Version 10.5

Fast Simulation

Anna Zaborowska

Geant4 Advanced Course @ CERN

March 26–27, 2019

Outline

1. Why do we need fast simulation?
2. How to use it in Geant4
   - where
   - what
   - how

   } to parametrise
3. Short summary
4. Examples

Outline

1. Why do we need 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 v10.5

```
...
```

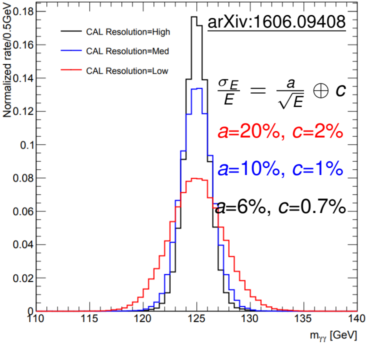
Why to use parametrisation / fast(er) simulation?

Why to use parametrisation / fast(er) simulation?

physics studies that assume certain detector performance

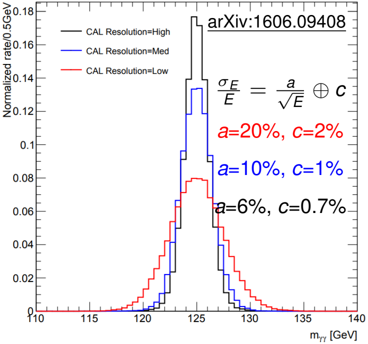
Why to use parametrisation / fast(er) simulation?

physics studies that assume certain detector performance



Why to use parametrisation / fast(er) simulation?

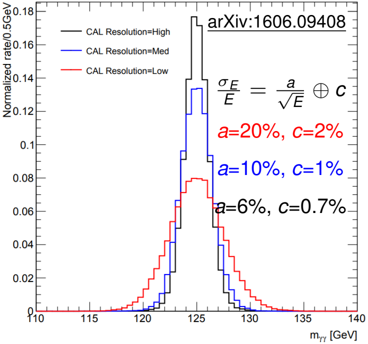
physics studies that assume certain detector performance

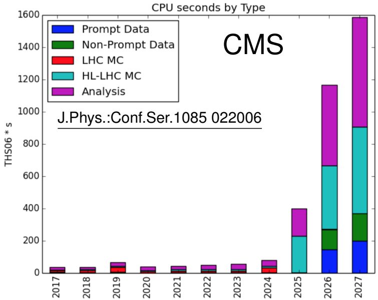


more data (⇒ CPU time) needed

Why to use parametrisation / fast(er) simulation?

physics studies that assume certain detector performance

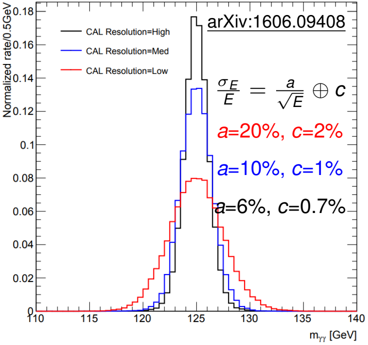


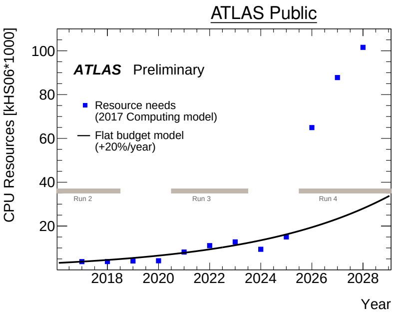


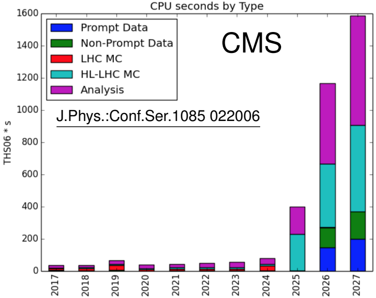
more data (⇒ CPU time) needed

Why to use parametrisation / fast(er) simulation?

physics studies that assume certain detector performance

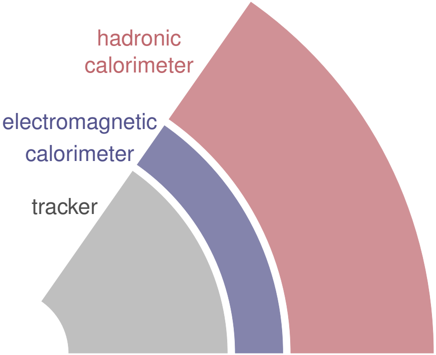




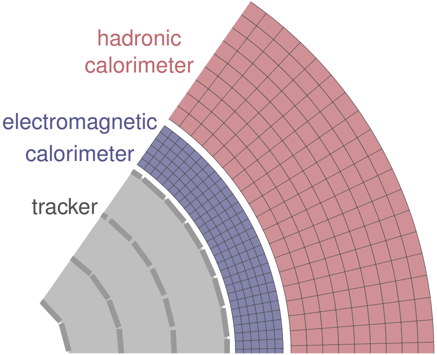


more data (⇒ CPU time) needed

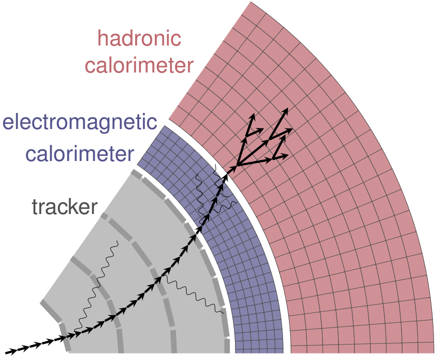
How we simulate particles?



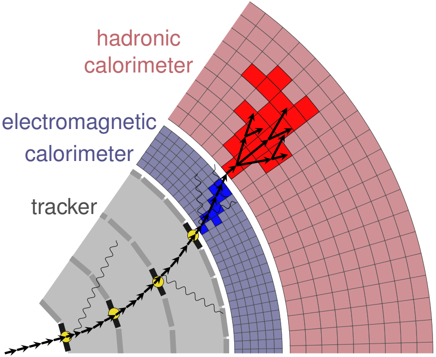
How we simulate particles?



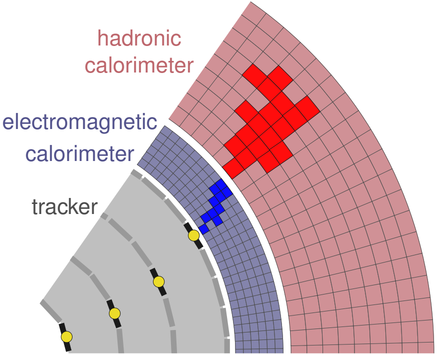
How we simulate particles?



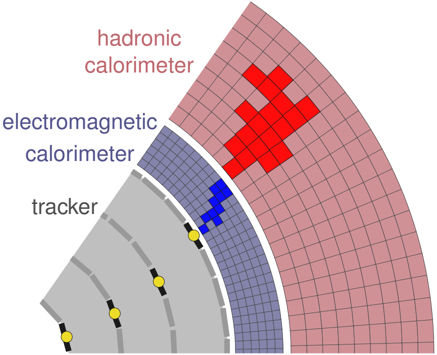
How we simulate particles?

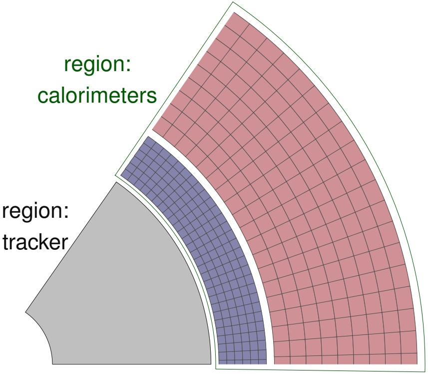


How we simulate particles?

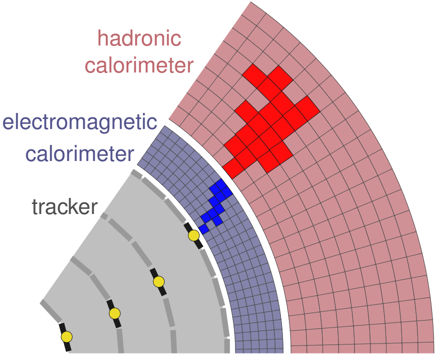


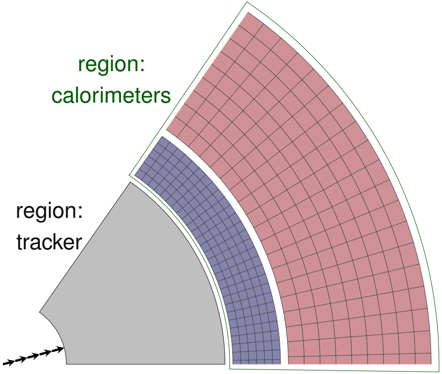
How we simulate particles?



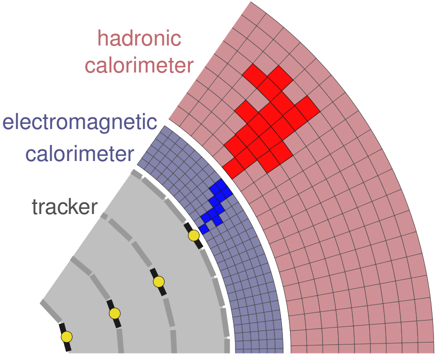


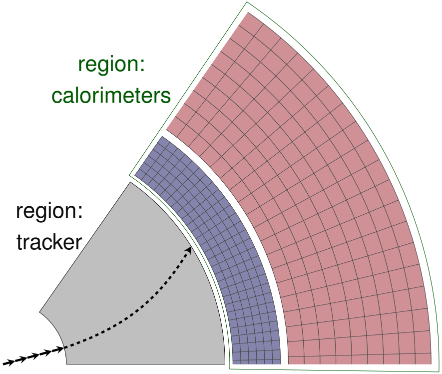
How we simulate particles?





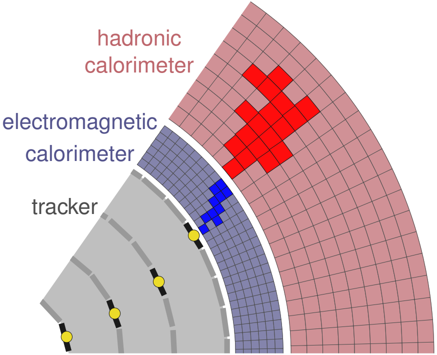
How we simulate particles?

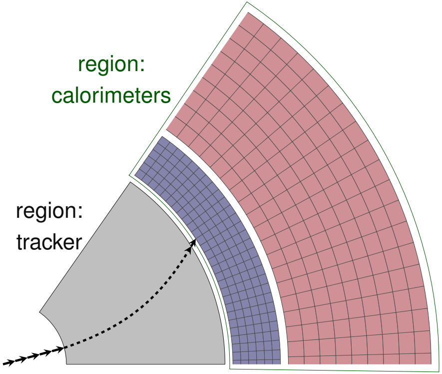




Fast simulation is a shortcut to the standard tracking and detailed simulation.

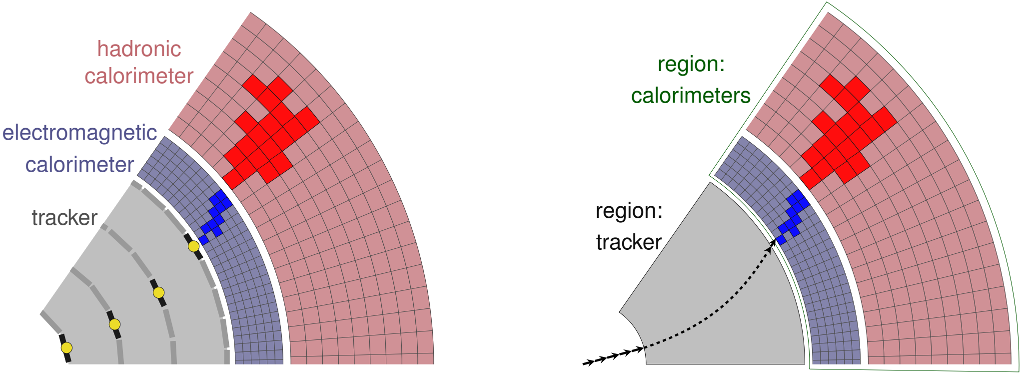
How we simulate particles?





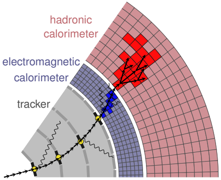
Fast simulation is a shortcut to the standard tracking and detailed simulation.

How we simulate particles?



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

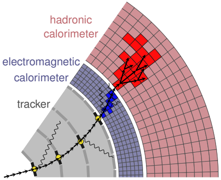
Simulation of particle passage



detailed / “full” simulation

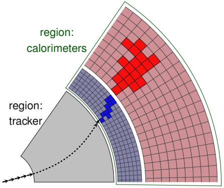
- detailed detector description
- definitions of particles and processes
- transport in e-m field

Simulation of particle passage



detailed / "full" simulation

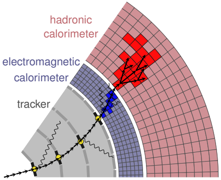
parametrisation / "fast" simulation



- detailed detector description
- definitions of particles and processes
- transport in e-m field

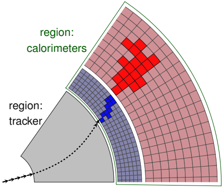
- where particles are parametrised
- which particles
- how/what happens

Simulation of particle passage



detailed / “full” simulation

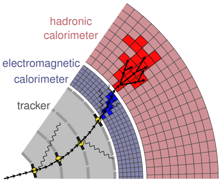
parametrisation / “fast” simulation



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

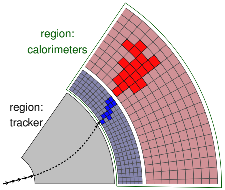
- where particles are parametrised
- which particles
- how/what happens
- detector / use-case dependent

Simulation of particle passage



detailed / “full”
simulation

parametrisation /
“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?

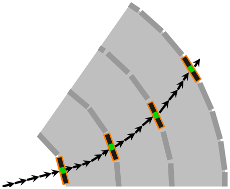
Parametrisation may be realised within:

Where?

Parametrisation may be realised within:

sub-volume

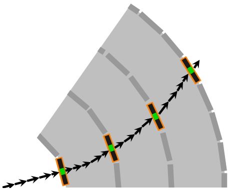
(many volumes)



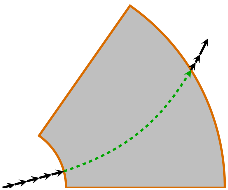
Where?

Parametrisation may be realised within:

sub-volume (many volumes)



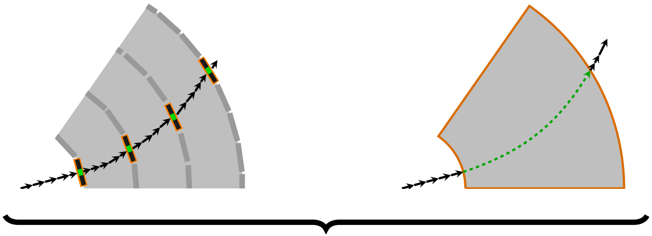
detector envelope (single volume)



Where?

Parametrisation may be realised within:

sub-volume (many volumes) | detector envelope (single volume)

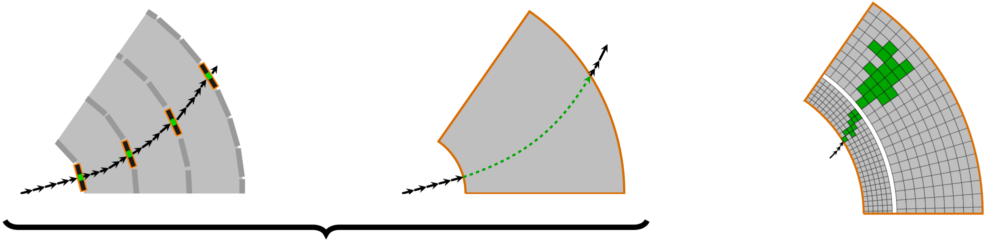


"mass" geometry

Where?

Parametrisation may be realised within:

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

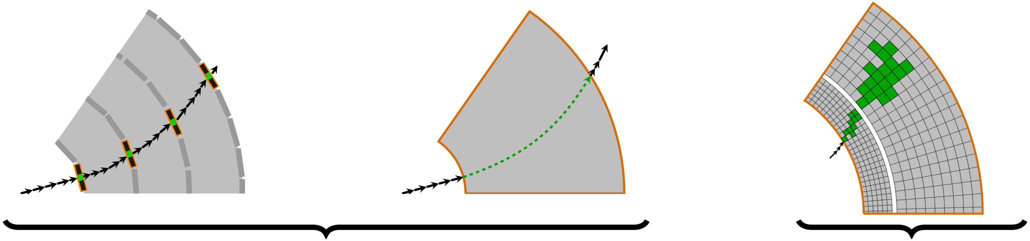


"mass" geometry

Where?

Parametrisation may be realised within:

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



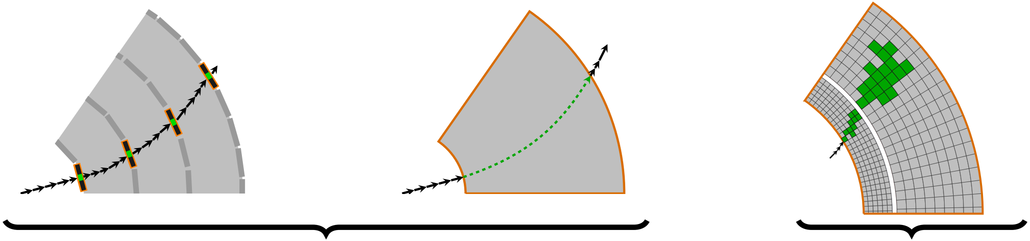
“mass” geometry

parallel geometry

Where?

Parametrisation 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 (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
```

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);
```

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?

Parametrisation is usually specified for selected particles.

Which particles?

Parametrisation is usually specified for selected particles.

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

Which particles?

Parametrisation is usually specified for selected particles.

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

It will invoke parametrisation (if all conditions met):

Which particles?

Parametrisation is usually specified for selected particles.

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

It will invoke parametrisation (if all conditions met):

- within selected volumes;

Which particles?

Parametrisation is usually specified for selected particles.

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

It will invoke parametrisation (if all conditions met):

- ☐ within selected volumes;
- ☐ for selected particle types;

Which particles?

Parametrisation is usually specified for selected particles.

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

It will invoke parametrisation (if all conditions met):

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

Which particles?

Parametrisation is usually specified for selected particles.

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

It will invoke parametrisation (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 parametrisation process on top of any other physics list (which is used where parametrisation is not invoked).

(since v10.3, for older versions consult user's guide or slide 25)

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
```

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
```

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
```

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

Which particles?

- within selected volumes

  G4Region attached to G4LogicalVolume;

Which particles?

- within selected volumes

  G4Region attached to G4LogicalVolume;

- for selected particle types

Which particles?

- ☐ within selected volumes

  G4Region attached to G4LogicalVolume;

- ☑ for selected particle types

  G4FastSimulationPhysics attached to physics list and activated for particles;

Which particles?

- ☐ within selected volumes

  G4Region attached to G4LogicalVolume;

- ☑ for selected particle types

  G4FastSimulationPhysics attached to physics list and activated for particles;

- ☐ if trigger is issued

Which particles?

- ☐ within selected volumes

  G4Region attached to G4LogicalVolume;

- ☑ for selected particle types

  G4FastSimulationPhysics attached to physics list and activated for particles;

- ☐ if trigger is issued

  - check particle type or intrinsic information (from G4ParticleDefinition)

Which particles?

- ☐ within selected volumes

  G4Region attached to G4LogicalVolume;

- ☑ for selected particle types

  G4FastSimulationPhysics attached to physics list and activated for particles;

- ☐ if trigger is issued

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

Which particles?

- ☐ within selected volumes

  G4Region attached to G4LogicalVolume;

- ☑ for selected particle types

  G4FastSimulationPhysics attached to physics list and activated for particles;

- ☐ if trigger is issued

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

Which particles?

- ☐ within selected volumes

  G4Region attached to G4LogicalVolume;

- ☑ 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)

Parametrisation trigger needs to be set in implementation of **G4VFastSimulationModel**,

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)

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

G4VFastSimulationModel (1/4)

Core of the parametrisation:

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

G4VFastSimulationModel (1/4)

Core of the parametrisation:

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

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

G4VFastSimulationModel (1/4)

Core of the parametrisation:

- **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 (1/4)

Core of the parametrisation:

- 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
```

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)
{



                                                          }
}
```

Par01PionShowerModel.cc

```
G4bool Par01PiModel::IsApplicable(const
↪   G4ParticleDefinition& particleType)
{



}
```

Par02FastSimModelTracker.cc

```
G4bool Par02FastSimModelTracker::IsApplicable( const
↪   G4ParticleDefinition& aParticleType ) {

}
```

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 ) {

}
```

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
```

G4VFastSimulationModel (3/4) — which particles?

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

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

Par01PionShowerModel.cc

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

G4VFastSimulationModel (3/4) — which particles?

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

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

Par01PionShowerModel.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
```

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

Summary: fast simulation in Geant4

Step-by-step:

Summary: fast simulation in Geant4

Step-by-step:

1. Implement model that specifies **which** particles, under what conditions and **how** should be parameterised

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**

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**)

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

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
3. Specify **where** parametrisation takes place

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
3. Specify **where** parametrisation takes place
   by creating **G4Region**, attaching a root G4LogicalVolume, and passing it to a constructor of implementation of G4VFastSimulationModel

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
3. Specify **where** parametrisation takes place
   by creating **G4Region**, attaching a root G4LogicalVolume, and passing it to a constructor of implementation of G4VFastSimulationModel

Existing examples: `examples/extended/parameterisations/`

Messenger

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`
- `GFlashShowerModel`

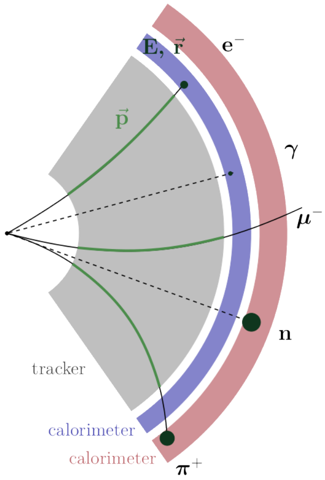
Example 1:

`examples/extended/parameterisations/Par02`

Example 1

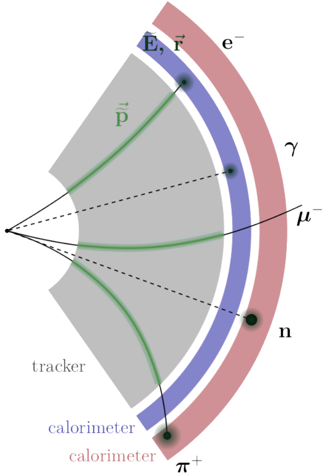
- Simple parametrisation

Example 1



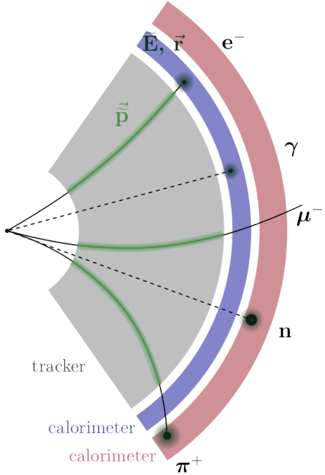
- Simple parametrisation

Example 1



- Simple parametrisation
- Smearing of the momentum in the tracker and energy in the calorimeter

Example 1



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

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

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

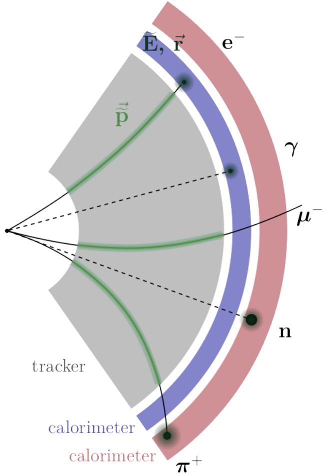
Example 1

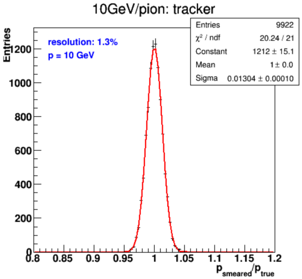
examples/extended/parameterisations/Par02

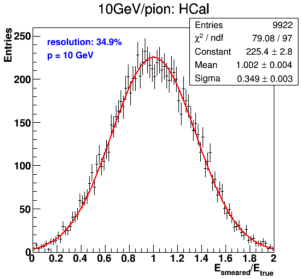
- Simple parametrisation
- Smearing of the momentum in the tracker and energy in the calorimeter
- User input: detector resolution;

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

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







Example 1: detector construction

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

Example 1: detector construction

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

Par02FullDetector.gdml

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

Example 1: detector construction

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

Par02FullDetector.gdml

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

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

examples/extended/parameterisations/Par02

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

Par02FullDetector.gdml

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

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

examples/extended/parameterisations/Par02

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

Par02FullDetector.gdml

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

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

Par02FullDetector.gdml

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

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

GEANT4 A SIMULATION TOOLKIT

- 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: detector construction

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

Par02FullDetector.gdml

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

Par02DetectorConstruction.cc

```cpp
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 (works also for versions < 10.3)
- process registered for all constructed particles

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: physics list

- register by-hand G4FastSimulationManagerProcess (works also for versions < 10.3)
- process registered for all constructed particles

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: physics list

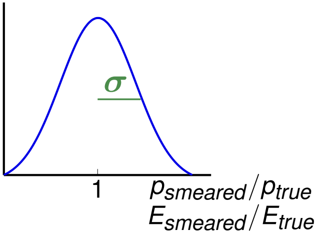
- register by-hand G4FastSimulationManagerProcess (works also for versions < 10.3)
- process registered for all constructed particles

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

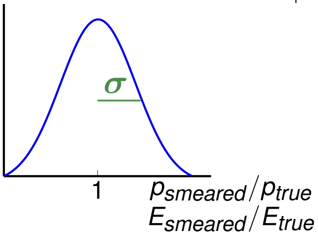
- smearing of momentum (tracker) / energy (calorimeters) with Gaussian;



Example 1: models

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

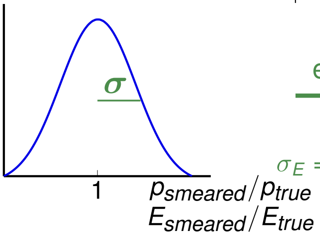
	CMS-like	ALEPH-like	ATLAS-like
σ (Tracker)	1.3%	1%	1%
σ (EMCAL)	$\frac{3\%}{\sqrt{E}} \oplus \frac{12\%}{E} \oplus 0.3\%$	$\frac{18\%}{\sqrt{E}} \oplus 0.9\%$	$\frac{10\%}{\sqrt{E}} \oplus 0.17\%$
σ (HCAL)	$\frac{110\%}{\sqrt{E}} \oplus 9\%$	$\frac{85\%}{\sqrt{E}}$	$\frac{55\%}{\sqrt{E}} \oplus 6\%$



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^- →

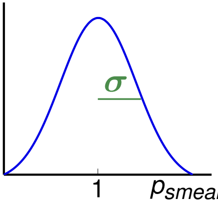
$\sigma_p = 1.3\%$

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

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)	$\frac{110\%}{\sqrt{E}} \oplus 9\%$	$\frac{85\%}{\sqrt{E}}$	$\frac{55\%}{\sqrt{E}} \oplus 6\%$



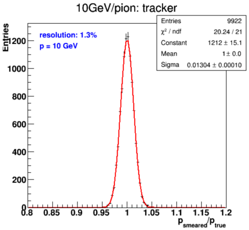
$p_{smeared}/p_{true}$

$E_{smeared}/E_{true}$

e.g. 10 GeV π^-

$\sigma_p = 1.3\%$

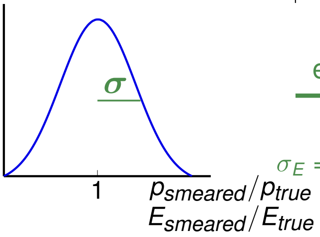
$\sigma_E = \frac{110\%}{\sqrt{E}} \oplus 9\% \approx 35\%$



Example 1: models

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

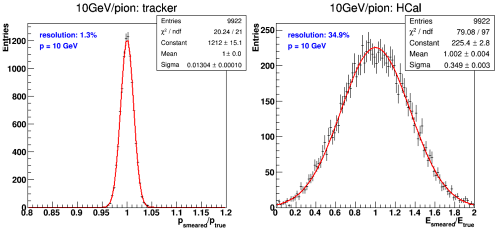
	CMS-like	ALEPH-like	ATLAS-like
σ (Tracker)	1.3%	1%	1%
σ (EMCAL)	$\frac{3\%}{\sqrt{E}} \oplus \frac{12\%}{E} \oplus 0.3\%$	$\frac{18\%}{\sqrt{E}} \oplus 0.9\%$	$\frac{10\%}{\sqrt{E}} \oplus 0.17\%$
σ (HCAL)	$\frac{110\%}{\sqrt{E}} \oplus 9\%$	$\frac{85\%}{\sqrt{E}}$	$\frac{55\%}{\sqrt{E}} \oplus 6\%$



e.g. 10 GeV π^-

$\sigma_p = 1.3\%$

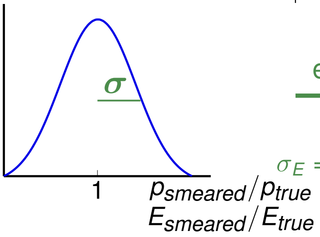
$\sigma_E = \frac{110\%}{\sqrt{E}} \oplus 9\% \approx 35\%$



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)	$\frac{110\%}{\sqrt{E}} \oplus 9\%$	$\frac{85\%}{\sqrt{E}}$	$\frac{55\%}{\sqrt{E}} \oplus 6\%$

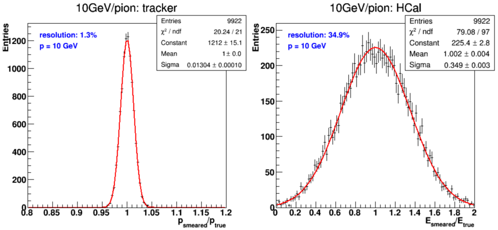


e.g. 10 GeV π^-

$\sigma_p = 1.3\%$

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

INPUT

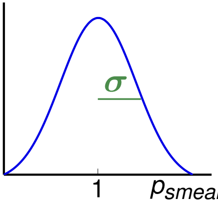


Example 1: models

`examples/extended/parameterisations/Par02`

- 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)	$\frac{110\%}{\sqrt{E}} \oplus 9\%$	$\frac{85\%}{\sqrt{E}}$	$\frac{55\%}{\sqrt{E}} \oplus 6\%$

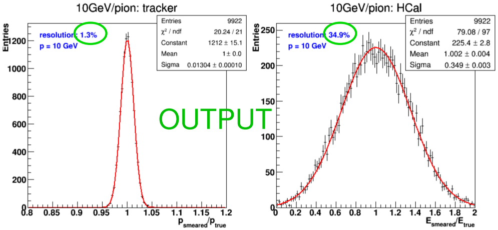


e.g. 10 GeV π^-

$\sigma_p = 1.3\%$

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

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() );
}
```

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() );
}
```

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 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 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 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

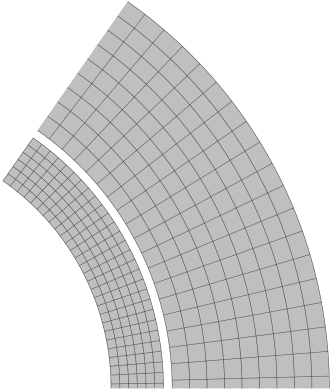
```
void Par02FastSimModelEMCal::DoIt( const G4FastTrack& aFastTrack, G4FastStep& aFastStep ) {
  aFastStep.KillPrimaryTrack();
  aFastStep.ProposePrimaryTrackPathLength( 0.0 );
  G4double Edep = aFastTrack.GetPrimaryTrack()->GetKineticEnergy();
  G4double Esm; [...] // Esm = smeared Edep
  aFastStep.ProposeTotalEnergyDeposited(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

Example 2:

`examples/extended/parameterisations/Par02`

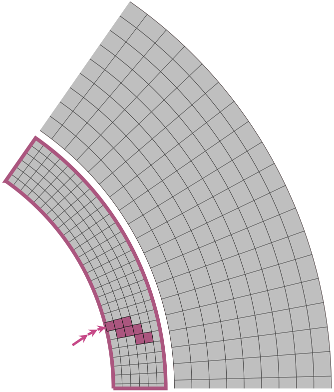
Example 2

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



Example 2

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

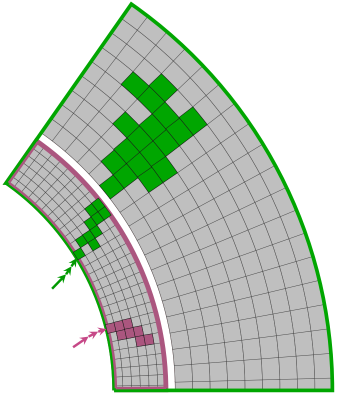


Par01EMShowerModel.cc

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

Example 2

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



`Par01EMShowerModel.cc`

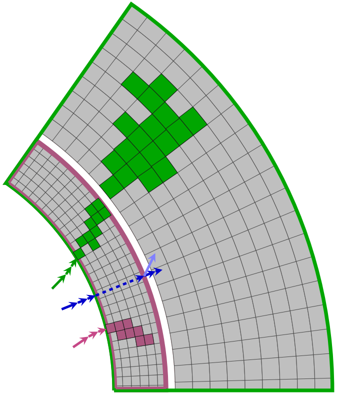
- electrons and photons
- electromagnetic calorimeter, envelope in mass geometry

`Par01PionShowerModel.cc`

- pions
- both calorimeters: envelope around EMCal and HCal ⇒ parallel geometry

Example 2

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



`Par01EMShowerModel.cc`

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

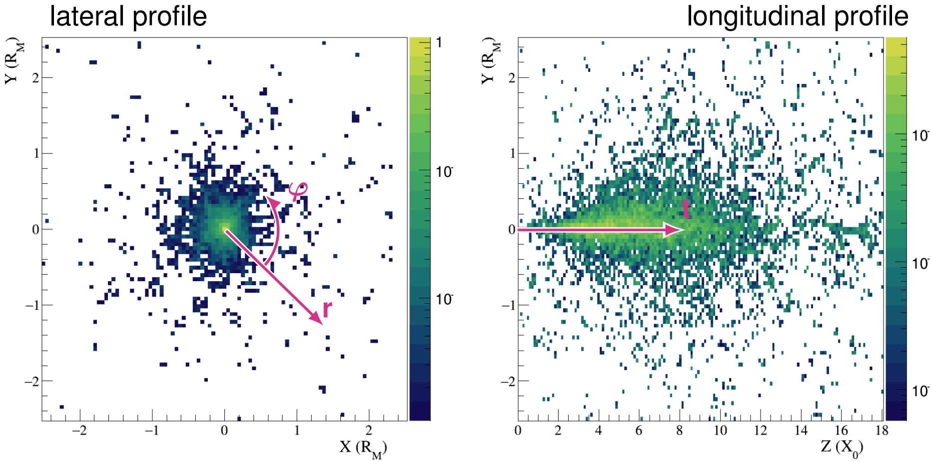
`Par01PionShowerModel.cc`

- pions
- both calorimeters: envelope around EMCal and HCal ⇒ parallel geometry

`Par01PiModel.cc`

- create secondaries

Shower profiles



Example 2 – models

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

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 $f(\varphi)$

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 $f(\varphi)$

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

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 $f(\varphi)$

$$f(\varphi) = \frac{1}{2\pi}, \qquad f(t; k, \theta) = \frac{x^{k-1} e^{-\frac{x}{\theta}}}{\theta^k \Gamma(k)}$$

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 $f(\varphi)$

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

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 $f(\varphi)$

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

4. deposit energy $\Delta E = \frac{E}{N}$ in $N = 100$ points

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 $f(\varphi)$

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

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

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();
}
```

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]);
  }
}
```

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

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 $f(r, 0, 10\text{cm})$

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

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

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

Example 2 – models

GEANT4 A SIMULATION TOOLKIT

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 $f(\varphi)$

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 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 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 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 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 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 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 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/gflash

Example 3

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

Example 3

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

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

Example 3

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

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

- flat distribution in azimuthal angle $f(\varphi) = \frac{1}{2\pi}$

Example 3

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

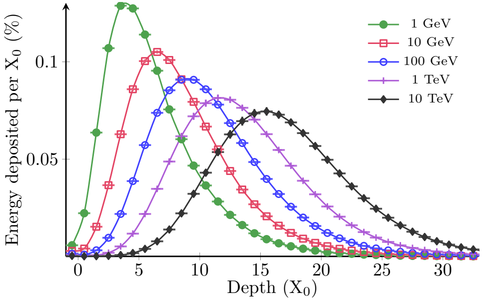
Example 3

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

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

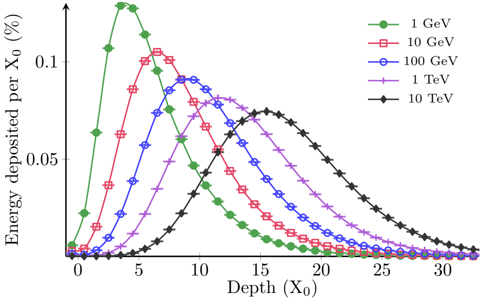
- flat distribution in azimuthal angle $f(\varphi) = \frac{1}{2\pi}$
- $f(t)$ and $f(r)$ parametrised as a function of particle's energy (E) and medium (Z)
- t and r are expressed in units of X_0 and R_M

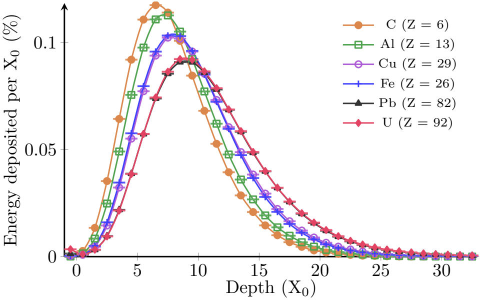
Example 3 - longitudinal profile



$$T \sim \ln E$$

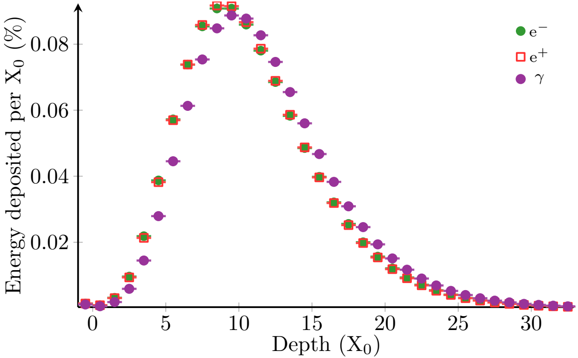
Example 3 - longitudinal profile





$T \sim \ln E$

Example 3 - longitudinal profile



Example 3 - longitudinal profile

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

Example 3 - 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 $T = \frac{\alpha-1}{\beta}$

Example 3 - 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 $y = \frac{E}{E_c}$:

$$T = \ln y + l_1$$
$$\alpha = l_2 + (l_3 + \frac{l_4}{Z}) \ln y$$

Example 3 - longitudinal profile

GEANT4 A SIMULATION TOOLKIT

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

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

$$T = \ln y + \boldsymbol{l_1}$$
$$\alpha = \boldsymbol{l_2} + (\boldsymbol{l_3} + \frac{\boldsymbol{l_4}}{Z}) \ln y$$

Example 3 - 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 $y = \frac{E}{E_c}$:

$$T = \ln y + I_1$$
$$\alpha = I_2 + (I_3 + \frac{I_4}{Z}) \ln y$$

A.1 Homogeneous Media

A.1.1 Average longitudinal profiles

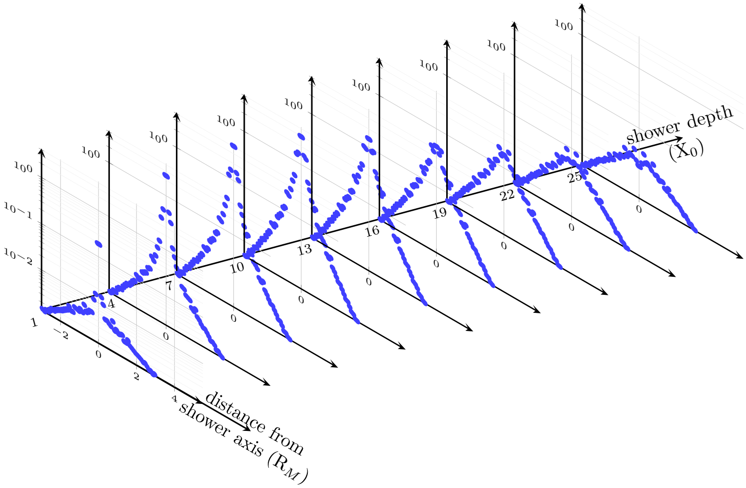
$$T_{hom} = \ln y - 0.858$$
$$\alpha_{hom} = 0.21 + (0.492 + 2.38/Z) \ln y$$

A.1.2 Fluctuated longitudinal profiles

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

arXiv:hep-ex/0001020

Example 3 – lateral profile



Example 3 – lateral profile

$$f(r) = \left\langle \frac{1}{dE(t)} \frac{dE(t,r)}{dr} \right\rangle$$

Example 3 – lateral profile

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

Example 3 – lateral profile

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

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

Example 3 – lateral profile

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

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

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

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

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

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

Example 3 – 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 $\tau = \frac{t}{T}$:

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

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

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

Example 3 – lateral profile

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

$$= p \frac{2rR^2_{\mathrm{core}}}{(r^2 + R^2_{\mathrm{core}})^2} + (1-p) \frac{2rR^2_{\mathrm{tail}}}{(r^2 + R^2_{\mathrm{tail}})^2}$$

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 3 – model

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 3 – model

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*          pbW04 = nistManager->FindOrBuildMaterial("G4_PbW04");
  // -- fast simulation models:
  // **********************************************
  // * Initializing shower modell
  // **********************************************
  G4cout << "Creating shower parameterization models" << G4endl;
  fFastShowerModel = new GFlashShowerModel("fFastShowerModel", fRegion);
  fParameterisation = new GFlashHomoShowerParameterisation(pbW04);
  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 3 – model

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 3 – model

examples/extended/parameterisations/gflash

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*          pbW04 = nistManager->FindOrBuildMaterial("G4_PbWO4");
  // -- fast simulation models:
  // **********************************************
  // * Initializing shower modell
  // ***********************************************
  G4cout << "Creating shower parameterization models" << G4endl;
  fFastShowerModel = new GFlashShowerModel("fFastShowerModel", fRegion);
  fParameterisation = new GFlashHomoShowerParameterisation(pbW04);
  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 3 – model

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 3 – model

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 3 – model

ExGflashSensitiveDetector.hh

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

Example 3 – model

ExGflashSensitiveDetector.hh

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

Example 3 – model

ExGflashSensitiveDetector.hh

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

Example 3 – model

ExGflashSensitiveDetector.hh

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

Sampling calorimeter

GVFlashHomoShowerTuning can be used to used tuned parameters (l_1, l_2, ... , r_1, r_2, ...)

Example 3 – model

ExGflashSensitiveDetector.hh

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

Sampling calorimeter

GVFlashHomoShowerTuning can be used to used tuned parameters (l_1, l_2, ... , r_1, r_2, ...)

For simulation in sampling detectors use GFlashSamplingShowerParameterisation and GFlashSamplingShowerTuning

Summary

- Fast simulation can be used within Geant4;

Summary

- Fast simulation can be used within Geant4;
- Hooks to take over control in chosen volumes/for chosen particles;

Summary

- Fast simulation can be used within Geant4;
- Hooks to take over control in chosen volumes/for chosen particles;
- Seamless mix of detaild and parametric simulation;

Summary

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

Summary

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

Questions/problems? anna.zaborowska@cern.ch