Version 11.0

Fast Simulation

Anna Zaborowska

October 12, 2022

Outline

1. What is fast simulation?
2. How to use it in Geant4
   - where
   - what
   - how

   } to parametrise
3. Short summary
4. Examples

example

link to code in G4 v11.0.p03

```
...
```

What do we mean by fast simulation?

Fast simulation **is not** Geant4 simulation that ‘magically’ produces results in less time.

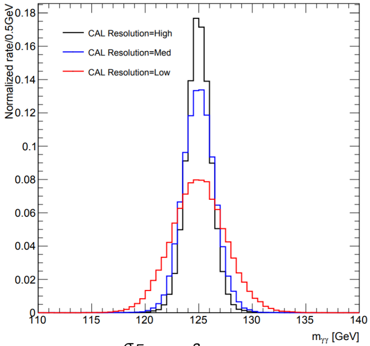
Fast simulation **is** a trade-off between simulation time and accuracy.

Time is gained by performing less operations in complicated region $\rightarrow$ no detailed physics of Geant4 $\rightarrow$ *parameterisation*.

Fast simulation hooks in Geant4 allow to overtake control over particles in certain regions – user decides what happens and when/if they return to the detailed simulation.

Why to use parameterisation / fast(er) simulation?

arXiv:1606.09408



$$\frac{\sigma_E}{E} = \frac{a}{\sqrt{E}} \oplus c$$

a=20%, c=2%

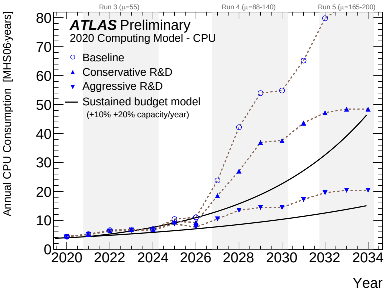
a=10%, c=1%

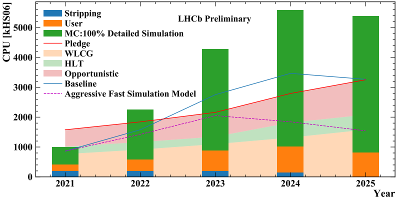
a=6%, c=0.7%

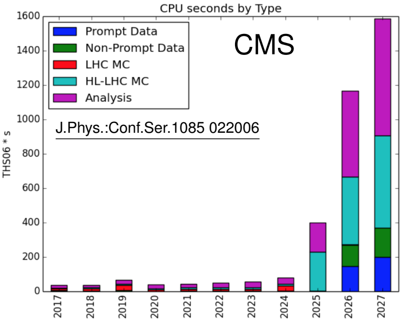
- Physics studies that assume certain detector performance
  - to speed-up simulation of already known detector (e.g. extracted efficiency/resolution from detailed simulation)
  - to study impact of detector performance on physics observables (example in the plot: calorimeter resolution on Higgs mass)
  - to study in detail only one detector, with others parametrised (e.g. parametrised tracker in front of in-detail calorimeter)

Why to use parameterisation / fast(er) simulation?

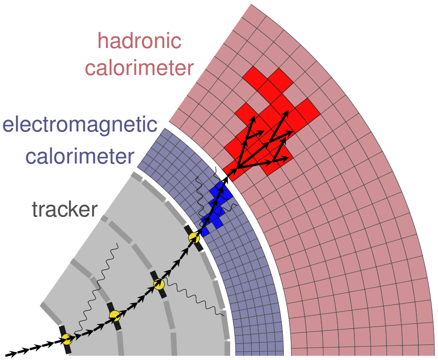
- To speed-up simulation in order to generate more data within same CPU time
  - to match available computing resources;
  - to provide sufficient amount of simulation data for comparison with the experimental data;

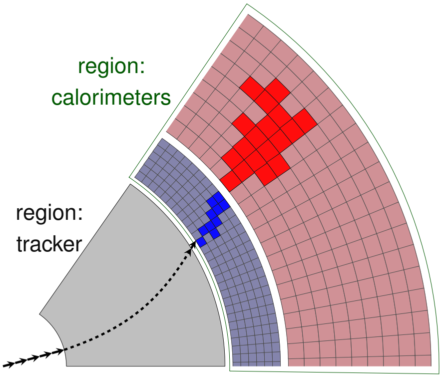






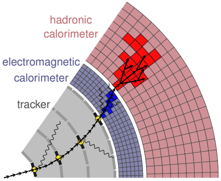
How we simulate particles?





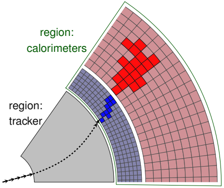
Fast simulation is a shortcut to the standard tracking and detailed simulation.

Simulation of particle passage



detailed / "full" simulation

parameterisation / "fast" simulation



- detailed detector description
- definitions of particles and processes
- transport in e-m field
- Geant4 a standard toolkit

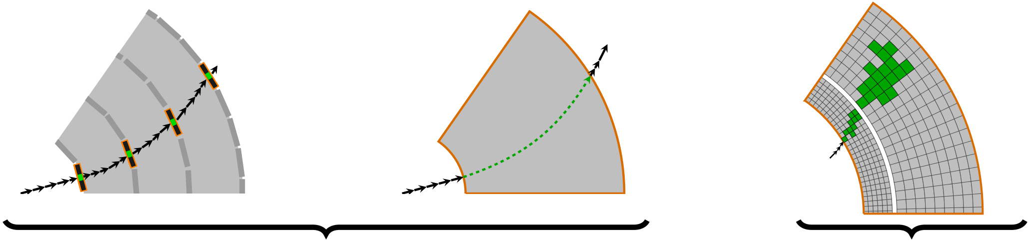
- where particles are parametrised
- which particles
- how/what happens
- detector / use-case dependent

Defining both 'full' and 'fast' simulation within one framework (Geant4) offers great flexibility to seamlessly mix both types.

Where?

Parameterisation may be realised within:

sub-volume (many volumes) | detector envelope (single volume) | assembly of volumes (non-physical volume)



"mass" geometry

parallel geometry

Fast simulation in Geant4 is attached to **G4Region**
(associated to root G4LogicalVolume in either mass or parallel geometry).

G4Region (envelope)

G4Region attached to root G4LogicalVolume is shared with daughters (and further ancestors).

```
G4Region(const G4String&) // constructor also registers to G4RegionStore
void G4Region::AddRootLogicalVolume (G4LogicalVolume*) // attach root volume to region
```

for mass geometry:

examples/extended/parameterisations/Par01/src/Par01DetectorConstruction.cc

```
G4Region* caloRegion = new G4Region("EM_calo_region");
caloRegion->AddRootLogicalVolume(calorimeterLog); // calorimeterLog is a G4LogicalVolume
```

for parallel geometry:

examples/extended/parameterisations/Par01/src/Par01ParallelWorldForPion.cc

```
  G4Region* ghostRegion = new G4Region("GhostCalorimeterRegion");
// ghostLogical is a G4LogicalVolume in parallel geometry, a box made of air encompassing both EM&H calorimeters
ghostRegion->AddRootLogicalVolume(ghostLogical);
```

Which particles?

Parameterisation is usually specified for selected particles.

In Geant4 parameterisation is defined as an additional process that (selected) particle may undergo – **G4FastSimulationManagerProcess**.

It will invoke parameterisation (if all conditions met):

- ☐ within selected volumes;
- ☐ for selected particle types;
- ☐ if trigger is issued (related to particle's properties or position within volume).

G4FastSimulationPhysics helps to add parameterisation process on top of any other physics list (which is used where parameterisation is not invoked).

(since v10.3, for older versions consult user's guide or slide 29)

G4FastSimulationPhysics

for mass and parallel geometry:

examples/extended/parameterisations/Par01/examplePar01.cc

```
FTFP_BERT* physicsList = new FTFP_BERT; // G4VModularPhysicsList
G4FastSimulationPhysics* fastSimulationPhysics = new G4FastSimulationPhysics(); // helper
fastSimulationPhysics->BeVerbose();
// - activation of fast simulation for particles having fast simulation models attached
↪  in the mass geometry:
fastSimulationPhysics->ActivateFastSimulation("e-");
fastSimulationPhysics->ActivateFastSimulation("e+");
fastSimulationPhysics->ActivateFastSimulation("gamma");
// - activation of fast simulation for particles having fast simulation models attached
↪  in the parallel geometry:
fastSimulationPhysics->ActivateFastSimulation("pi+","pionGhostWorld");
fastSimulationPhysics->ActivateFastSimulation("pi-","pionGhostWorld");
physicsList->RegisterPhysics( fastSimulationPhysics ); // attach to the physics list
```

Which particles?

- ☑ within selected volumes

  G4Region attached to G4LogicalVolume;and linked to implementation of G4VFastSimulationModel;

- ☑ for selected particle types

  G4FastSimulationPhysics attached to physics list and activated for particles;

- ☑ if trigger is issued

  implementation of G4VFastSimulationModel class;

  - check particle type or intrinsic information (from G4ParticleDefinition)
  - check dynamic conditions (from G4FastTrack)
    - energy, momentum, direction, ... (from G4Track)
    - local coordinates (from G4LogicalVolume)

Parameterisation trigger needs to be set in implementation of **G4VFastSimulationModel**, which is added to **G4Region**.

G4VFastSimulationModel (1/4)

Core of the parameterisation:

- **which** particles
- **how**/what happens

```
// constructor adds this model to G4FastSimulationManager of given envelope
G4VFastSimulationModel(const G4String&, G4Region*)
```

examples/extended/parameterisations/Par01/src/Par01DetectorConstruction.cc

```
G4RegionStore* regionStore = G4RegionStore::GetInstance();

G4Region* caloRegion = regionStore->GetRegion("EM_calo_region");
// builds a model and sets it to the envelope of the calorimeter:
new Par01EMShowerModel("emShowerModel",caloRegion);
```

G4VFastSimulationModel (2/4) — which particles?

Check intrinsic particle information (mass, charge, spin, quark content, ...)

```
virtual G4bool G4VFastSimulationModel::IsApplicable  (const G4ParticleDefinition&) = 0
```

Par01EMShowerModel.cc

```
G4bool Par01EMShowerModel::IsApplicable(const G4ParticleDefinition& particleType)
{
  return
    &particleType == G4Electron::ElectronDefinition() ||
    &particleType == G4Positron::PositronDefinition() ||
    &particleType == G4Gamma::GammaDefinition();
}
```

Par01PionShowerModel.cc

```
G4bool Par01PiModel::IsApplicable(const G4ParticleDefinition& particleType)
{
  return
    &particleType == G4PionMinus::PionMinusDefinition() ||
    &particleType == G4PionPlus::PionPlusDefinition();
}
```

Par02FastSimModelTracker.cc

```
G4bool Par02FastSimModelTracker::IsApplicable( const G4ParticleDefinition& aParticleType ) {
  return aParticleType.GetPDGCharge() != 0;  // Applicable for all charged particles
}
```

G4VFastSimulationModel (3/4) — which particles?

Check dynamic conditions (momentum, direction, position, distance to boundary, ...)

```
virtual G4bool G4VFastSimulationModel::ModelTrigger  (const G4FastTrack&) = 0
```

Par01EMShowerModel.cc

```
G4bool Par01EMShowerModel::ModelTrigger(const G4FastTrack& fastTrack)
{
  // Applies the parameterisation above 100 MeV:
  return fastTrack.GetPrimaryTrack()->GetKineticEnergy() > 100*MeV;
}
```

Par01PiModel.cc

```
G4bool Par01PiModel::ModelTrigger(const G4FastTrack& fastTrack) {
  // - example - position:
  fastTrack.GetPrimaryTrack()->GetPosition() // global coord.
  fastTrack.GetPrimaryTrackLocalPosition() // envelope coord.
  // - example - direction:
  fastTrack.GetPrimaryTrack()->GetMomentum().unit() // global
  fastTrack.GetPrimaryTrackLocalDirection() // envelope
  return true;
}
```

G4VFastSimulationModel (3/4) — which particles?

Check dynamic conditions (momentum, direction, position, distance to boundary, ...)

```
virtual G4bool G4VFastSimulationModel::ModelTrigger  (const G4FastTrack&) = 0
```

GFlashShowerModel.cc

```
G4bool GFlashShowerModel::ModelTrigger(const G4FastTrack & fastTrack )

{
  G4bool select = false;
  if(FlagParamType != 0)
  {
    G4double  ParticleEnergy = fastTrack.GetPrimaryTrack()->GetKineticEnergy();
    G4ParticleDefinition &ParticleType =
      *(fastTrack.GetPrimaryTrack()->GetDefinition());
    if(ParticleEnergy > PBound->GetMinEneToParametrise(ParticleType) &&
       ParticleEnergy < PBound->GetMaxEneToParametrise(ParticleType) )
    {
      // check conditions depending on particle flavour
      // performance to be optimized @@@@@@@
      Parameterisation->GenerateLongitudinalProfile(ParticleEnergy);
      select     = CheckParticleDefAndContainment(fastTrack);
      if (select) EnergyStop= PBound->GetEneToKill(ParticleType);
    }
  }
  return select;
}
```

G4VFastSimulationModel (4/4) – What happens?

Once particle is in a chosen volume, fulfils all conditions
– take over tracking within volume and decide what to do, e.g.:

- alter energy
- move to different position (e.g. exit from volume)
- create energy deposit(s)
- kill particle
- create secondaries

```
virtual G4bool G4VFastSimulationModel::DoIt(const G4FastTrack&, G4FastStep&) = 0
```

input information: G4FastTrack → output information: G4FastStep

How to deposit energy with your model?

since Geant4 v10.7

- Helper classes (`G4FastHit`, `G4FastSimHitMaker`) to deposit energy at given positions (if within the sensitive detector).
- Geant4 will look for a sensitive detector at given position, and if found – deposit energy.

`G4FastSimHitMaker::make(const G4FastHit& aHit, const G4FastTrack& aTrack)` Par03EMShowerModel.cc

```
void Par03EMShowerModel::DoIt(const G4FastTrack& aFastTrack, G4FastStep& aFastStep)
{
  ...
  G4double energy = aFastTrack.GetPrimaryTrack()->GetKineticEnergy();
  ...
  G4ThreeVector position;
  ...
  G4int generatedHits = 0;
  while(generatedHits < fNbOfHits)
  {
    position = ...
    // Create energy deposit in the detector
    // This will call appropriate sensitive detector class
    fHitMaker->make(G4FastHit(position, energy / fNbOfHits), aFastTrack);
    generatedHits++;
  }
}
```

G4VFastSimSensitiveDetector

since Geant4 v10.7

- `G4VFastSimSensitiveDetector` must me used as base class in addition to inheritance from the usual base class `G4VSensitiveDetector`.
- `ProcessHits(...)` method must be implemented and describe how hits should be saved in the hit collections.

G4VFastSimSensitiveDetector   examples/extended/parameterisations/Par03/include/zPar03SensitiveDetector.hh

```
class Par03SensitiveDetector
  : public G4VSensitiveDetector
  , public G4VFastSimSensitiveDetector
{
  ...
  /// Process energy deposit from the full simulation.
  virtual G4bool ProcessHits(G4Step* aStep, G4TouchableHistory* aROhist) final;
  /// Process energy deposit from the fast simulation.
  virtual G4bool ProcessHits(const G4FastHit* aHit, const G4FastTrack* aTrack,
                             G4TouchableHistory* aROhist) final;
  ...
}
```

Summary: fast simulation in Geant4

Step-by-step:

1. Implement model that specifies **which** particles, under what conditions and **how** should be parameterised
   user's implementation of **G4VFastSimulationModel**
2. Register the parameterisation(s) for the particles (**which**)
   by adding to physics list **G4FastSimulationManagerProcess** and activating it for certain particles (recommended: via **G4FastSimulationPhysics**)
3. Specify **where** parameterisation takes place
   by creating **G4Region**, attaching a root G4LogicalVolume, and passing it to a constructor of implementation of G4VFastSimulationModel

Existing examples: `examples/extended/parameterisations/`

UI commands

Fast simulation messenger

```
/param/ // Fast Simulation print/control commands.
/param/showSetup // Show fast simulation setup (for each world: fast simulation manager
↪ process - which particles, region hierarchy - which models)
/param/listEnvelopes <ParticleName (default:all)> // List all the envelope names for a
↪ given particle (or for all particles if without parameters).
/param/listModels <EnvelopeName (default:all)> // List all the Model names for a given
↪ envelope (or for all envelopes if without parameters).
/param/listIsApplicable <ModelName (default:all)> // List all the Particle names a
↪ given model is applicable (or for all models if without parameters).
/param/ActivateModel <ModelName> // Activate a given Model.
/param/InActivateModel <ModelName> // InActivate a given Model.
```

Examples

Examples

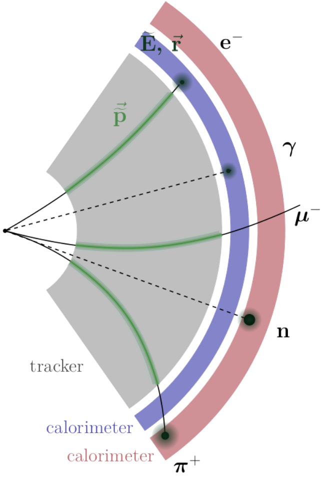
Existing examples: `examples/extended/parameterisations/`

- examples/extended/parameterisations/Par01/src/
  - Par01EMShowerModel.cc
  - Par01PionShowerModel.cc
  - Par01PiModel.cc
- examples/extended/parameterisations/Par02/src/
  - Par02FastSimModelEMCal.cc
  - Par02FastSimModelHCal.cc
  - Par02FastSimModelTracker.cc
- examples/extended/parameterisations/Par03/src/
  - Par03EMShowerModel.cc
- examples/extended/parameterisations/Par04/src/
  - Par04MLFastSimModel.cc
- GFlashShowerModel

Example 1:

examples/extended/parameterisations/Par02

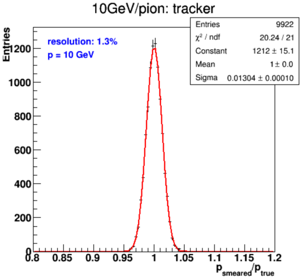
Example 1

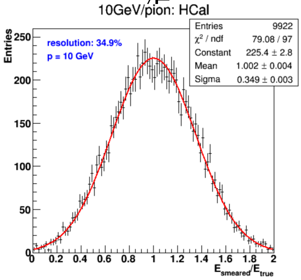


- Simple parameterisation
- Smearing of the momentum in the tracker and energy in the calorimeter
- User input: detector resolution;

$$\sigma_{p_T} = 1.3\%$$

$$\sigma_E = \frac{110\%}{\sqrt{E}} \oplus 9\%$$





Example 1: detector construction

- from GDML;
- explore auxiliary information field to create regions

Par02DetectorConstruction.cc

```
G4VPhysicalVolume* Par02DetectorConstruction::Construct() {
  G4GDMLParser parser;
  parser.Read( "Par02FullDetector.gdml" );
  const G4GDMLAuxMapType* aAuxMap = parser.GetAuxMap();
  for ( G4GDMLAuxMapType::const_iterator iter = aAuxMap->begin(); iter != aAuxMap->end(); ++iter ) {
    for ( G4GDMLAuxListType::const_iterator vit = (*iter).second.begin(); vit !=
↪   (*iter).second.end(); ++vit ) {
      if ( (*vit).type == "FastSimModel" ) {
        G4LogicalVolume* myvol = (*iter).first;
        if ( ( myvol->GetName() ).find( "Tracker" ) != std::string::npos ) {
          fTrackerList.push_back( new G4Region( myvol->GetName() ) );
          fTrackerList.back()->AddRootLogicalVolume( myvol );
        } else [...]
      }
    }
  } ...
}
```

Example 1: detector construction

- from GDML;
- explore auxiliary information field to create regions and fast simulation models

Par02FullDetector.gdml

```
<volume name="TrackerBarrelLog">
  <materialref ref="Beryllium0x7ff5f9e3baf0"/>
  <solidref ref="TrackerBarrel"/>
  <auxiliary auxtype="FastSimModel"
    auxvalue="TrackerBarrel"/>
</volume>
```

Par02DetectorConstruction.cc

```
void Par02DetectorConstruction::ConstructSDandField() {
  for ( G4int iterTracker = 0; iterTracker < G4int(
    fTrackerList.size() ); iterTracker++ ) {
    // Bound the fast simulation model for the tracker subdetector
    // to all the corresponding Geant4 regions
    Par02FastSimModelTracker* fastSimModelTracker
    = new Par02FastSimModelTracker( "fastSimModelTracker",
    fTrackerList[ iterTracker ], Par02DetectorParametrisation::eCMS );
    // Register the fast simulation model for deleting
    G4AutoDelete::Register(fastSimModelTracker);
  }...
}
```

Example 1: physics list

- register by-hand G4FastSimulationManagerProcess
- process registered for all constructed particles

(Not recommended!
But also works for versions < 10.3)

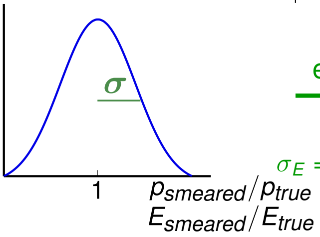
Par02PhysicsList.cc

```
void Par02PhysicsList::AddParameterisation() {
  G4FastSimulationManagerProcess* fastSimProcess =
    new G4FastSimulationManagerProcess( "G4FSMP" );
  // Registers the fastSimProcess with all the particles as a discrete
↪   and
  // continuous process (this works in all cases; in the case that
↪   parallel
  // geometries are not used, as in this example, it would be enough to
  // add it as a discrete process).
  auto particleIterator=GetParticleIterator();
  particleIterator->reset();
  while ( (*particleIterator)() ) {
    G4ParticleDefinition* particle = particleIterator->value();
    G4ProcessManager* pmanager = particle->GetProcessManager();
    //pmanager->AddDiscreteProcess( fastSimProcess );    // No parallel
↪   geometry
    pmanager->AddProcess( fastSimProcess, -1, 0, 0 );  // General
  }
}
```

Example 1: models

- smearing of momentum (tracker) / energy (calorimeters) with Gaussian;
- resolution defined arbitrarily in Par02DetectorParametrisation ([E] = GeV)

	CMS-like	ALEPH-like	ATLAS-like
σ (Tracker)	1.3%	1%	1%
σ (EMCAL)	$\frac{3\%}{\sqrt{E}} \oplus \frac{12\%}{E} \oplus 0.3\%$	$\frac{18\%}{\sqrt{E}} \oplus 0.9\%$	$\frac{10\%}{\sqrt{E}} \oplus 0.17\%$
σ (HCAL)	$\frac{110\%}{\sqrt{E}} \oplus 9\%$	$\frac{85\%}{\sqrt{E}}$	$\frac{55\%}{\sqrt{E}} \oplus 6\%$

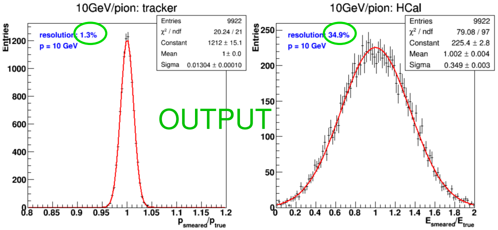


e.g. 10 GeV π^-

$\sigma_p = 1.3\%$

$\sigma_E = \frac{110\%}{\sqrt{E}} \oplus 9\% \approx 35\%$

INPUT



Example 1: models

- Tracker: transport it in EM field to the exit-from-envelope, smear momentum;

Par02FastSimModelTracker.cc

```
void Par02FastSimModelTracker::DoIt( const G4FastTrack& aFastTrack, G4FastStep& aFastStep ) {
  G4Track track = * aFastTrack.GetPrimaryTrack();
  ...
  G4PathFinder* fPathFinder = G4PathFinder::GetInstance();
  fPathFinder->ComputeStep( ... );
  aFastStep.ProposePrimaryTrackFinalPosition( endTrack.GetPosition() );
}
```

- Calorimeters: particles deposit all energy and are killed (e/γ in EMCal, hadrons in HCal);

Par02FastSimModelEMCal.cc

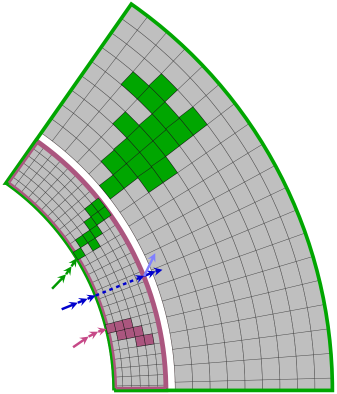
```
void Par02FastSimModelEMCal::DoIt( const G4FastTrack& aFastTrack, G4FastStep& aFastStep ) {
  aFastStep.KillPrimaryTrack();
  aFastStep.ProposePrimaryTrackPathLength( 0.0 );
  G4double Edep = aFastTrack.GetPrimaryTrack()->GetKineticEnergy();
  G4double Esm; [...] // Esm = smeared Edep
  aFastStep.ProposeTotalEnergyDeposited( Esm );
}
```

Example 2:

`examples/extended/parameterisations/Par01`

Example 2

Time consuming simulation of calorimeters replaced by creation of energy deposits.



Par01EMShowerModel.cc

- electrons and photons
- electromagnetic calorimeter, envelope in mass geometry

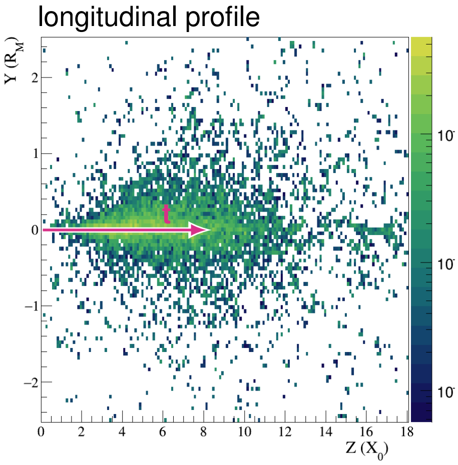
Par01PionShowerModel.cc

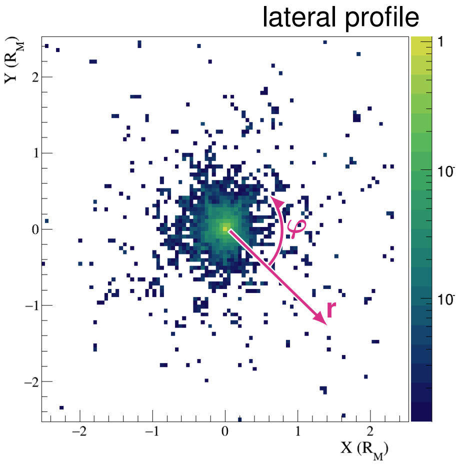
- pions
- both calorimeters: envelope around EMCal and HCal $\Rightarrow$ parallel geometry

Par01PiModel.cc

- create secondaries

Shower profiles





Example 2 – models

How to deposit energy E of electrons/photons? Par01EMShowerModel.cc

$$f(t,r,\varphi) = f(t)f(r)f(\varphi)$$

1. longitudinal shower profile $f(t)$
2. lateral profile $f(r)$
3. flat azimuthal angle distribution $f(\varphi)$

$$f(\varphi) = \frac{1}{2\pi}, \quad f(t;k,\theta) = \frac{x^{k-1}e^{-\frac{x}{\theta}}}{\theta^k \Gamma(k)}, \quad f(r) = \begin{cases} \frac{0.9}{2 \cdot R_M} & \text{for} \quad |r| \leqslant R_M \\ \frac{0.1}{5 \cdot R_M} & \text{for} \quad R_M < |r| \leqslant 3.5 \cdot R_M \\ 0 & \text{for} \quad |r| \geqslant 3.5 \cdot R_M \end{cases}$$

4. deposit energy $\Delta E = \frac{E}{N}$ in $N = 100$ points
   - pick t, r and φ from $f(t)$, $f(r)$, and $f(\varphi)$

   in (t,r,φ) inside electromagnetic calorimeter

Example 2 – models

Par01EMShowerModel.cc

```
void Par01EMShowerModel::DoIt(const G4FastTrack& fastTrack, G4FastStep& fastStep) {
  ...
  BuildDetectorResponse();
}
```

```
void Par01EMShowerModel::BuildDetectorResponse() {
  for (size_t i = 0; i < feSpotList.size(); i++) {
    AssignSpotAndCallHit(feSpotList[i]);
  }
}
```

```
void Par01EMShowerModel::AssignSpotAndCallHit(const Par01EnergySpot &eSpot)
{
  FillFakeStep(eSpot);
  G4VPhysicalVolume* pCurrentVolume = fFakeStep->GetPreStepPoint()->GetPhysicalVolume();
  G4VSensitiveDetector* pSensitive;
  if( pCurrentVolume != 0 ) {
    pSensitive = pCurrentVolume->GetLogicalVolume()->GetSensitiveDetector();
    if( pSensitive != 0 ) {
      pSensitive->Hit(fFakeStep);
    }
  }
}
```

Example 2 – models

How to deposit energy E of pions?

Par01PionShowerModel.cc

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 / 2\sigma^2}$$

1. longitudinal shower profile $f(t, 0, 20\text{cm})$
2. lateral profile $f(r, 0, 10\text{cm})$
3. azimuthal angle

$$f(\varphi) = \frac{1}{2\pi}$$

4. deposit energy $\Delta E = \frac{E}{N}$ in $N = 50$ points
   - pick t, r and φ from $f(t)$, $f(r)$, and $f(\varphi)$

   in (t,r,φ) inside electromagnetic + hadronic calorimeter envelope

Example 2 – models

How to create secondaries?

Par01PiModel.cc

Par01PiModel.cc

```
// - First, user has to say how many secondaries will be created:
fastStep.SetNumberOfSecondaryTracks(1);
G4ParticleMomentum direction(fastTrack.GetPrimaryTrackLocalDirection());
direction.setZ(direction.z()*0.5);
direction.setY(direction.y()+direction.z()*0.1);
direction = direction.unit(); // necessary ?
// - dynamics (Note that many constructors exists for G4DynamicParticle
G4DynamicParticle dynamique(G4Gamma::GammaDefinition(),
                            direction,
                            fastTrack.GetPrimaryTrack()->
                            GetKineticEnergy()/2.);
G4double Dist;
Dist = fastTrack.GetEnvelopeSolid()->
DistanceToOut(fastTrack.GetPrimaryTrackLocalPosition(),
              direction);
G4ThreeVector posi;
posi = fastTrack.GetPrimaryTrackLocalPosition() + Dist*direction;
fastStep.CreateSecondaryTrack(dynamique, posi,
                              fastTrack.GetPrimaryTrack()->GetGlobalTime());
```

Example 2 – models

How to transport particles to the outer boundary?

Par01PiModel.cc

Par01PiModel.cc

```
  G4ThreeVector position;
  G4double distance;
  distance = fastTrack.GetEnvelopeSolid()->
    DistanceToOut(fastTrack.GetPrimaryTrackLocalPosition(),
                  fastTrack.GetPrimaryTrackLocalDirection());
  position = fastTrack.GetPrimaryTrackLocalPosition() +
    distance*fastTrack.GetPrimaryTrackLocalDirection();

  // - set final position:
  fastStep.ProposePrimaryTrackFinalPosition(position);
```

Example 3:

examples/extended/parameterisations/Par03

Example 3 – EM shower model

Based on PDG (chapter 33.5) Par03EMShowerModel.cc

1. longitudinal shower profile

$$\frac{\mathrm{d}E}{\mathrm{d}t} = E_0 b \frac{(bt)^{a-1} \mathrm{e}^{-bt}}{\Gamma(a)} \quad (33.35)$$

$$b = 0.5 \ \ (Fig\ 33.21), \quad \frac{a-1}{b} = ln\frac{E}{E_c} + C_j, \ \ C_j = \begin{cases} +0.5 & \text{for} \gamma \\ -0.5 & \text{for} e^{\pm} \end{cases} \quad (33.36)$$

2. Gaussian lateral profile with 90% energy deposited within a cylinder of radius equal to Moliere radius
3. Deposit energy $\Delta E = \frac{E}{N}$ in N (100 by default) points sampling position from Gamma and Gaussian distributions
4. Created hits are deposited in the detector using its readout geometry, using the helper class `G4FastSimHitMaker` that locates the volume, and calls appropriate sensitive detector class.

Example 4:

examples/extended/parameterisations/gflash/gflash1

Example 4 - previous Geant4 versions

Prior to v10.6, example `parameterisations/gflash/gflash1` was `parameterisations/gflash/`.

Set of examples is extended, to present different options:

Example	`gflash1`	`gflash2`	`gflash3`
Block of homogeneous material	mass geo	mass geo	mass geo
Crystals (readout geometry)	mass geo	mass geo	parallel geo
Sensitive detector	mass geo	mass geo	parallel geo
Envelope for parameterisation	mass geo	parallel geo	mass geo

Additionally, `examples/extended/parameterisations/gflash/gflasha` contains simple post-event analysis of shower shapes.

All examples feature parametrisation of the same homogeneous calorimeter, only technical details change.

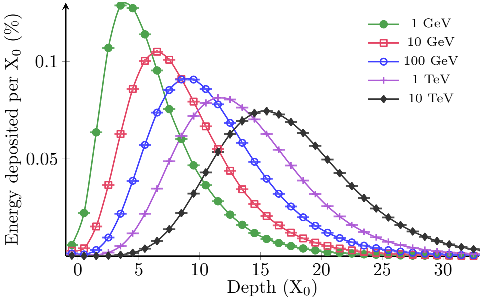
Example 4

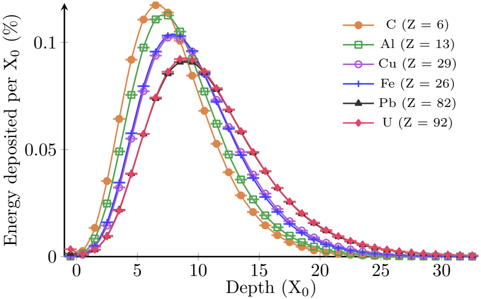
- the only implementation of G4VFastSimulationModel in Geant4 (outside `examples/`)
- `arXiv:hep-ex/0001020`
- physics reference manual, chapter 18
- parameterisation of electromagnetic cascades:

$$dE(\bar{r}) = Ef(t)dtf(r)drf(\varphi)d\varphi$$

- flat distribution in azimuthal angle $f(\varphi) = \frac{1}{2\pi}$
- $f(t)$ and $f(r)$ parametrised as a function of particle's energy (E) and medium (Z)
- t and r are expressed in units of X_0 and R_M

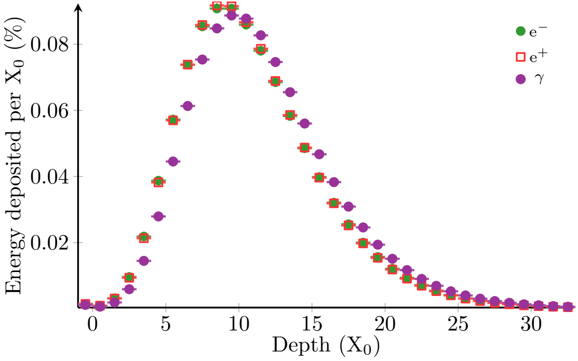
Example 4 - longitudinal profile





$$T \sim \ln E$$

Example 4 - longitudinal profile



Example 4 - longitudinal profile

$$f(t) = \left\langle \frac{1}{E}\frac{dE(t)}{dt} \right\rangle = \frac{(\beta t)^{\alpha-1}\beta e^{-\beta t}}{\Gamma(\alpha)}$$

- shower maximum $T = \frac{\alpha-1}{\beta}$
- Description dependent on $y = \frac{E}{E_c}$:

$$T = \ln y + \textit{l}_1$$
$$\alpha = \textit{l}_2 + (\textit{l}_3 + \frac{\textit{l}_4}{Z}) \ln y$$

A.1 Homogeneous Media

A.1.1 Average longitudinal profiles

$$\begin{aligned} T_{hom} &= \ln y - 0.858 \\ \alpha_{hom} &= 0.21 + (0.492 + 2.38/Z) \ln y \end{aligned}$$

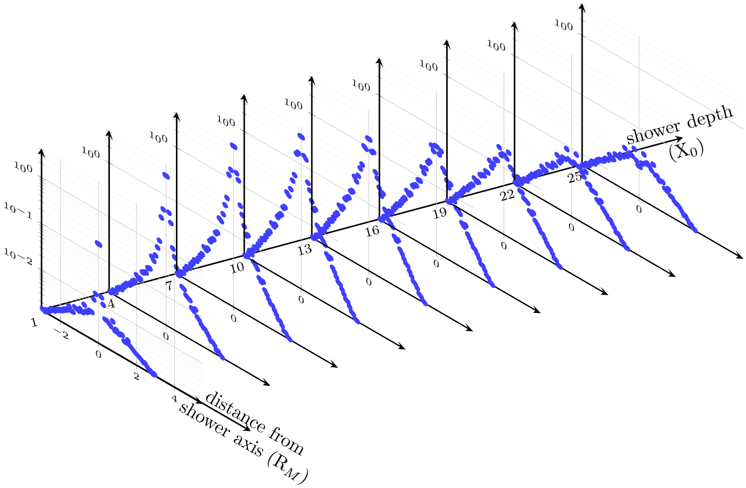
A.1.2 Fluctuated longitudinal profiles

$$\begin{aligned} \langle \ln T_{hom} \rangle &= \ln(\ln y - 0.812) \\ \sigma(\ln T_{hom}) &= (-1.4 + 1.26 \ln y)^{-1} \\ \langle \ln \alpha_{hom} \rangle &= \ln(0.81 + (0.458 + 2.26/Z) \ln y) \\ \sigma(\ln \alpha_{hom}) &= (-0.58 + 0.86 \ln y)^{-1} \\ \rho(\ln T_{hom}, \ln \alpha_{hom}) &= 0.705 - 0.023 \ln y \end{aligned}$$

arXiv:hep-ex/0001020

Example 4 – lateral profile

examples/extended/parameterisations/gflash/gflash1



Example 4 – lateral profile

$$f(r) = \left\langle \frac{1}{dE(t)} \frac{dE(t,r)}{dr} \right\rangle = pf_{\mathrm{core}}(r) + (1-p) f_{\mathrm{tail}}(r) =$$

$$= p \frac{2rR_{\mathrm{core}}^2}{\left(r^2 + R_{\mathrm{core}}^2\right)^2} + (1-p) \frac{2rR_{\mathrm{tail}}^2}{\left(r^2 + R_{\mathrm{tail}}^2\right)^2}$$

Description dependent on $\tau = \frac{t}{T}$:

$$R_{\mathrm{core}}(\tau) = \boldsymbol{r_1} + \boldsymbol{r_2}\tau$$

$$R_{\mathrm{tail}}(\tau) = \boldsymbol{r_3} \left(\mathrm{e}^{\boldsymbol{r_4}(\tau - \boldsymbol{r_5})} + \mathrm{e}^{\boldsymbol{r_6}(\tau - \boldsymbol{r_7})} \right)$$

$$p(\tau) = \boldsymbol{r_8} \exp\left(\frac{\boldsymbol{r_9} - \tau}{\boldsymbol{r_{10}}} - \exp\left(\frac{\boldsymbol{r_9} - \tau}{\boldsymbol{r_{10}}} \right) \right)$$

A.1.3 Average radial profiles

$$R_{C,hom}(\tau) = z_1 + z_2 \tau$$
$$R_{T,hom}(\tau) = k_1\{\exp(k_3(\tau - k_2)) + \exp(k_4(\tau - k_2))\}$$
$$p_{hom}(\tau) = p_1 \exp\left\{ \frac{p_2 - \tau}{p_3} - \exp\left(\frac{p_2 - \tau}{p_3} \right) \right\}$$

with

$$z_1 = 0.0251 + 0.00319 \ln E$$
$$z_2 = 0.1162 + -0.000381 Z$$
$$k_1 = 0.659 + -0.00309 Z$$
$$k_2 = 0.645$$
$$k_3 = -2.59$$
$$k_4 = 0.3585 + 0.0421 \ln E$$
$$p_1 = 2.632 + -0.00094 Z$$
$$p_2 = 0.401 + 0.00187 Z$$
$$p_3 = 1.313 + -0.0686 \ln E$$

A.1.4 Fluctuated radial profiles

$$\tau_i = \frac{t}{\langle t \rangle_i} \frac{\exp(\langle \ln \alpha \rangle)}{\exp(\langle \ln \alpha \rangle) - 1}$$
$$N_{Spot} = 93 \ln(Z) E^{0.876}$$
$$T_{Spot} = T_{hom}(0.698 + 0.00212 Z)$$
$$\alpha_{Spot} = \alpha_{hom}(0.639 + 0.00334 Z)$$

arXiv:hep-ex/0001020

Example 4 – model

examples/extended/parameterisations/gflash/gflash1

ExGflashDetectorConstruction.cc

```
void ExGflashDetectorConstruction::ConstructSDandField()
{
  // - sensitive detectors:
  G4SDManager* SDman = G4SDManager::GetSDMpointer();
  ExGflashSensitiveDetector* CaloSD
  = new ExGflashSensitiveDetector("Calorimeter",this);
  SDman->AddNewDetector(CaloSD);
  fCrystal_log->SetSensitiveDetector(CaloSD);

  // Get nist material manager
  G4NistManager* nistManager = G4NistManager::Instance();
  G4Material*          pbWO4 = nistManager->FindOrBuildMaterial("G4_PbWO4");
  // - fast simulation models:
  // **********************************************
  // * Initializing shower modell
  // ***********************************************
  G4cout « "Creating shower parameterization models" « G4endl;
  fFastShowerModel = new GFlashShowerModel("fFastShowerModel", fRegion);
  fParameterisation = new GFlashHomoShowerParameterisation(pbWO4);
  fFastShowerModel->SetParameterisation(*fParameterisation);
  // Energy Cuts to kill particles:
  fParticleBounds = new GFlashParticleBounds();
  fFastShowerModel->SetParticleBounds(*fParticleBounds);
  // Makes the EnergieSpots
  fHitMaker = new GFlashHitMaker();
  fFastShowerModel->SetHitMaker(*fHitMaker);
  G4cout«"end shower parameterization."«G4endl;
  // **********************************************
}
```

Example 4 – model

ExGflashSensitiveDetector.hh

```
class ExGflashSensitiveDetector: public G4VSensitiveDetector,
                                 public G4VGFlashSensitiveDetector {
            ...
            virtual G4bool ProcessHits(G4Step*,G4TouchableHistory*);
            virtual G4bool ProcessHits(G4GFlashSpot*aSpot,G4TouchableHistory*);
            ...
};
```

`GVFlashHomoShowerTuning` can be used to change parameters (l_1, l_2, ... , r_1, ...)

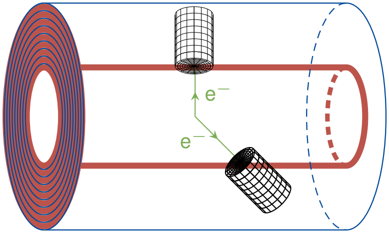
Sampling calorimeter For simulation in sampling detectors use `GFlashSamplingShowerParameterisation` and `GFlashSamplingShowerTuning`. Readout should collect signal from both active and pasive material (e.g. by constructing SD in parallel world). Those calorimeters have not been tested in Geant4, so implementation of GFlash in Geant4 may require further work (which is on-going).

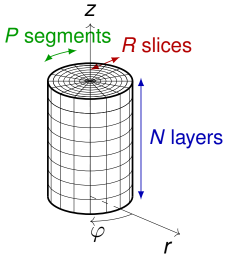
Example 5:

`examples/extended/parameterisations/Par04`

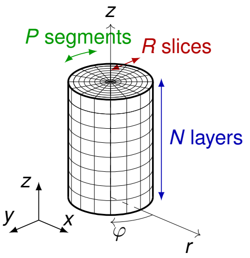
Example 5

- examples/extended/parameterisations/Par04 new example since Geant4 11.0 release
- Detector geometry is simplistic and easy to configure
- Collider-style concentric cylinders with up to two materials (active and optionally passive)
- Particle direction and position is measured at the entrance to calorimeter (many possible ways to do it: we chose fast sim model that is attached to calorimeter)
- Scoring of energy deposits is done relative to the particle direction
- Similar granularity 'pictures' are obtained independently on angle
- Granularity of shower deposition is configurable

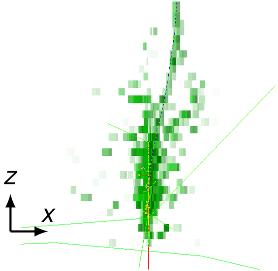




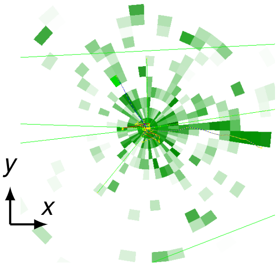
Example 5 – EM showers



10 GeV electron

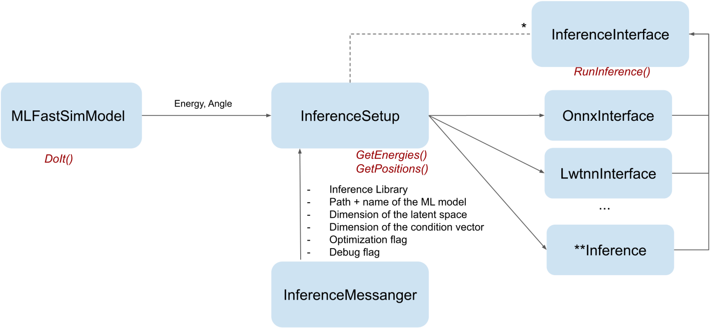


10 GeV electron



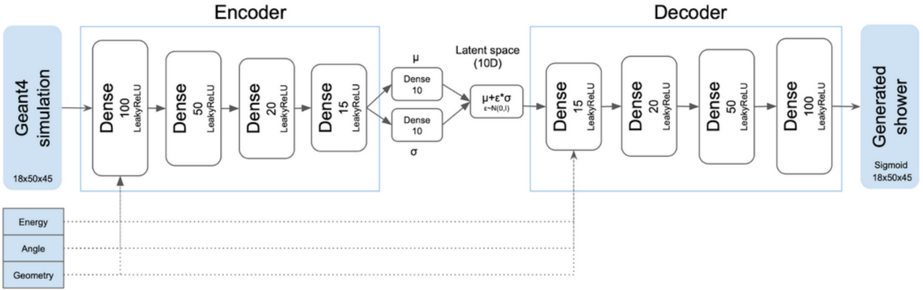
- Example uses 0.3 mm Si and 1.4 mm W layers
- Readout granularity is $\Delta r \times \Delta\varphi \times \Delta z = 2.3\,\text{mm} \times \frac{2\pi}{50} \times 3.4\,\text{mm}$ aiming for $\Delta r \approx 0.25\, R_M$ and $\Delta z \approx 0.6\, X_0$
- Number of readout cells is $R \times P \times N = 18x50x45$ aiming for 95% containment of 1 TeV particles
- **Open access dataset for SiW (and scintillator-Pb) released 10.5281/zenodo.6082201**
- This dataset is a base of ML studies, including CaloChallenge.

Example 5 – inference within C++ framework



- Fast simulation with ML within Geant4
- Demonstrates how to incorporate inference libraries (ONNX Runtime, LWTNN)
- Par04 can run full and fast simulation (if any of the inference libraries is available, e.g. via LCG)

Example 5 – ML model



Variational autoencoder that is a subject of study in our group. Provided model is trained on the specified geometry, it **requires changes with the changes to the geometry**. More details on work on generalisation are discussed in our webpage: g4fastsim.web.cern.ch

Summary

- Fast simulation can be used within Geant4;
- Hooks to take over control in chosen volumes/for chosen particles;
- Seamless mix of detailed and parametric simulation;
- Examples in `examples/extended/parameterisations/`;

Questions/problems? anna.zaborowska@cern.ch