer radius for annuli.
/gps/pos/inner_radius	len unit	Sets the inner radius [default 0 cm] for annuli.
/gps/pos/sigma_r	sigma unit	Sets the transverse (radial) standard deviation [default 0 cm] of beam position profile.
/gps/pos/sigma_x	sigma unit	Sets the standard deviation [default 0 cm] of beam position profile in x-direction.
/gps/pos/sigma_y	sigma unit	Sets the standard deviation [default 0 cm] of beam position profile in y-direction.
/gps/pos/paralp	alpha unit	Used with a Parallelepiped. The angle [default 0 rad] α formed by the y-axis and the plane joining the centre of the faces parallel to the zx plane at y and +y. Units: deg or rad.
/gps/pos/parthe	theta unit	Used with a Parallelepiped. Polar angle [default 0 rad] θ of the line connecting the centre of the face at z to the centre of the face at +z. The units can only be deg or rad.
/gps/pos/parphi	phi unit	Used with a Parallelepiped. The azimuth angle [default 0 rad] φ of the line connecting the centre of the face at z with the centre of the face at +z. The units can only be deg or rad.
/gps/pos/confine	name	Allows the user to confine the source to the physical volume *name* [default NULL].

General Particle Source: Source Direction and Angular Distribution

Command	Arguments	Description and restrictions
/gps/ang/type	AngDis	Sets the angular distribution type (*iso* [default], *cos, planar, beam1d, beam2d, focused, user*) to either isotropic, cosine-law or user-defined.
/gps/ang/rot1	$AR1_x$ $AR1_y$ $AR1_z$	Defines the first (x' direction) rotation vector AR1 [default (1,0,0)] for the angular distribution and is not necessarily a unit vector. Used with /gps/ang/rot2 to compute the angular distribution rotation matrix.
/gps/ang/rot2	$AR2_x$ $AR2_y$ $AR2_z$	Defines the second rotation vector AR2 in the xy plane [default (0,1,0)] for the angular distribution, which does not necessarily have to be a unit vector. Used with /gps/ang/rot2 to compute the angular distribution rotation matrix.
/gps/ang/mintheta	MinTheta unit	Sets a minimum value [default 0 rad] for the θ distribution. Units: deg or rad.
/gps/ang/maxtheta	MaxTheta unit	Sets a maximum value [default π rad] for the θ distribution. Units: deg or rad.
/gps/ang/minphi	MinPhi unit	Sets a minimum value [default 0 rad] for the φ distribution. Units: deg or rad.
/gps/ang/maxphi	MaxPhi unit	Sets a maximum value [default 2π rad] for the φ distribution. Units: deg or rad.
/gps/ang/sigma_r	sigma unit	Sets the standard deviation [default 0 rad] of beam directional profile in radial. The units can only be deg or rad.
/gps/ang/sigma_x	sigma unit	Sets the standard deviation [default 0 rad] of beam directional profile in x-direction. The units can only be deg or rad.
/gps/ang/sigma_y	sigma unit	Sets the standard deviation [default 0 rad] of beam directional profile in y-direction. The units can only be deg or rad.
/gps/ang/focuspoint	X Y Z unit	Set the focusing point (X,Y,Z) for the beam [default (0,0,0) cm]. The units can only be micron, mm, cm, m or km.
/gps/ang/user_coor	bool	Calculate the angular distribution with respect to the user definded co-ordinate system (*true*), or with respect to the global co-ordinate system (*false*, default).
/gps/ang/surfnorm	bool	Allows user to choose whether angular distributions are w.r.t the co-ordinate system (*false*, default) or surface normals (*true*) for user-defined distributions.

General Particle Source: Energy Spectra

Command	Arguments	Description and Restrictions
/gps/ene/type	EnergyDis	Sets the energy distribution type to one of: *Mono* (mono-energetic, default), Lin (linear), Pow (power-law), Exp (exponential), Gauss (Gaussian), Brem (bremsstrahlung), Bbody (black-body), Cdg (cosmic diffuse gamma-ray), User (user-defined histogram), Arb (point-wise spectrum), Epn (energy-per-nucleon histogram)
/gps/ene/min	Emin unit	Sets the minimum [default 0 keV] for the energy distribution.
/gps/ene/max	Emax unit	Sets the maximum [default 0 keV] for the energy distribution.
/gps/ene/mono	E unit	Sets the energy [default 1 MeV] for mono-energetic sources.
/gps/ene/sigma	sigma unit	Sets the standard deviation [default 0 keV] in energy for Gaussian or Mono energy distributions.
/gps/ene/alpha	alpha	Sets the exponent α [default 0] for a power-law distribution.
/gps/ene/temp	T	Sets the temperature in kelvins [default 0] for black body and bremsstrahlung spectra.
/gps/ene/ezero	E0	Sets scale E_0 [default 0] for exponential distributions.
/gps/ene/gradient	gradient	Sets the gradient (slope) [default 0] for linear distributions.
/gps/ene/intercept	intercept	Sets the Y-intercept [default 0] for the linear distributions.
/gps/ene/biasAlpha	alpha	Sets the exponent α [default 0] for a biased power-law distribution. Bias weight is determined from the power-law probability distribution.
/gps/ene/calculate		Prepares integral PDFs for the interally-binned cosmic diffuse gamma ray (*Cdg*) and black body (*Bbody*) distributions.
/gps/ene/emspec	bool	Allows user to specify distributions are in momentum (*false*) or energy (*true*, default). Only valid for *User* and *Arb* distributions.
/gps/ene/diffspec	bool	Allows user to specify whether a point-wise spectrum is integral (*false*) or differential (*true*, default). The integral spectrum is only usable for *Arb* distributions.

General Particle Source: Example

- planar source
  - square, 4 cm by 4 cm
  - centred at (1,2,1) cm
  - by default the normal of this plane is the z-axis
- angular distribution follows the cosine-law.
- energy spectrum
  - linear
  - with gradient m and intercept I_0 equal to 1: $I \propto I_0 + m \times E$
  - from 2 to 10 MeV.
- 10,000 primaries to be generated.

```
/gps/verbose 2
/gps/particle gamma
/gps/pos/type Plane
/gps/pos/shape Square
/gps/pos/centre 1 2 1 cm
/gps/pos/halfx 2 cm
/gps/pos/halfy 2 cm
/gps/ang/type cos
/gps/ene/type Lin
/gps/ene/min 2 MeV
/gps/ene/max 10 MeV
/gps/ene/gradient 1
/gps/ene/intercept 1
/run/beamOn 10000
```

Comparison

- Particle Gun
  - simple and intuitive
  - shoot one track at a time (one can shoot more than one track within an event)
  - easy to handle
    - use set methods to change values (either track by track or event by event)
- General Particle Source
  - very powerful
  - controlled by user interface commands
    - almost impossible to control through set methods
  - capability of shooting particles from a surface or a volume
  - capability of randomizing kinetic energy, position, and/or direction following pre-defined or user-specified distributions
  - support of multiple independent sources with different intensities

- What shall I use?
  - If you need to shoot primary particles from a surface of a complicated volume, use GPS
  - If you need a complicated distribution, use GPS
  - Otherwise, use Particle Gun

Exercise: Particle Sources

1. Download DetectorPhys_T9.tar.gz and decompress it.
   - This code uses Geant4’s histogram functions and produces histograms automatically using the Run Action and the Tracking Action. Starting positions, energies, and directions of each track are stored in histograms.
   - Particle source produces geantinos (no interactions)
2. Compile the code, run it, and check the histograms in hist_T9.root:
   - start root: `root hist_T9.root` and open a browser: `new TBrowser()`
   - use the browser to view the histograms
   - exit root: `.q`
3. Modify GeneratePrimaries() so that the shape of the beam is a square (edge length 5 cm) in X-Y plane at Z = -10 cm and the particle’s direction is in Z direction.
   - Include Randomize.hh and use G4UniformRand() in order to get random numbers between 0 and 1.
4. Compile your code, run it, and check the histograms.
5. Make the kinetic energy of the particles uniformly distributed between 1 and 10 GeV.
6. Compile the code, run it, and check the histograms.
7. Make the direction of the particles uniformly distributed with an opening angle of 10°.
8. Compile the code, run it, and check the histograms.
9. Repeat the previous tasks using General Particle Source.

   http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/ForApplicationDeveloper/html/GettingStarted/generalParticleSource.html