

I was doing some checks with low energy neutron elastic scattering (QGSP\_BIC\_HP physics list) when the angular distribution of scattered neutron is represented in CMS (i.e. frameFlag==2) and found a bug which results in momentum non conservation.

So, I have a primary neutron at PreStepPoint:

$$E_{kin} = 8.66851E - 08 \text{ MeV}$$

$$\vec{p} = (-0.00579051, \quad 0.0111418, \quad 0.00228551) \text{ MeV}/c$$

$$|\vec{p}| = 0.012763 \text{ MeV}/c$$

And the C-12 target which is sampled in G4ParticleHPElasticFS::ApplyYourself():

$$E_{kin} = 1.91023E - 08 \text{ MeV}$$

$$\vec{p} = (-0.0109931, \quad 0.00226974, \quad -0.0173519) \text{ MeV}/c$$

$$|\vec{p}| = 0.0206661 \text{ MeV}/c$$

The total before the interaction:

$$E_{kin} = 1.05787E - 07 \text{ MeV}$$

$$\vec{p} = (-0.0167836, \quad 0.01341154, \quad -0.0150664) \text{ MeV}/c$$

$$|\vec{p}| = 0.0262403 \text{ MeV}/c$$

After the interaction I have elastically scattered neutron at PostStepPoint:

$$E_{kin} = 9.06803E - 08 \text{ MeV}$$

$$\vec{p} = (-0.000673047, \quad 0.00658468, \quad 0.0112512) \text{ MeV}/c$$

$$|\vec{p}| = 0.0130538 \text{ MeV}/c$$

The recoiled C-12 nuclei in the vector of secondaries corresponding to the current step:

$$E_{kin} = 1.51122E - 08 \text{ MeV}$$

$$\vec{p} = (-0.00567584, \quad -0.0160092, \quad -0.00701742) \text{ MeV}/c$$

$$|\vec{p}| = 0.0183781 \text{ MeV}/c$$

The total after the interaction:

$$E_{kin} = 1.05793E - 07 \text{ MeV}$$

$$\vec{p} = (-0.0063489, \quad -0.00942452, \quad 0.0042338) \text{ MeV}/c$$

$$|\vec{p}| = 0.0121266 \text{ MeV}/c$$

The momentum is not conserved. This is caused by 2 things:

1. When G4HadProjectile is created its constructor applies rotation to Z axis for the original momentum of G4Track/G4DynamicParticle but the target nuclei which is sampled in G4ParticleHPElasticFS::ApplyYourself() is not rotated.
2. When G4HadronElasticProcess::PostStepDoit() returns the results it rotates the momentum of incident particle here –

<http://www-geant4.kek.jp/lxr/source/processes/hadronic/processes/src/G4HadronElasticProcess.cc#L246>

and the momentum of recoil here –

<http://www-geant4.kek.jp/lxr/source/processes/hadronic/processes/src/G4HadronElasticProcess.cc#L269>

These rotations are not inversed to those ones in G4HadProjectile. So they are incorrect (AT LEAST!) for particleHP model.

The fix:

1. In G4ParticleHPElasticFS::ApplyYourself()

Starting at this line

[http://www-geant4.kek.jp/lxr/source/processes/hadronic/models/particle\\_hp/src/G4ParticleHPElasticFS.cc#L202](http://www-geant4.kek.jp/lxr/source/processes/hadronic/models/particle_hp/src/G4ParticleHPElasticFS.cc#L202)

```
//theNeutron.SetMomentum( incidentParticle->Get4Momentum().vect() );
```

```

G4HadProjectile* theTrackNonConst = const_cast<G4HadProjectile*>(&theTrack);

G4LorentzRotation toLab = theTrackNonConst->GetTrafoToLab();

G4LorentzVector New4Momentum = incidentParticle->Get4Momentum();

New4Momentum.transform(toLab);

theNeutron.SetMomentum(New4Momentum.vect());

```

## 2. In G4HadronElasticProcess::PostStepDoit

On entering the function declare

```
G4bool isParticleHP = false;
```

Then change the try-block here

<http://www-geant4.kek.jp/lxr/source/processes/hadronic/processes/src/G4HadronElasticProcess.cc#L161>

```
try
```

```
{
```

```
hadi = ChooseHadronicInteraction( theProj, *targNucleus, material, elm );
```

```
G4ParticleHPElastic* itsHP = dynamic_cast<G4ParticleHPElastic*>(hadi);
```

```
if (itsHP) isParticleHP = true;
```

```
}
```

Then change this if-block

<http://www-geant4.kek.jp/lxr/source/processes/hadronic/processes/src/G4HadronElasticProcess.cc#L245>

```
if(efinal > 0.0) {
```

```
    (!isParticleHP) outdir.rotate(phi, it);
```

```
    (!isParticleHP) outdir.rotateUz(indir);
```

```

theTotalResult->ProposeMomentumDirection(outdir);

}

```

And do the same in this if-block:

<http://www-geant4.kek.jp/lxr/source/processes/hadronic/processes/src/G4HadronElasticProcess.cc#L264>

```

(!isParticleHP) outdir.rotate(phi, it);

(!isParticleHP) outdir.rotateUz(indir);

```

After the fix I have a primary neutron at PreStepPoint:

$$E_{kin} = 8.66851E - 08 \text{ MeV}$$

$$\vec{p} = (-0.00579051, \quad 0.0111418, \quad 0.00228551) \text{ MeV}/c$$

$$|\vec{p}| = 0.012763 \text{ MeV}/c$$

And the C-12 target which is sampled in G4ParticleHPElasticFS::ApplyYourself():

$$E_{kin} = 1.91023E - 08 \text{ MeV}$$

$$\vec{p} = (-0.0109931, \quad 0.00226974, \quad -0.0173519) \text{ MeV}/c$$

$$|\vec{p}| = 0.0206661 \text{ MeV}/c$$

The total before the interaction:

$$E_{kin} = 1.05787E - 07 \text{ MeV}$$

$$\vec{p} = (-0.0167836, \quad 0.01341154, \quad -0.0150664) \text{ MeV}/c$$

$$|\vec{p}| = 0.0262403 \text{ MeV}/c$$

After the interaction I have elastically scattered neutron at PostStepPoint:

$$E_{kin} = 7.16531E - 08 \text{ MeV}$$

$$\vec{p} = (0.0041171, \quad 0.0104916, \quad 0.00276069) \text{ MeV}/c$$

$$|\vec{p}| = 0.0116037 \text{ MeV}/c$$

The recoiled C-12 nuclei in the vector of secondaries corresponding to the current step:

$$E_{kin} = 3.41461E - 08 \text{ MeV}$$

$$\vec{p} = (-0.0209005, \quad 0.00291988, \quad -0.0178269) \text{ MeV}/c$$

$$|\vec{p}| = 0.0276253 \text{ MeV}/c$$

The total after the interaction:

$$E_{kin} = 1.05799E - 07 \text{ MeV}$$

$$\vec{p} = (-0.0167834, \quad 0.01341148, \quad -0.01506621) \text{ MeV}/c$$

$$|\vec{p}| = 0.0262401 \text{ MeV}/c$$

Now momentum is conserved.

P.S.

Also this whole thing

[http://www-geant4.kek.jp/lxr/source/processes/hadronic/models/particle\\_hp/src/G4ParticleHPElasticFS.cc#L313](http://www-geant4.kek.jp/lxr/source/processes/hadronic/models/particle_hp/src/G4ParticleHPElasticFS.cc#L313)

```
G4ThreeVector tempVector;
```

```
tempVector.setX(std::cos(theta)*std::sin(cms_theta)*std::cos(cms_phi)
                +std::sin(theta)*std::cos(phi)*std::cos(cms_theta)*std::cos(cms_phi)
                -std::sin(theta)*std::sin(phi)*std::sin(cms_phi) );
```

```
tempVector.setY(std::cos(theta)*std::sin(cms_theta)*std::sin(cms_phi)
                +std::sin(theta)*std::cos(phi)*std::cos(cms_theta)*std::sin(cms_phi)
                +std::sin(theta)*std::sin(phi)*std::cos(cms_phi) );
```

```
tempVector.setZ(std::cos(theta)*std::cos(cms_theta)
```

```
-std::sin(theta)*std::cos(phi)*std::sin(cms_theta) );
```

could be more naturally presented as

```
G4ThreeVector tempVector =
```

```
G4ThreeVector(std::sin(theta)*std::cos(phi), std::sin(theta)*std::sin(phi), std::cos(theta));
```

```
tempVector.rotateUz(cmsMom_tmp.unit());
```

P.P.S.

Is there any references regarding G4Nucleus::GetBiasedThermalNucleus() – method?

I found only one very old Geant4 presentation (2001-2002) where it is described as the Doppler shift which increases momentum of the thermal nuclei from the POV of the projectile. This raises some questions since the cross section is calculated in target rest frame and all the changes are applied in LAB or CMS. Could it be replaced with G4Nucleus::GetThermalNucleus() – method ?