Extension of Geant4 particles: hyper-nuclei / anti-nuclei

12th Geant4 Collaboration Workshop Hisaya Kurashige/Kobe Univ.

Current Status: geant4-09-00

Light Ion: individual classes derived from G4Ions

○ ²H : G4Deuteron ○ ³H : G4Triton

○ ³He : G4He3 ○ ⁴He : G4Alpha

lons : individual objects of G4lons

Specify

Z : Atomic Number = charge of nuclei in unit of +e

A: Atomic Mass = number of nucleons (#protons + #neutrons)

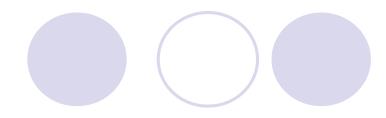
Q: Charge of Ion charge of nuclei in unit of +e

to Create ions

No limitation of combination of (Z, A) (as far as $Z \le A$)

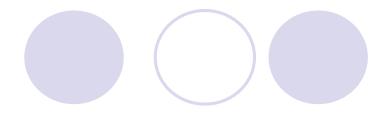
No Hyper-Nuclei nor Anti-Nuclei have been supported





- Light Nuclei
 - Given by experimental values according to PDG
- Other Nuclei (ground state)
 - Given by G4NucleiPropertiesTable for major nuclei according to experimental Data from G. Audi and A.H. Wapstra, Nucl. Physics, A595 vol 4 p 409-480, 25. Dec. 1995.
 - Given by G4NucleiPropertiesTheoreticalTable for other nuclei according to theoretical calculation from W.D. Myers, W.J. Swiatecki, P. Moller and J.R. Nix, 1. Jan. 1995.
 - Given by G4NucleiProperties for Nuclei not covered by above Tables according to Weitzsaecker's Mass formula

Nucleon Mass



- Mass for excited state
 - Excitation energy is given by hand

```
G4ParticleTable::FindIon(G4int atomicNumber,
G4int atomicMass,
G4double excitationEnergy);
```

 Some Excitation states are given by tables used for G4RadioActiveDecay

Nuclear Magnetic Moment (New from 08-03)

- Light Nuclei
 - Given by experimental values according to PDG
- Other Nuclei
 - Given by G4IsotopeMagneticMomentTable using G4IsotopeMagneticMoment.table
 - not applied in default
 - to activate code in PhysicsList

```
G4ParticleTable* particleTable= G4ParticleTable::GetParticleTable;
G4IonTable* ionTable= particleTable->GetIonTable;
ionTable->RegisterIsotopeTable( new G4IsotopeMagneticMomentTable());
```

define variable

```
export G4IONMAGNETICMOMENT
```

= \$(G4INSTALL)/source/particles/utils/
G4IsotopeMagneticMoment.table

PDG encoding for nuclei

- PDG codes for Nuclei : defined in PDG2006
 code = 10LZZZAAAI
- I = Isomer Number
 - I=0 for Ground State isotopes
 - I>0 for excitations
- ZZZ = charge (number of protons)
- A = total baryon number
 - = number of protons + number of neutrons + number of Λ s
- L = number of Λ^0 s (for hyper-nuclei)
 - No Σ hyper-nuclei is supported
 - 0 is reserved for high S nuclear states (strangelets) or for charmed nuclei (0=C)

PDG encoding for nuclei

- PDG codes for Nuclei was introduced into Geant4 from geant4-08-02 release
 - code = 10LZZZAAAI
- I = Isomer Number
 - I=0 for Ground State isotopes
 - *l*=1 for all excitations
- ZZZ = charge
- A = total baryon number
- No hyper-nuclei are supported
- No anti-nuclei are supported

Hyper-Nuclei in Geant4

- CHIPS model need to handle Hyper-Nuclei
- New Classes/Methods will be introduced for hyper-nuclei
 - Hyper-Nuclei is created by using

```
G4ParticleTable::FindIon(G4int atomicNumber,
G4int atomicMass,
G4int numbreOfLambda,
G4double excitationEnergy);
```

- PDG encoding for Hyper-Nuclei
- Name of Hyper-Nuclei
 - The letter 'L' is added at the head for each Lambda
 - For example
 - $\Lambda^{12}C \rightarrow LC12[0.0]$

Hyper-Nuclei in Geant4

 Mass for ground states are given by using mass formula based on CHIPS model (G4HyperNucleiProperties)

```
const G4double mL= lambda->GetPDGMass(); // mLambda
static const G4double b7=25.*MeV;
static const G4double b8=10.5; // Slope
static const G4double a2=0.13*MeV; // BindingEnergy for d+Lambda
static const G4double a3=2.2*MeV; // BindingEnergy for(t/He3)+Lambda
static const G4double eps =0.0001*MeV; // security value
G4double mass = G4NucleiProperties::GetNuclearMass(A-L, Z);
G4double bs=0.;
if (A-L ==2) bs=a2; // for nnL,npL,ppL
else if(A-L ==3) bs=a3; // for 3nL,2npL,n2pL,3pL
else if(A-L >3) bs=b7*exp(-b8/(A-L+1.));
mass += L*(mL-bs) + eps;
```

Anti-Nuclei in Geant4

- CHIPS model need to handle Anti-Nuclei also
- Only small modification is needed to define G4ParticleDefinition for Anti-Nuclei
- But Implementation of G4IonTable depends on is
 Which Energy-Loss process can be applied to Anti-Nuclei?
 - G4ionIonisation is used for ions in default now
 - Processes registered to G4Genericlon will be used for ions
 - If the current G4ionIonisation can not be applied Introduce New Process? Or extend G4ionIonisation
 I need answer from EM group