Migration of low-energy physics to common EM design

Luciano Pandola
INFN – LNGS

on behalf of the LowE EM working group

Integration of the EM physics models available in Geant4

- Historically, two sets of Electromagnetic models available in Geant4, developed by two different working groups
  - Standard models: HEP and other high energy applications
  - Low-energy models: medical, space and astroparticle applications

- **Explicit request** in the Geant4 Review to improve the **integration** of the EM sets of models, also to allow for the use of different models in different energy ranges

Recommendation 5: We recommend integrating the two EM models into a single package, similar to what exists in hadronic models. This will allow a user to choose one model in one energy range and the other model in a different energy range in order to optimize physics and computing performance for his application.

Objectives and motivations - 1

- Build a coherent aproach of EM interactions in Geant4, allowing for a multi-model approach for the same process (namely, more models for a given process, in different energy ranges)
- Make physics more transparent
  - retrieve cross sections, mean free paths and stopping powers
  - give the users the possibility to use the already-existing general purpose tools (e.g. G4EmCalculator)
  - easier to create new models, also allowing for cross section biasing

Objectives and motivations - 2

- Improve CPU performances
- *By-product*: a few existing bugs for LowEnergy processes automatically fixed
- Better coordination and reduction of duplication:
  - common physics lists, where the best models for low and high energies are used
  - common software design and validation effort
  - cross references between Standard EM and Low Energy EM web pages

From Geant4 9.3 BETA

Software design - 1

- All Low Energy Electromagnetic classes migrated to adopt the software design proposed by the Standard EM working group
- A physical interaction or process is described by a process class (inherits from **G4VEmProcess or G4VEnergyLossProcess**)
  - Naming scheme : « G4ProcessName »
  - Eg. : « G4ComptonScattering » for photon Compton scattering
  - Processes **already existing**: default models are the Standard ones (e.g. G4KleinNishinaCompton for G4ComptonScattering)
  - Same approach as for hadronic models (although it was not possible to use the very same design)

From Geant4 9.3 BETA

Software design - 2

- A physical process can be simulated according to several models, each model being described by a model class (inherits from `G4VEmModel`)
  - Naming scheme : « G4ModelNameProcessNameModel »
  - Eg. : « G4LivermoreComptonModel » for the Livermore Compton model
  - `myProcess->SetModel(myModel);` use a single model
  - `myProcess->AddEmModel(order,myModel);` use **multiple** models
    - default model is always instantiated (with least priority). During the tracking it is used the model with **highest priority** among those applicable at that energy
  - Models can be alternative and/or complementary on certain energy ranges

From Geant4 9.3 BETA

Software design - 3

- Model classes take care to calculate:
  - total cross section (cross section above threshold, for continuous processes)
  - stopping power
  - final state (kinematics, production of secondaries…)

- Old-style Low Energy processes (e.g. `G4LowEnergyCompton`) presently kept for backwards-compatibility but they're **not** going to be **maintained**, corrected or updated
  - warning printed on the screen

The G4VEmModel interface

- Models have to implement the virtual methods:
  - `Initialize()` → initialize, clean up tables, recalculate, etc.
  - `ComputeCrossSectionPerAtom()` [= XS], or `CrossSectionPerVolume()` [= XS*volume density] to build the mean free path table (given the threshold, for continuous processes)
  - `SampleSecondaries()` → update primary particle and/or generate secondary particles in the final state
- For models describing **continuous** processes:
  - `ComputeDEDXPerVolume()` → calculates stopping power (Energy/Length)
  - `SampleDeexcitationAlongStep()` *[optional]* → produces secondary particles AlongStep

Migrated models

- Penelope ✓
  - Compton, Gamma Conversion, Rayleigh, Photoelectric (γ-rays)
  - Ionisation, Bremsstrahlung ($e^{\pm}$), Annihilation
- Livermore ✓
  - Compton, Gamma Conversion, Rayleigh, Photoelectric (γ-rays)
  - Ionisation, Bremsstrahlung (e^{-} only)
- Livermore Polarized ✓
  - Compton, Rayleigh (γ-rays)
- Ions
  - Ion parametrized loss
  - G4hLowEnergyIonisation **not** migrated, buggy and *de facto* obsolete
- DNA ✓
  - Ionisation, Excitation, Elastic, Charge Exchange

Low-E models for γ-rays

	Process	**Models**
Compton scattering	G4ComptonScattering	G4KleinNishinaCompton (*) G4PenelopeComptonModel G4LivermoreComptonModel G4LivermorePolarizedComptonModel
Photoelectric effect	G4PhotoElectricEffect	G4PEEffectModel (*) G4PenelopePhotoElectricModel G4LivermorePhotoElectricModel
Gamma conversion	G4GammaConversion	G4BetheHeitlerModel (*) G4PenelopeGammaConversionModel G4LivermoreGammaConversionModel
Rayleigh scattering	G4Rayleigh	G4LivermoreRayleighModel (*) G4PenelopeRayleighModel G4LivermorePolarizedRayleighModel

* = default

Default models in blue are from the **Standard package**

Low-E models for $e^{\pm}$

	Process	**Models**
Ionisation	G4eIonisation	G4MollerBhabhaModel (*) G4PenelopeIonisationModel G4LivermoreIonisationModel (e^- only)
Bremsstrahlung	G4eBremsstrahlung	G4eBremsstrahlungModel (< 1 GeV) (*) G4eBremsstrahlungRelModel (> 1 GeV) (*) G4PenelopeBremsstrahlungModel G4LivermoreBremsstrahlungModel (e^- only)
e^+ Annihilation	G4eplusAnnihilation	G4eeToTwoGammaModel (*) G4PenelopeAnnihilationModel

Model to describe **fluctuations** in E_{loss} is always G4UniversalFluctuation

* = default

Default models in blue are from the **Standard package**

(more models available in the Standard package)

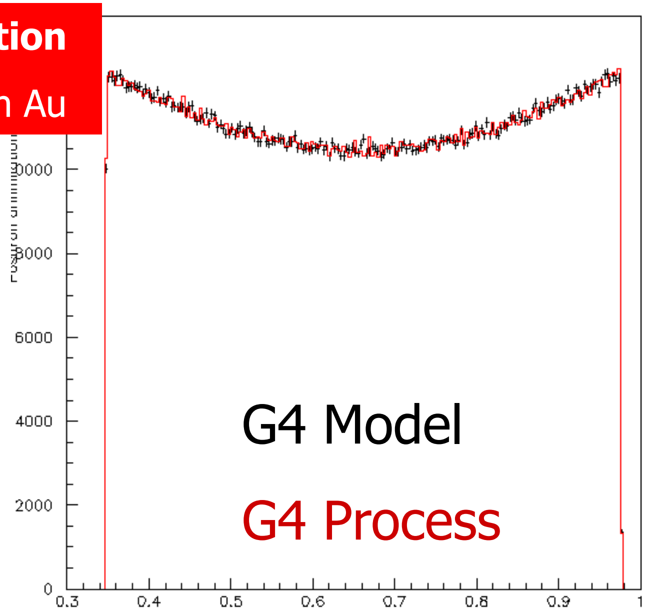
Verification of migration - 1

For the newly implemented models, it has been verified that **results** (cross section, stopping power and final state) are the **same** than the previous Low Energy models

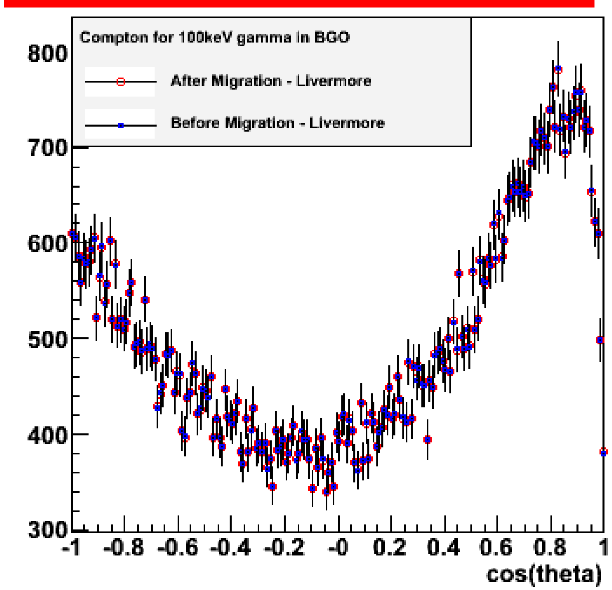
→ **check** that no (evident) **bugs** are introduced in the migration!

e+ annihilation

300 keV e+ in Au



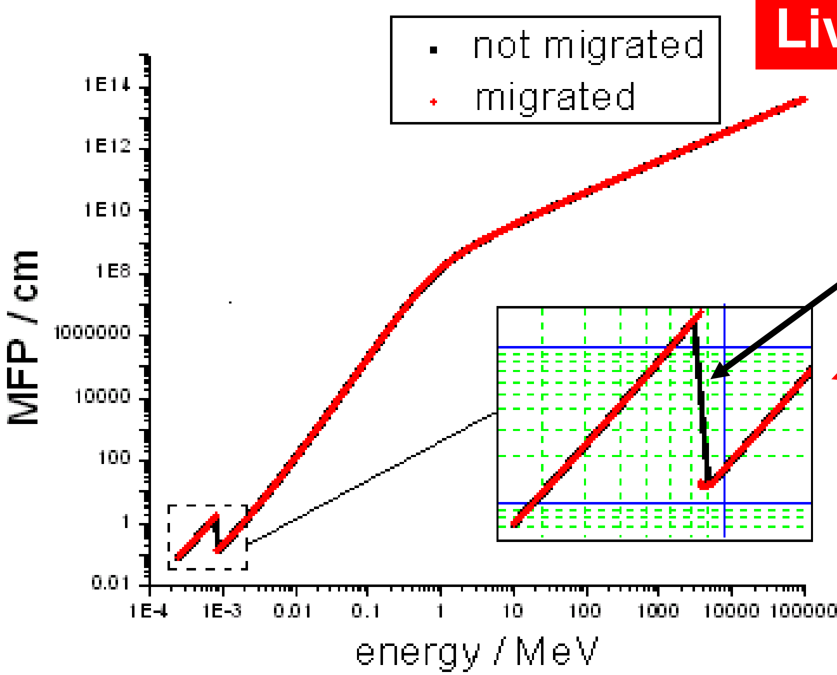
Compton 100 keV in BGO



Verification of migration - 2

- Systematic testing of migrated Livermore, Penelope and Geant4-DNA models is performed on a regular basis for several particles, energies and materials by independent teams (CENBG, INFN, IPHC, JLab)

Livermore photoelectric on Ne

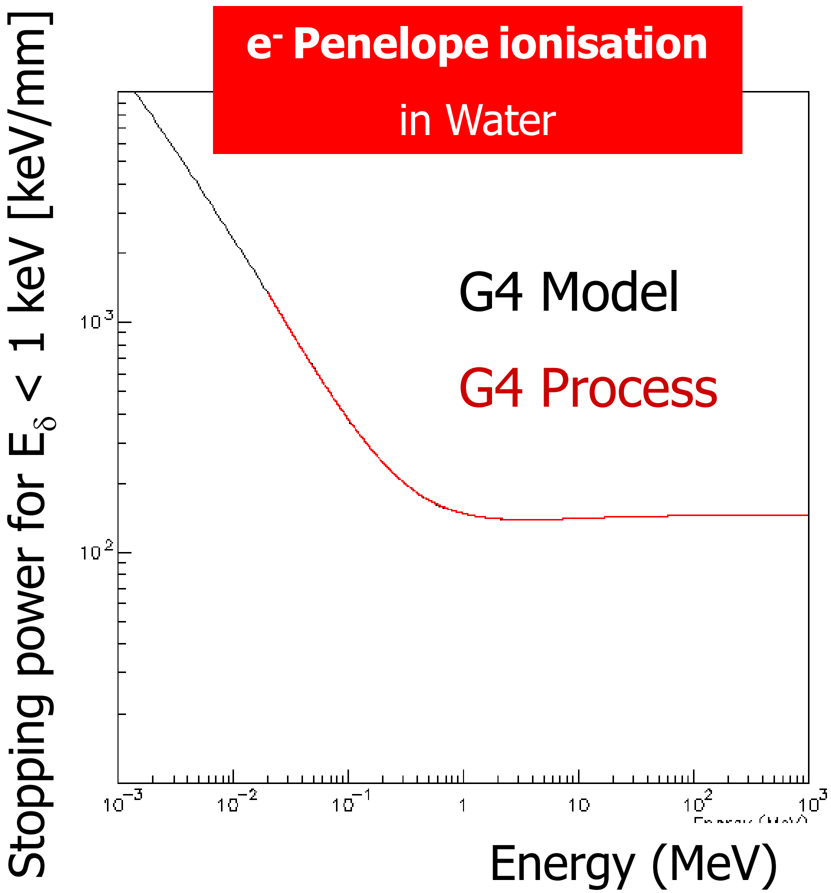


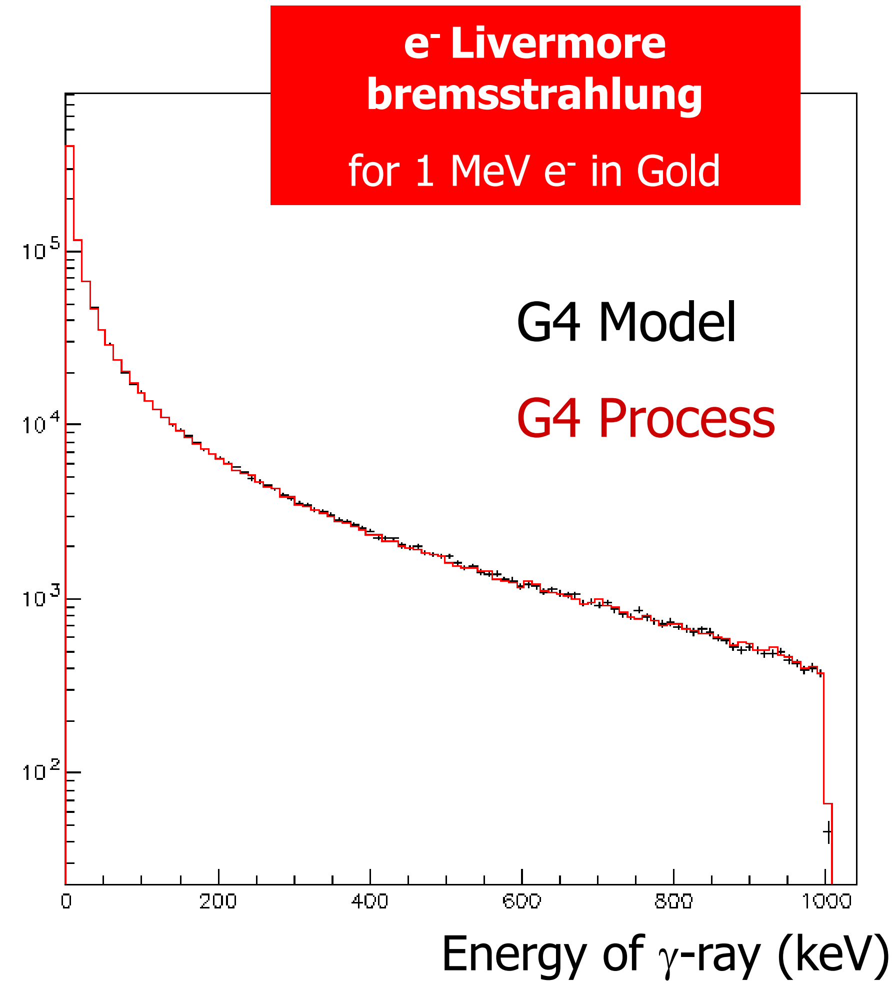
MeanFreePath table built by G4LowEnergyPhotoElectric (**200** bins)

MFP by **ComputeMeanFreePath()** method of the model (arbitrary fine grid)

The process builds (defaults) the MFP table with **77 points** (*220 points* in `G4EmPenelopePhysicsList`)

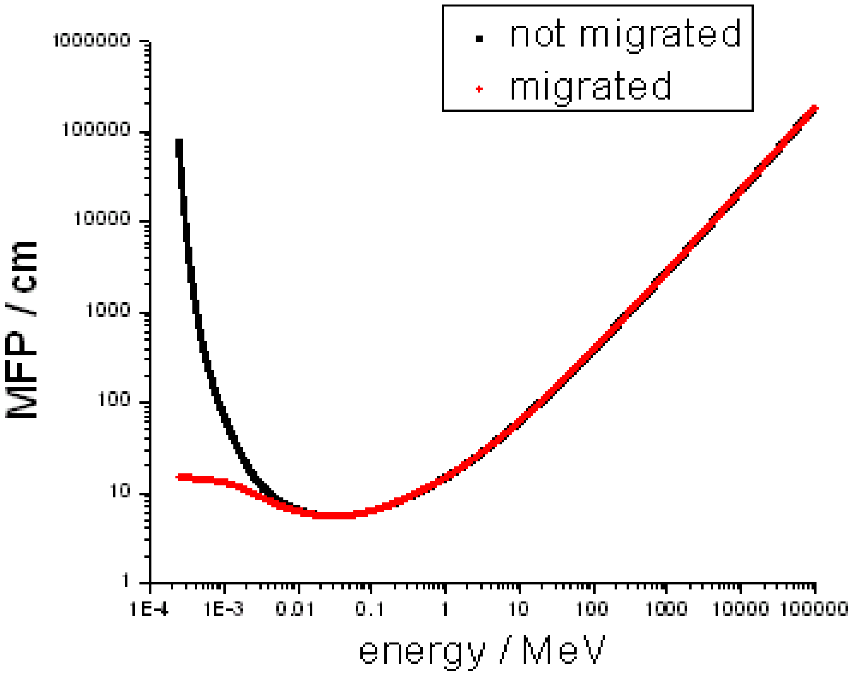
Verification of migration - 3





Verification of migration - 4

Penelope Compton on Water



Migration was also a good chance to check some parts of the code and make **improvements**

Found one difference between G4Penelope and the original FORTRAN (Compton cross section < 10 keV) → irrelevant for most applications. It has been corrected in the model

From Geant4 9.3 BETA

Extract Physics

- Thanks to this software design, it is possible to use retrieve Physics quantities with a G4EmCalculator object **also** for the **Low-Energy models**, similarly to what was possible for the Standard ones.
- Alternatively, one can invoke directly the methods from the model to dump physics quantities, e.g.

```
G4LivermoreBremsstrahlungModel* brem = new
    G4LivermoreBremsstrahlungModel();
G4DataVector dummy;
brem->Initialise(G4Electron::ElectronDefinition,dummy);
G4double xs1 =
 brem->CrossSectionPerVolume(material,electron,energy,tCut);
G4double sp1 =
 brem->ComputeDEDXPerVolume(material,electron,energy,tCut);
```

New physics lists

- New EM physics list builders to include Livermore, Polarized, DNA or Penelope models
  - G4EmPenelopePhysics, G4EmLivermorePhysics, G4EmLivermorePolarizedPhysics, G4EmDNAPhysics
- Use Low-Energy models **up to a certain energy** E_0 (e.g. 1 GeV for Penelope) and Standard models (default) for $E>E_0$
  - Performance optimization (meet Review **Recommendation**)

```
G4double PenelopeHighEnergyLimit = 1.0*GeV;
G4PhotoElectricEffect* thePhotoElectricEffect = new
  G4PhotoElectricEffect();
G4PenelopePhotoElectricModel* thePEPenelopeModel = new
       G4PenelopePhotoElectricModel();
thePEPenelopeModel
   ->SetHighEnergyLimit(PenelopeHighEnergyLimit);
thePhotoElectricEffect->AddEmModel(0,thePEPenelopeModel);
pmanager->AddDiscreteProcess(thePhotoElectricEffect);
```

Penelope model has **top priority** over its own applicability energy range

Conclusions

- Physics EM models developed in the Low Energy package have been **migrated** to the same design adopted for Standard models
  - Meets Recommendation from the G4 Review
  - Allows a multi-model approach for the same process (performance optimization)
  - Better integration, cross-check, more transparent to physics
- Extensively tested to mitigate the risk of introducing physics **bugs** during the migration
  - Old-style Low Energy process will be kept for compatibility, but not maintained/upgraded
- Created physics lists using Low Energy models (plus Standard ones, at higher energy)