HEAVY PARTICLES IN PHYSICS LISTS, CROSS SECTION SCALING

V. Ivanchenko CERN & Tomsk State University, Russia

03 August 2020

Outline

- Destruction end of job
- Initialization of nuclear level data
- New utilities for hadron physics configuration
- Hadronic parameters
- Variation of hadronic cross sections
- Summary

- Basic ideas or recent modification of Physics Lists:
  - *Reduce duplicated code in EM and hadronic configurations*
  - *More transparent configuration of models and cross sections*
  - *Optional addition of b-, c- hadrons*
  - *Cross section variation required for systematics studies*

Destruction end of job

- Destruction end of job was not working properly
  - *Some data members are thread local*
  - *Some was not deleted end of job*
- Difficulty of destruction of physics is in the fact, that cross section, model, and process classes may be shared between different particles in different ways for different Physics Lists
  - *This is strongly needed to reduce memory and CPU for initialization of physics*
- For 10.7beta the most part of physics is destructed now due to use of register/deregister mechanism
  - *Processes, models, and cross sections should be deleted by Physics List classes*
  - *The most part thread local variables in Physics Lists are removed*
- What is not done: some hadronic builders are thread local

Initialisation of nuclear level data

- Recently Gabriele identify a bug in MT mode in G4NuclearLevelData class
  - *In 10.5 we had only lazy initialization per isotope*
  - *In 10.6 lazy initialization of the data per isotope badly interacts with implementation of initialization of all needed isotopes before the run*
- The fix is implemented and already merged to the master and to the 10.6 patch branches
  - *In the new variant of initialization, in BuildPhysicsTable() method the G4ExcitationHandler calls download nuclear level data for all isotopes with Z<= Zmax*
    - Zmax is defined from the material list
  - *For Z> Zmax the lazy initialization remains*
  - *CPU penalty at initialization is about 1-2 seconds*
- Potentially we may decide reading all data at initialization
  - *Implemented in 10.6ref07*
  - *May be enabled/disabled in the Physics List*

New utilities for hadron physics configuration

- G4HadParticles – returns several lists of particle PDG codes
  - *std::vector<G4int>& G4HadParticles::GetKaons();*
  - *std::vector<G4int>& G4HadParticles::GetHyperons);*
  - *std::vector<G4int>& G4HadParticles::GetAntiHyperons();*
  - *…*
- G4HadProcesses – return pointers to hadronic processes per particle and allows adding extra cross section per particle
  - *G4HadronicProcess* G4HadProcesses::FindInelasticProcess(const G4String& partname);*
  - *G4HadronicProcess* G4HadProcesses::FindInelasticProcess(const G4String& partname);*
  - *G4bool G4HadronicProcesses::AddInelasticCrossSection(const G4ParticleDefinition*, G4VCrossSectionDataSet* my_xs);*
  - *……*
- G4HadronicBuilder – build standard set of models and cross sections for group of particles
  - *G4HadronicBuilder::BuildHyperonsFTFP_BERT();*
  - *G4HadronicBuilder::BuildBCHadronsFTFP_BERT();*
  - *G4HadronicBuilder::BuildHyperonsQGSP_BERT();*

Hadronic parameters

- Utilities in the previous slide are using G4HadronicParameters class
  - *User has a chance to change any parameter between instantiation of the PhysicsList and run initialization*
    - G4State_PreInit
    - Both C++ interface and UI commands (not for all) are available
    - Should we force, at least, a printout on each UI command?
- New parameters:
  - *G4bool EnableBCParticles()*
- Proposed extra parameters:
  - *G4double EnergyThresholdForHeavyParticles()*
    - The default is 1.1 GeV
    - If max energy is below, then no hyperons, anti-ions, b-, c- particle physics
    - We need to check if this bring some advantages to low-energy simulations

Variation of hadronic cross sections

- For study of systematic uncertainty due to simulation we may considered following approach:
  - *For hadronic models we propose to use different Physics Lists*
    - FTFP_BERT -> QGSP_BIC, FTFP_INCLXX, or QBBC
  - *For cross sections we may propose to use a factor to vary cross section value*
    - +- 5-10% would be within Geant4 accuracy
- Cross section factors are defined via G4HadronicParameters class:
  - *G4bool ApplyFactorXS() const; // false by default*
  - *G4double XSFactorNucleonInelastic() const ;*
  - *G4double XSFactorNucleonElastic() const ;*
  - *G4double XSFactorPionInelastic() const ;*
  - *G4double XSFactorPionElastic() const ;*
  - *G4double XSFactorHadronInelastic() const ;*
  - *G4double XSFactorHadronElastic() const ;*
  - *G4double XSFactorEM() const ;*
- User must change the flag and set corresponding factor via C++ interface

Summary

- Described developments together with development of Alberto for decay channels of b-, c- hadrons complete our plans to improve Physics List configurations for 2020
  - *We can offer c- and b- hadron physics*
  - *We can offer a method of Geant4 physics variation for study of systematics*
    - Until now is available for hadron elastic and QBBC inelastic
    - Should be propagated to other hadron inelastic constructors
  - *Bug fix and tuning of the approach are not excluded*
- Destruction of all EM and hadronic physics end of run is achieved
- Optimization of the initialization of nuclear level data structures is not yet finalized
  - *It is possible to upload all data begin of run*
  - *It is possible to upload only data for Z<Zmax*
  - *Recently Makoto proposed convert ASCII data files into binary compressed file(s)*