Comments on hyper-nucleus nucleus integral cross section in the Glauber-Gribov approach

V. Grichine, V. Ivantchenko, F. Ribon, V. Uzhinsky

(CERN, RAS, Dubna)

Motivation & outline

- LHC experiments (ALICE) observe hyper-nuclei (& anti).
- First, the nucleus interface is extended to allow extra lambda-particles.
- Then, the cross-sections are corrected for additional contribution from lambda particles as well as small change of the hyper-nucleus radius.
- The Glauber-Gribov cross-section interface allows us to extend to the case of hyper-nucleus as shooting particle (lambda-nucleon XS, radius correction).
- The extension includes modifications in G4Nucleus, G4ComponenGGNuclNuclXsc and G4ComponentGGHadronNucleusXsc classes.
- Many modifications (new particles, interfaces) are available for 11.0 other (XS) will be in 11.1 releases.

New particles

- The following 6 light hyper-nuclei and corresponding anti-hyper-nuclei (in source/particles/hadrons/ions/) were added:

G4HyperTriton	(G4AntiHyperTriton)
G4HyperAlpha	(G4AntiHyperAlpha)
G4HyperH4	(G4AntiHyperH4)
G4HyperHe5	(G4AntiHyperHe5)
G4DoubleHyperDoubleNeutron	(G4AntiDoubleHyperDoubleNeutron)
G4DoubleHyperH4	(G4AntiDoubleHyperH4

G4Nucleus extension

- The new class member number of lambda-particles is added
  G4Nucleus(const G4double A, const G4double Z, const G4int numberOfLambdas=0);
  G4Nucleus(const G4int A, const G4int Z, const G4int numberOfLambdas = 0);
- New methods (are used in XS classes):

G4bool G4ParticleDefinition::IsHypernucleus() const;
G4int G4ParticleDefinition::GetNumberOfLambdasInHypernucleus() const;

G4ComponentGGNuclNuclXsc class

```
void G4ComponentGGNuclNuclXsc::ComputeCrossSections(
     const G4ParticleDefinition* aParticle, G4double kinEnergy,
    G4int Z, G4int A) { …
G4int pL = aParticle->GetNumberOfLambdasInHypernucleus();
G4bool pHN = aParticle->IsHypernucleus();
G4double cHN(0.88); // from XS ratio
if(pHN) sigma += pL*A*fHNXsc->HadronNucleonXscNS(theLambda,
theProton, pTkin);
if(pHN) tR *= std::sqrt( pG4Pow->Z23( pA - pL ) + cHN*pG4Pow->Z23( pL )
)/pG4Pow->Z13(pA);
```

G4ComponentGGHadronNucelusXsc class

- In the case a hyper-nucleus shoots hydrogen, we use inverse kinematics. So the hydrogen protons shoots the hyper nucleus.

```
void G4ComponentGGHadronNucleusXsc::ComputeCrossSections(
const G4ParticleDefinition* aParticle (=theProton),          G4double
kinEnergy, G4int Z, G4int A, G4int nL)
```

- The modifications are similar to G4ComponentGGNuclNuclXsc taking into account double inverse kinematics for lambda-nucleon cross-section:

```
if( nL > 0 ) {    G4double mp = theProton->GetPDGMass();    G4double ml =
theLambda->GetPDGMass();    G4double kinCof = ml/mp; // moving
hyperon - rest nucleon    G4double cHN(0.88);

sigma += nL*hnXsc->HadronNucleonXsc(theLambda, theProton,
kinEnergy*kinCof);

R *= std::sqrt( pG4Pow->Z23( A - nL ) + cHN*pG4Pow->Z23( nL ) )/pG4Pow-
>Z13(A);   }
```

Thanks

Discussion ?