Translation of Penelope 2018 to C++ and its interface to Geant4

Makoto Asai (SLAC)
Miguel A. Cortés-Giraldo (U. Sevilla)
Vicent Giménez Alventosa (U. Poly. Valencia)
Vicent Giménez Goméz (U. Valencia)
Francesc Salvat (U. Barcelona)

UNIVERSITAT DE BARCELONA
VNIVERSITAT DE VALÈNCIA
UNIVERSITAT POLITÈCNICA DE VALÈNCIA
UNIVERSIDAD DE SEVILLA
SLAC NATIONAL ACCELERATOR LABORATORY
U.S. DEPARTMENT OF ENERGY
Office of Science

Disclaimer

- This particular talk does not represent any part of the Geant4 Collaboration. I'm not speaking on behalf of any part of the Geant4 Collaboration.
- The outcoming code and documents of this work will not be a part of Geant4 release, but will be distributed separately.

Penelope 2018

- The Fortran code system Penelope performs Monte Carlo simulation of coupled electron-photon transport in arbitrary materials for a wide energy range (~1 GeV – 50 eV). Penelope implements the most reliable interaction models that are currently available, limited only by the required generality, and gives results in good agreement with a variety of experimental data.
- We present a translation of the latest Penelope 2018 physics routines to C++ that is designed to be used with the Geant4 toolkit with multithreading capabilities.
  - It runs in both pure Penelope mode as well as in hybrid mode where Geant4 EM physics is used for higher energy region.
- This program effectively uses the available computational resources, giving results equivalent to those from the original Penelope programs with a shorter running time, which is roughly inversely proportional to number of threads used.

Penelope 2018

- Photon interactions
  - Photo-electric effect
  - Coherent (Rayleigh) scattering
  - Incoherent (Compton) scattering
  - Electron-positron pair production
  - Polarized scatterings
- Electron and positron interactions
  - Elastic collision
  - Inelastic collision
  - Bremsstrahlung emission
  - Positron annihilation
  - Multiple-scattering and soft-scattering / soft-energy-loss
- Details can be found in the Penelope Write-up

Two-step translation

Original Penelope 2018 (FORTRAN)

↓

Penelope 2018 (C)

↓

Penelope 2018 (C++) + Geant4 interface

Original Penelope code is kept maintained in FORTRAN.

Line-by-line hand translation to C.

Bit-by-bit comparison and verification with original FORTRAN code.

Line-by-line hand translation to C++ with thread-safety plus Geant4 interface classes.

Bit-by-bit comparison and verification with original FORTRAN code in sequential mode.

Statistical comparison and verification with original FORTRAN code in multithreaded mode.

main()

```
int main(int argc,char** argv)
{
 // Construct the run manager
 G4MTRunManager * runManager = new G4MTRunManager;
 runManager->SetNumberOfThreads(G4Threading::G4GetNumberOfCores());
 G4ScoringManager::GetScoringManager();

 // Set detector construction
 G4VUserDetectorConstruction* detector = new Tst1DetectorConstruction;
 runManager->SetUserInitialization(detector);

 // Set physics list
 G4VModularPhysicsList* physics = new FTFP_BERT;
 physics->RegisterPhysics(new PenelopeEMPhysics( 500.*keV ));
 runManager->SetUserInitialization(physics);
```

Use as many cores as you have

Threshold parameter

Material definition

```
void Tst1DetectorConstruction::DefineMaterials()
{
  auto nistManager = G4NistManager::Instance();
  auto air = nistManager->FindOrBuildMaterial("G4_AIR ");
  auto water = nistManager->FindOrBuildMaterial("G4_WATER")
  auto lead = nistManager->FindOrBuildMaterial("G4_Pb");

//***************************************
// Registering G4Material to Penelope interface
//***************************************

// Use material properties defined in G4Material
//  PeneInterface::GetInstance()->RegisterMaterial(air);
//  PeneInterface::GetInstance()->RegisterMaterial(water);
//  PeneInterface::GetInstance()->RegisterMaterial(lead);

// Use material properties defined in Penelope pre-defined material list
  PeneInterface::GetInstance()->RegisterMaterial(air, 104);
  PeneInterface::GetInstance()->RegisterMaterial(water, 278);
  PeneInterface::GetInstance()->RegisterMaterial(lead, 82);
}
```

Material properties are shared with Geant4 NIST material database

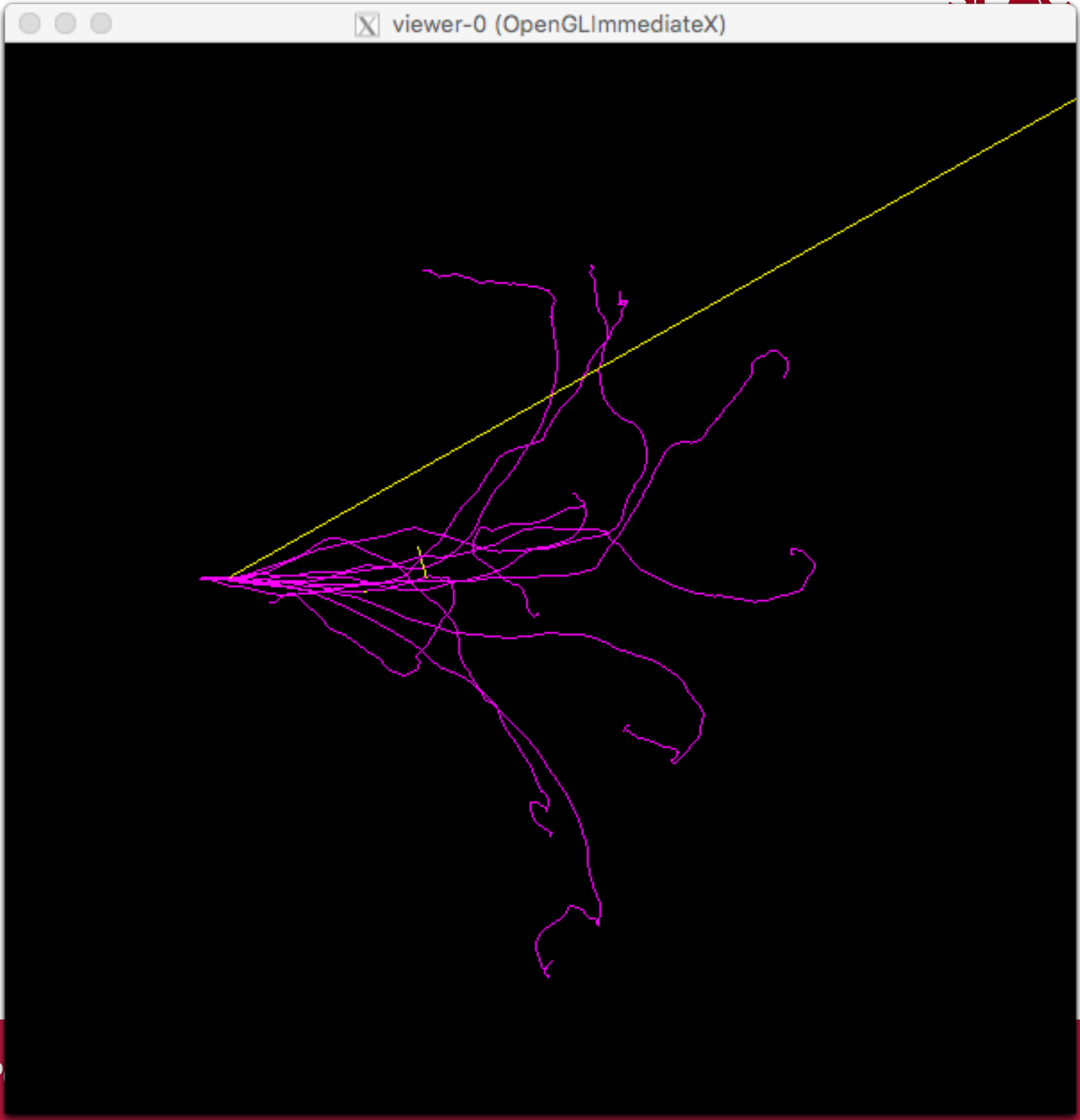
OR

Penelope's material properties are used for Penelope EM physics

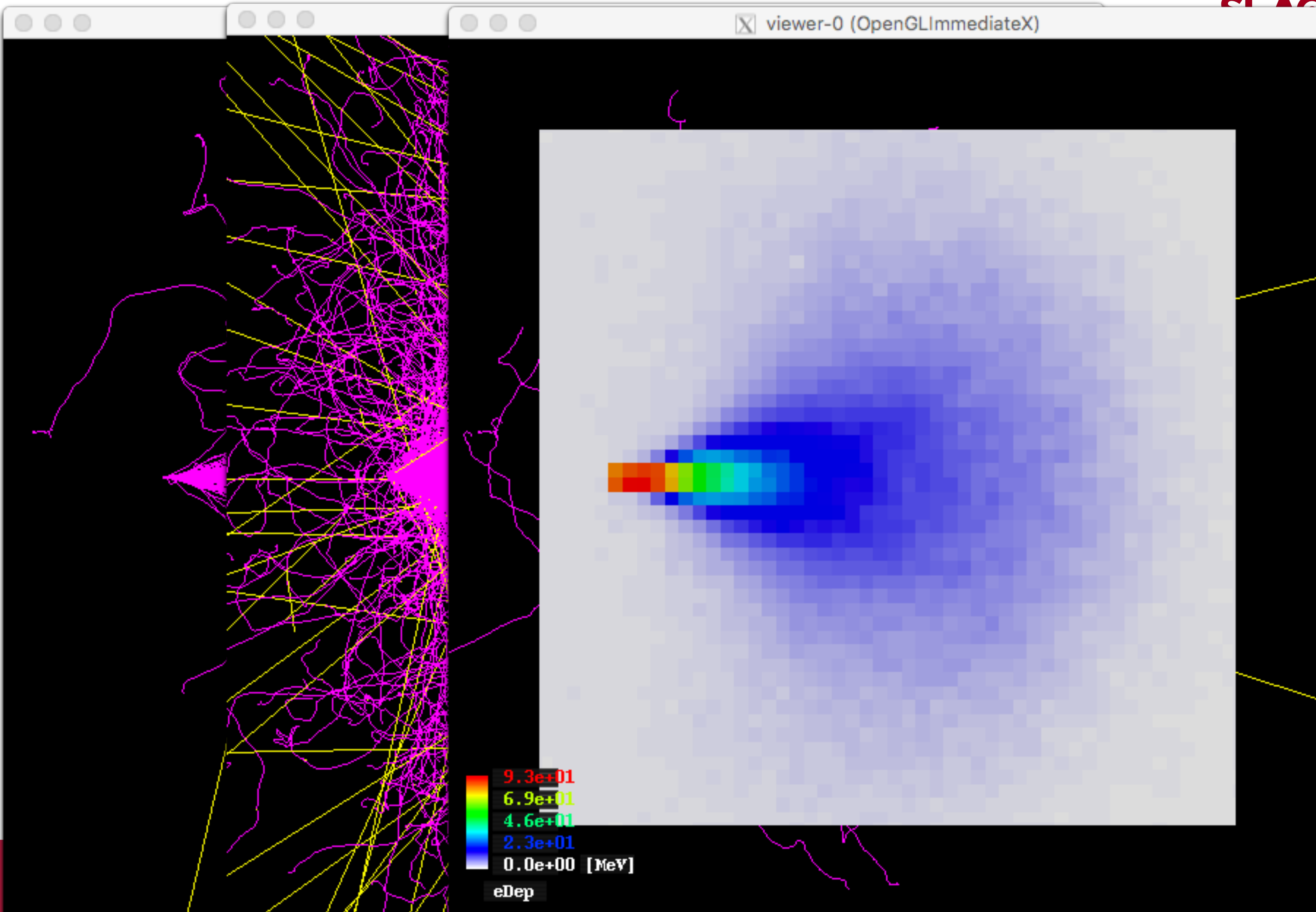
Penelope's material list can be found in [the Penelope Write-up](#)

Pure Penelope mode

- Penelope EM physics covers from ~ 1 GeV down to 50 eV.
- 10 electrons of 1 MeV in water.
  - Purple lines : e-
  - Yellow lines : γ



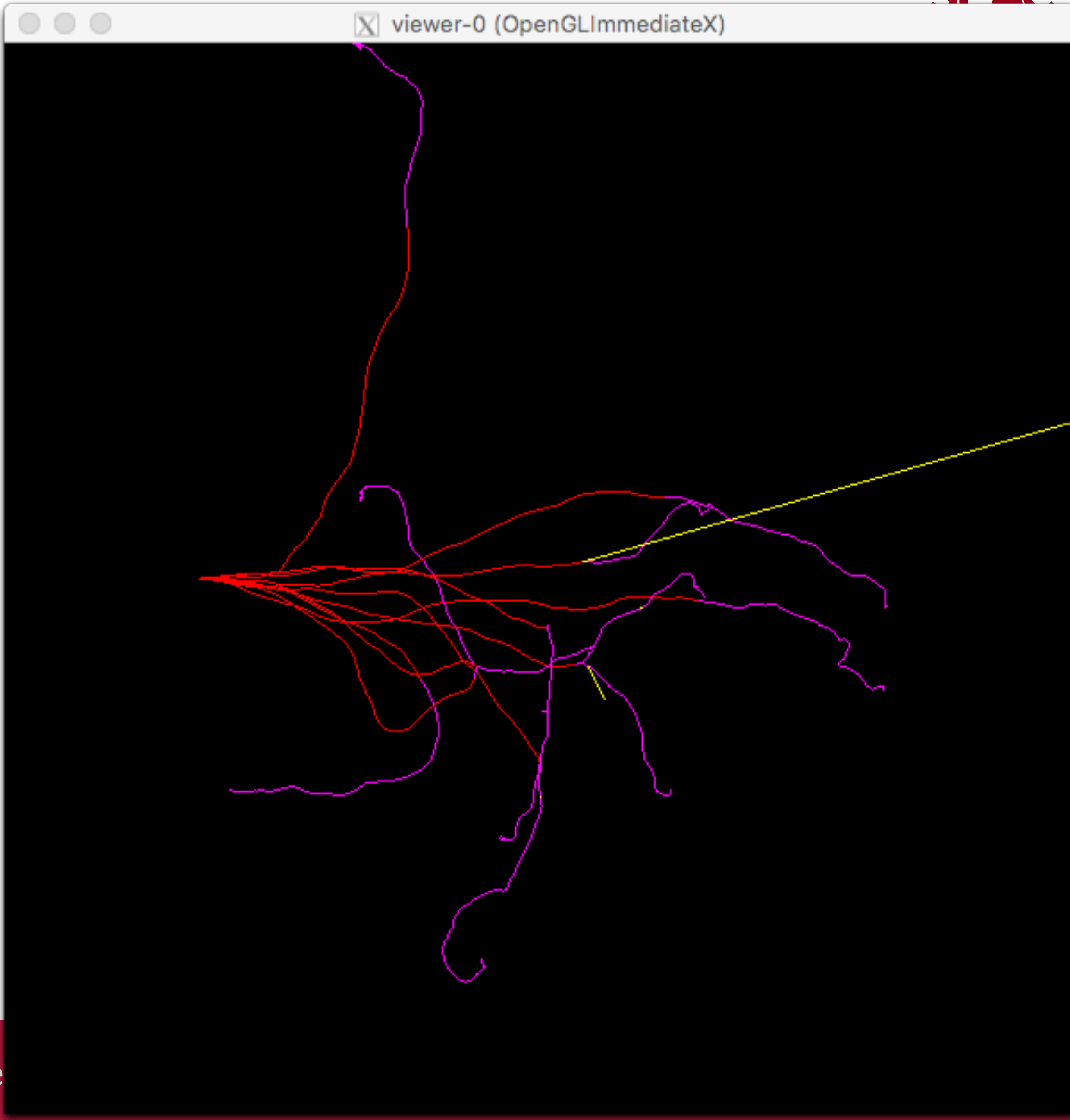
Pure Penelope mode



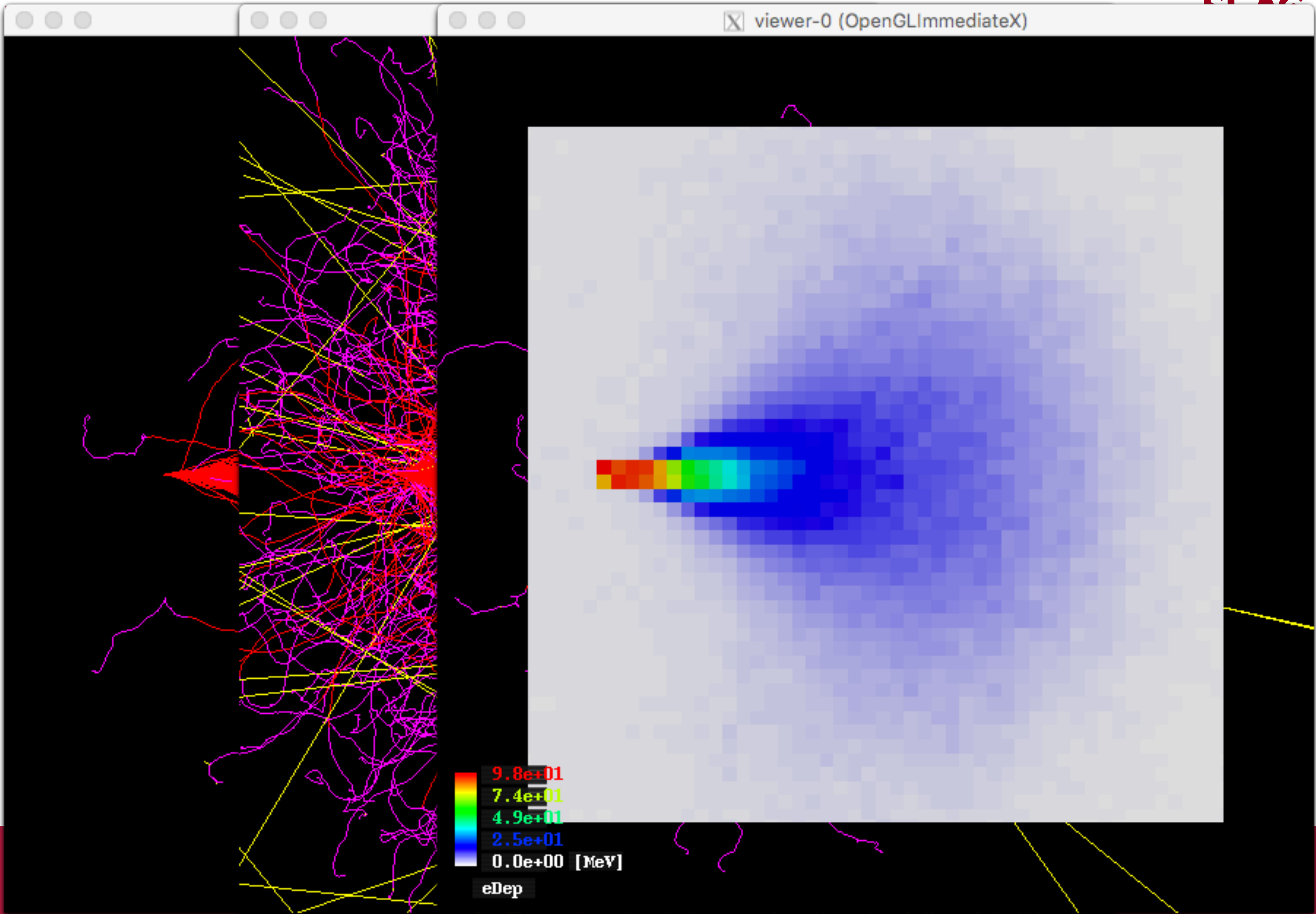
Hybrid mode

- Higher energy particle is taken care by the Geant4 physics (in this case Standard EM).
- Penelope EM physics takes over from a certain energy (in this case 500 keV) down to 50 eV.

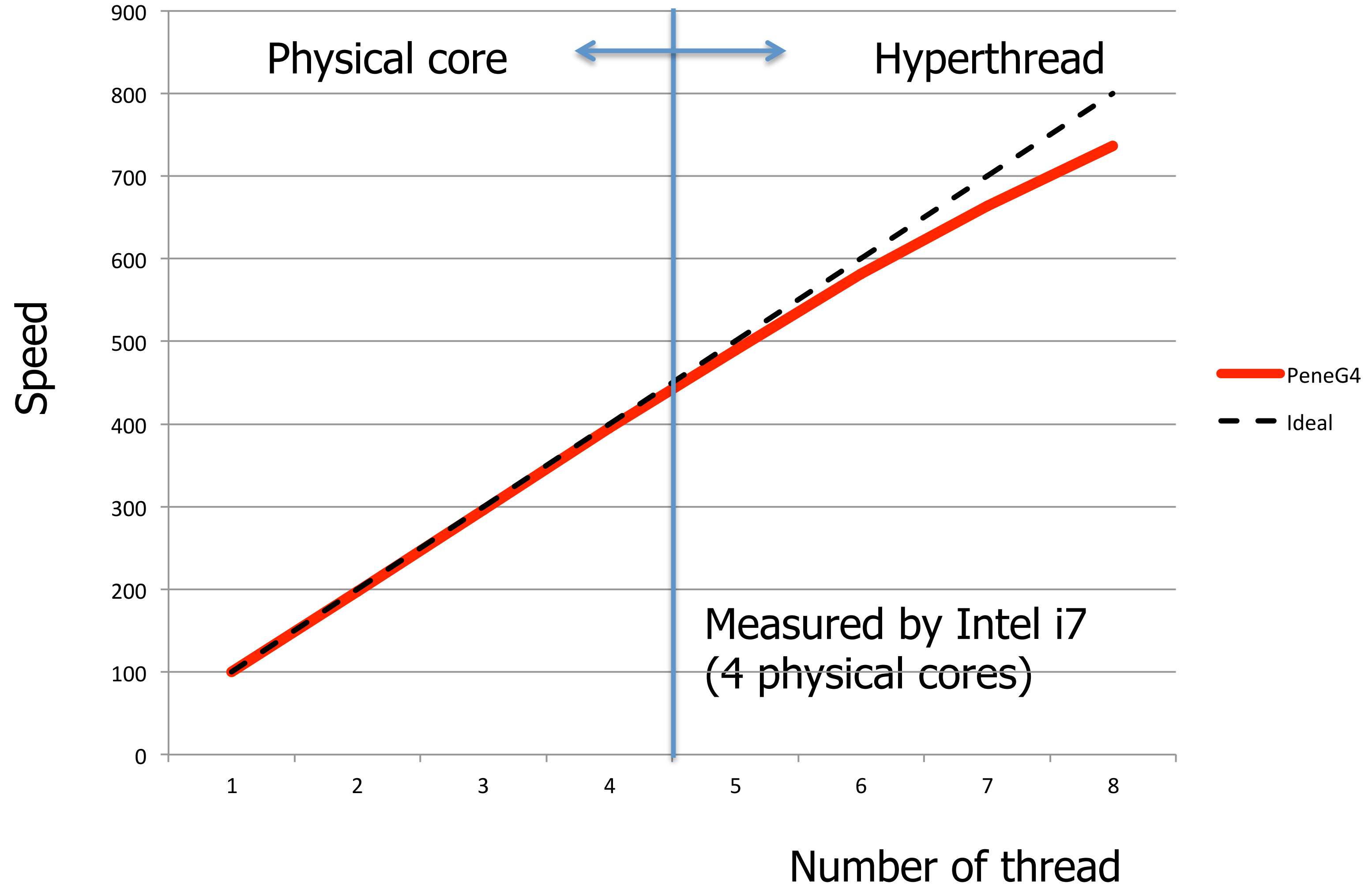
- 10 electrons of 1 MeV in water.
  - Red line : e- with G4 physics
  - Purple lines : e- with Penelope physics
  - Yellow lines : γ with Penelope physics



Hybrid mode


viewer-0 (OpenGLImmediateX)
9.8e+01
7.4e+01
4.9e+01
2.5e+01
0.0e+00 [MeV]
eDep

Linearity (preliminary)



Status and plan

- Large-scale benchmarking and minor code modifications / cleanups are ongoing.
- We are also working on the manuscript to be submitted to CPC.
- We plan to make the code and associated data files downloadable from NEA web site sometime in early/middle 2019.
- Significant C/C++ code rewriting for performance improvement may be considered in 2019 or later (based on budget / manpower).
- Extension to EM physics of proton (and maybe ions) is planned in 2019/2020.