



*Geant4* ASSOCIATES  
INTERNATIONAL  
*Experts in Radiation Simulation*

**Geant 4**

# Hadronic Cross Sections

*V.Ivanchenko*

15th Geant4 Collaboration Workshop

*4-8 October 2010 Noordwijk, The Netherlands*

# Outline

- Hadr00 in 9.4
- Neutron cross sections
- Hadronic cross section re-design

# Hadr00 in 9.4beta

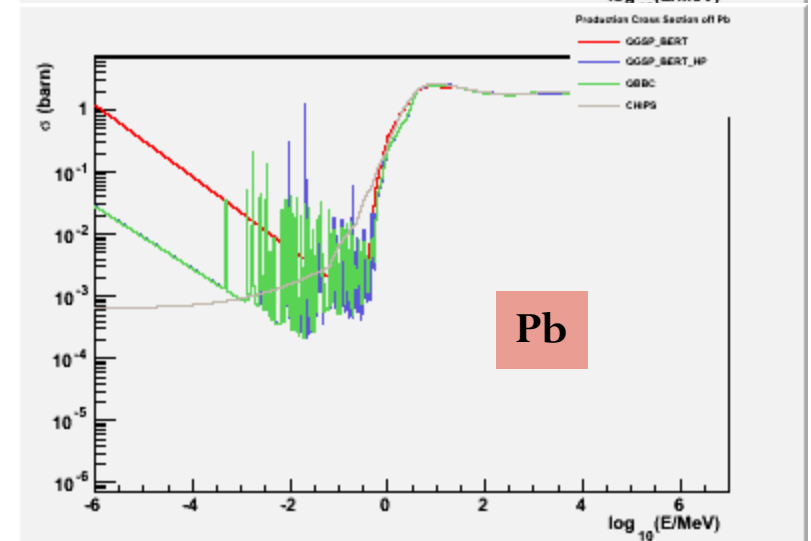
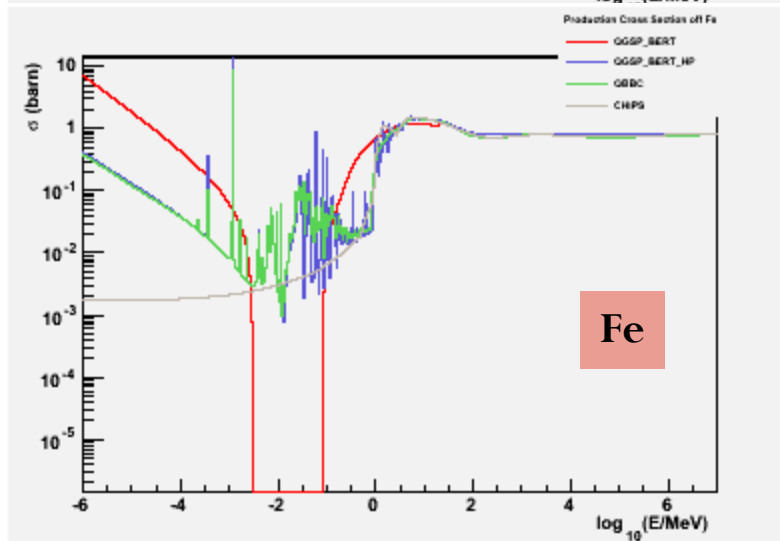
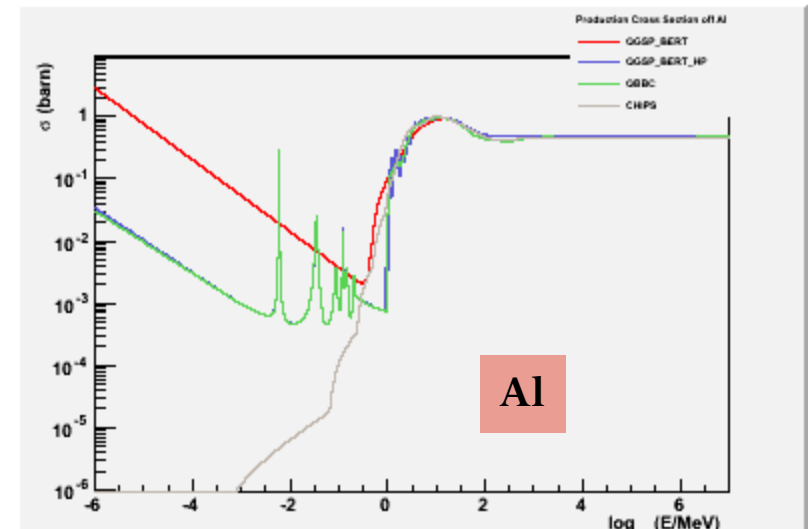
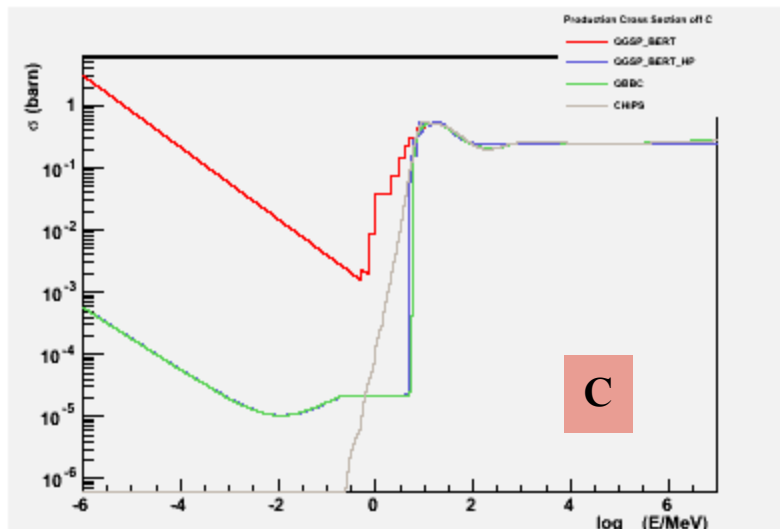
- Main example how to access cross sections
  - Inelastic, elastic, capture, fission, total x-sections
- Work for all reference Physics Lists except CHIPS, because G4QInelastic is not registered in G4HadronicProcessStore
  - Can be fixed by using more complicate code but should this be done in public example?

# Hadrcap test for neutron x-sections

## A.Ivanchenko

- Similar to Hadr00
- Provided elastic, inelastic, production, capture x-sections
- CHIPS elastic and production are available
- Script is working currently for 4 predefined reference Physics Lists for all elements
- In few minutes all neutron x-sections are available in form of plots

# Results for the production x-section (inelastic + capture) for 9.4beta



# How activate XS neutron inelastic and capture x-sections?

- Usage of HP or XS cross section may change shower shape
- The simplest method is to add extra physics constructor  
`physList->RegisterPhysics(new G4NeutronCrossSectionXS());`
- Why not include this in majority of Physics Lists?

# Hadronic cross section re-design

## G.Folger, V.Ivanchenko, D.Wright

- Number of concerns were expressed from time to time to Geant4 cross sections (incomplete list):
  - Too many classes in cross\_sections subdirectory
  - Function of classes are mixed
  - Not easy identify what class can be used in Physics List, what inside a model, what is combined class
  - Not possible to exchange internal cross section between models
  - Authorship problem
- During Dennis visit to CERN a discussion took place after which software requirement document was created and finally cross section re-design proposal was done, implemented and already path stt after 9.4beta
- **Simplest and backward compatible solution was chosen:**
  - Directory structure is not changed
  - G4VCrossSectionDataSet interface was extended
  - New G4VComponentCrossSection interface is added
  - It is agreed that cross\_section library should not depend on any model library
  - Model library can depend on cross\_section library

# Extension of G4VCrossSectionDataSet

- New methods are added:

```
inline G4int GetMinKineticEnergy() const;
```

```
inline void SetMinKineticEnergy(G4double value);
```

```
inline G4int GetMaxKineticEnergy() const;
```

```
inline void SetMaxKineticEnergy(G4double value);
```

```
inline const G4String& GetName() const;
```

- Any derived class can have original name and optimal interval of energy
- No other change introduced



# G4VComponentCrossSection interface

- Has following pure virtual methods:

```
virtual G4double GetTotalZandACrossSection(const G4DynamicParticle*, G4int /*Z*/, G4int /*N*/) = 0;  
virtual G4double ComputeTotalCrossSection(const G4DynamicParticle*, G4int /*Z*/, G4double /*N*/) = 0;  
virtual G4double GetInelasticZandACrossSection(const G4DynamicParticle*, G4int /*Z*/, G4int /*N*/) = 0;  
virtual G4double ComputeInelasticCrossSection(const G4DynamicParticle*, G4int /*Z*/, G4double /*N*/) = 0;  
virtual G4double GetElasticZandACrossSection(const G4DynamicParticle*, G4int /*Z*/, G4int /*N*/) = 0;  
virtual G4double ComputeElasticCrossSection(const G4DynamicParticle*, G4int /*Z*/, G4double /*N*/) = 0;  
virtual G4double ComputeQuasiElasticRatio(const G4DynamicParticle*, G4int /*Z*/, G4int /*N*/) = 0;  
virtual void BuildPhysicsTable(const G4ParticleDefinition&) = 0;  
virtual void DumpPhysicsTable(const G4ParticleDefinition&) = 0;
```

- Each derived class should have name **G4ComponentXXXXXX**
- Each derived class should have original name and interval energy where it is applicable
- These classes have no **IsApplicable** methods
  - There is no requirement that a component is capable to work for all isotope and all energy range, if it is used not properly it is recommended to issue a Hadronic Exception
- These cross section cannot be used directly in Physics List, they should be used in CrossSectionDataSet classes
- Derived class can implement internal hadron-nucleon cross section for a model or free hadron-nucleon cross section

# Migration plan and problems

- Backward compatibility is guaranteed by design
  - Interfaces path stt testing
- We encourage authors do migration now by creation of new component classes from existing ones
  - No change of results is expected
- Old classes will be kept until the migration is complete
  - The target – 9.4
- Open questions:
  - Generic interface to build a DataSet from Components is not yet designed
  - CHIPS cross sections sub-library dependence cannot be provided, so CHIPS x-sections may follow this design but cannot be used in cross\_sections library