
Subject: Inconsistency geant3/gcons/gpart.F and TGean3.cxx

Posted by [asanchez](#) on Thu, 14 May 2009 14:50:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

Dear all, i have noticed there is some inconsistency between the defined particles by TGean3.cxx or TGean4.cxx and geant3/gcons/gpart.F

I was using the geanetrackrep for tracking, and i saw that the tracking macro crashes when a deuteron is to be tracked by geane.

here can you see the message,

Quote:11 hits in track 5

Position : (-0.40, 0.76, -76.)

Slopes : dx/dz = -0.84, dy/dz = 1.8

q/p = 1.6

Propagate in flight direction

Error in ERTRAK : particle type 0 unknown in GEANT

*** Break *** floating point exception

Using host libthread_db library "/lib/libthread_db.so.1".

Attaching to program: /proc/10412/exe, process 10412

Failed to read a valid object file image from memory.

done.

done.

[Thread debugging using libthread_db enabled]

[New Thread 16384 (LWP 10412)]

done.

done.

done.

done.

done.

done.

done.

done.

done.

0x4117f788 in waitpid () from /lib/libc.so.6

#1 0x412078c0 in __DTOR_END__ () from /lib/libc.so.6

#2 0x41118442 in do_system () from /lib/libc.so.6

#3 0x41093c5f in system () from /lib/libpthread.so.0

#4 0x40237363 in TUnixSystem::Exec (this=0x80e2a78,

#5 0x40237836 in TUnixSystem::StackTrace (this=0x80e2a78) at core/unix/src/TUnixSystem.cxx:2121

#6 0x402356f5 in TUnixSystem::DispatchSignals (this=0x80e2a78, sig=kSigFloatingException)

at core/unix/src/TUnixSystem.cxx:1089

#7 0x402336b4 in SigHandler (sig=kSigFloatingException) at core/unix/src/TUnixSystem.cxx:351

#8 0x4023a6d3 in sighandler (sig=8) at core/unix/src/TUnixSystem.cxx:3344

#9 0x41092825 in __pthread_sighandler () from /lib/libpthread.so.0

```

#10 <signal handler called>
#11 0x44e17e2e in CbmGeanePro::Track2ToPoint (this=0xa2008f0, x1=
    {<TObject> = {_vptr.TObject = 0x43249e08, fUniqueID = 0, fBits = 33554432, static
fgDtorOnly = 0, static fgObjectStat = false, static fglsA = 0x82b48c0}, fX = 0, fY = 0, fZ = 0,
static fglsA = 0x9e4e3c8}, x2=
    {<TObject> = {_vptr.TObject = 0x43249e08, fUniqueID = 0, fBits = 33554432, static
fgDtorOnly = 0, static fgObjectStat = false, static fglsA = 0x82b48c0}, fX = 0, fY = 0, fZ = 0,
static fglsA = 0x9e4e3c8},
    w1=
    {<TObject> = {_vptr.TObject = 0x43249e08, fUniqueID = 0, fBits = 33554432, static
fgDtorOnly = 0, static fgObjectStat = false, static fglsA = 0x82b48c0}, fX = 0, fY = 0, fZ = 0,
static fglsA = 0x9e4e3c8}, Pfinal=@0xa200af0, Dist=@0xa200ae8, Length=@0xbf803f38)
    at /u/asanchez/FairJan09/trunk/geane/CbmGeanePro.cxx:762
#12 0x44e14f8b in CbmGeanePro::FindPCA (this=0xa2008f0, pca=1, PDGCode=1000010020,
point=
    {<TObject> = {_vptr.TObject = 0x43249e08, fUniqueID = 0, fBits = 33554432, static
fgDtorOnly = 0, static fgObjectStat = false, static fglsA = 0x82b48c0}, fX = 0, fY = 0, fZ = 0,
static fglsA = 0x9e4e3c8}, wire1=
    {<TObject> = {_vptr.TObject = 0x44e2b1e8, fUniqueID = 0, fBits = 50331648, static
fgDtorOnly = 0, static fgObjectStat = false, static fglsA = 0x82b48c0}, fX =
1.563365687775424e-260, fY = 1.5633904981780072e-260, fZ = 149.63770000687941, static
fglsA = 0x9e4e3c8}, wire2=
    {<TObject> = {_vptr.TObject = 0x43249e08, fUniqueID = 0, fBits = 33554432, static
fgDtorOnly = 0, static fgObjectStat = false, static fglsA = 0x82b48c0}, fX = 0, fY = 0, fZ = 0,
static fglsA = 0x9e4e3c8}, maxdistance=151.28160462801051, Rad=@0xa200ae0,
vpf=@0xa200af0, vwi=@0xa200b14, Di=@0xa200ae8,
    trklength=@0xa200b38) at /u/asanchez/FairJan09/trunk/geane/CbmGeanePro.cxx:554
#13 0x44e13158 in CbmGeanePro::Propagate (this=0xa2008f0, TStart=0xbf804980,
TEnd=0xbf804460,
    PDG=1000010020) at /u/asanchez/FairJan09/trunk/geane/CbmGeanePro.cxx:180
#14 0x441dd9f8 in GeaneTrackRep::extrapolateToPoca (this=0xa432300, pos=@0xa25e874,
    statePred=@0xbf8050f0, covPred=@0xbf804ff0, pl=@0xbf805220)
    at /u/asanchez/FairJan09/trunk/trackrep/GeaneTrackRep.cxx:297
#15 0x441dbb06 in GeaneTrackRep::getVirtualDetPlane (this=0xa432300, hit=@0xa25e874)
    at /u/asanchez/FairJan09/trunk/trackrep/GeaneTrackRep.cxx:97
#16 0x4418af2f in SpacepointHitPolicy::detPlane (this=0xa25e870, hit=0xa25e570,
rep=0xa432300)
    at /u/asanchez/FairJan09/trunk/genfit/SpacepointHitPolicy.cxx:94
#17 0x44174696 in RecoHitLfc<SpacepointHitPolicy>::getDetPlane (this=0xa25e570,
rep=0xa432300)
    at RecoHitLfc.h:39
#18 0x44176099 in Kalman::processHit (this=0xbf806560, hit=0xa25e570, rep=0xa432300,
hitIndex=0)
    at /u/asanchez/FairJan09/trunk/genfit/Kalman.cxx:167
#19 0x441758cb in Kalman::continueTrack (this=0xbf806560, trk=0xa9e9268, direction=1)
    at /u/asanchez/FairJan09/trunk/genfit/Kalman.cxx:92
#20 0x441756ad in Kalman::processTrack (this=0xbf806560, trk=0xa9e9268)
    at /u/asanchez/FairJan09/trunk/genfit/Kalman.cxx:48
#21 0x44f8adfd in PndHypDKalmanTask::Exec (this=0x9e3b4c0, opt=0x43d55440 "")
    at /u/asanchez/FairJan09/trunk/hyp/hypTracking/PndHypDKalmanTask.cxx:165
#22 0x401b16af in TTask::ExecuteTasks (this=0x86a0390, option=0x43d55440 "")
    at core/base/src/TTask.cxx:298

```

```

#23 0x401b14b1 in TTask::ExecuteTask (this=0x86a0390, option=0x43d55440 "")
    at core/base/src/TTask.cxx:261
#24 0x43cd89a0 in CbmRunAna::Run (this=0x86a0308, Ev_start=0, Ev_end=4917)
    at /u/asanchez/FairJan09/trunk/base/CbmRunAna.cxx:195
#25 0x43d192de in G__CbmDict_531_0_5 (result7=0xbf80da70, funcname=0x869e508
"\001", libp=0xbf807bf0,
    hash=0) at /u/asanchez/FairJan09/build/base/CbmDict.cxx:9268
#26 0x407b6126 in Cint::G__ExceptionWrapper (funcp=0x43d191da
<G__CbmDict_531_0_5>, result7=0xbf80da70,
    funcname=0x869e508 "\001", libp=0xbf807bf0, hash=0) at cint/cint/src/Api.cxx:364
#27 0x408757f5 in G__execute_call (result7=0xbf80da70, libp=0xbf807bf0, ifunc=0x869e508,
ifn=0)
    at cint/cint/src/newlink.cxx:2305
#28 0x40875ed8 in G__call_cppfunc (result7=0xbf80da70, libp=0xbf807bf0, ifunc=0x869e508,
ifn=0)
    at cint/cint/src/newlink.cxx:2471
#29 0x40855818 in G__interpret_func (result7=0xbf80da70, funcname=0xbf80d670 "Run",
libp=0xbf807bf0,
    hash=309, p_ifunc=0x869e508, funcmatch=1, memfunc_flag=1) at
cint/cint/src/ifunc.cxx:5245
#30 0x40834ca1 in G__getfunction (item=0xbf8104c6 "Run(0,nEvents)", known3=0xbf80fd0c,
memfunc_flag=1)
    at cint/cint/src/func.cxx:2534
#31 0x40940b6b in G__getstructmem (store_var_type=112, varname=0xbf80dd00 "",
    membername=0xbf8104c6 "Run(0,nEvents)", tagname=0xbf80df10 "fRun",
known2=0xbf80fd0c,
    varglobal=0x409ebec0, objptr=2) at cint/cint/src/var.cxx:6623
#32 0x409329d7 in G__getvariable (item=0xbf8104c0 "fRun->Run(0,nEvents)",
known=0xbf80fd0c,
    varglobal=0x409ebec0, varlocal=0xbf812a10) at cint/cint/src/var.cxx:5252
#33 0x40825f01 in G__getitem (item=0xbf8104c0 "fRun->Run(0,nEvents)") at
cint/cint/src/expr.cxx:1884
#34 0x4082387b in G__getexpr (expression=0xbf811d90 "fRun->Run(0,nEvents)")
    at cint/cint/src/expr.cxx:1470
#35 0x4089fde5 in G__exec_function (statement=0xbf811d90 "fRun->Run(0,nEvents)",
pc=0xbf8121bc,
    piout=0xbf8121b4, plargestep=0xbf8121a4, presult=0xbf811d60) at
cint/cint/src/parse.cxx:601
#36 0x408af2a6 in G__exec_statement (mparen=0xbf8126dc) at cint/cint/src/parse.cxx:6972
#37 0x408587a3 in G__interpret_func (result7=0xbf818eb0, funcname=0xbf818ab0 "runReco",
    libp=0xbf813030, hash=734, p_ifunc=0x82b5048, funcmatch=1, memfunc_flag=0)
    at cint/cint/src/ifunc.cxx:6080
#38 0x4083584a in G__getfunction (item=0xbf8197b0 "runReco()", known3=0xbf818ffc,
memfunc_flag=0)
    at cint/cint/src/func.cxx:2745
#39 0x40826044 in G__getitem (item=0xbf8197b0 "runReco()") at cint/cint/src/expr.cxx:1896
#40 0x4082387b in G__getexpr (expression=0x810ba88 "runReco()") at
cint/cint/src/expr.cxx:1470
#41 0x408114ff in G__calc_internal (exprwithspace=0xbf81cf00 "runReco()") at
cint/cint/src/expr.cxx:1061
#42 0x408b7d27 in G__process_cmd (line=0x406e9b0e "elize", prompt=0x80e6264 "",

```

```
more=0x80e625c,  
    err=0xbf81d77c, rslt=0xbf81d780) at cint/cint/src/pause.cxx:2234  
#43 0x4021e958 in TCint::ProcessLine (this=0x80e6240, line=0x406e9b0e "elize",  
error=0xbf81ff84)  
    at core/meta/src/TCint.cxx:339  
#44 0x4021ed70 in TCint::ProcessLineSynch (this=0x80e6240, line=0x406e9b0e "elize",  
error=0xbf81ff84)  
    at core/meta/src/TCint.cxx:406  
#45 0x4013581a in TApplication::ExecuteFile (file=0xbf81df23 "runReco.C", error=0xbf81ff84)  
    at core/base/src/TApplication.cxx:938  
#46 0x40135094 in TApplication::ProcessFile (this=0x81078b8, file=0xbf81df23 "runReco.C",  
error=0xbf81ff84) at core/base/src/TApplication.cxx:825  
#47 0x40134fd7 in TApplication::ProcessLine (this=0x81078b8, line=0xbf81df20 ".x  
runReco.C",  
    sync=false, err=0xbf81ff84) at core/base/src/TApplication.cxx:798  
#48 0x4101d78c in TRint::Run (this=0x81078b8, retrn=false) at core/rint/src/TRint.cxx:355  
#49 0x08048ece in main (argc=1, argv=0xbf820044) at main/src/rmain.cxx:29  
Root > Function runReco() busy flag cleared
```

It seems that geane doesn't recognise the particule deuteron even if it already defined.

By looking in geant3/erdecks/ertrack.F and in geant3/gcons/gpart.F i realized that indeed the array NPART which contains the id of the defined particles, doesn't contains the id for the deuteron, alpha, helium, ..etc.

But if you look into the TGeant3.cxx and TGeant4.cxx these ions are already defined.

So my question is,

Is geane taking as defined particles only what is provided by gpart.F?

I thought, that the particles already defined for the montecarlo simulation should be also recognised by geane.

thanks a lot

ALicia

Subject: Re: Inconsistency geant3/gcons/gpart.F and TGeant3.cxx

Posted by [Lia Lavezzi](#) on Mon, 18 May 2009 10:07:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Alicia,

I had a look to the propagation of a deuteron in GEANE and actually I see that GEANE can propagate it: the problem is in the conversion from the PDG code to the Geant code of the particle, which is performed before calling ertrak, precisely in the line (in FairGeanePro.cxx):
GeantCode=fdbPDG->ConvertPdgToGeant3(PDG);

Here TDatabasePDG is called to make the conversion, but when the PDG of the deuteron is put as input it returns as GeantCode the default value 0 instead of the right one (45).

In TDatabasePDG::ConvertPdgToGeant3 no line for the deuteron is present; if TDatabasePDG is set to handle correctly the deuteron, I think GEANE will have no more problem in tracking it.

As a test, I "bypassed" the TDatabasePDG adding by hand a line just after
GeantCode=fdbPDG->ConvertPdgToGeant3(PDG);
to set the correct code:
if(PDGCode == 1000010020) GeantCode=45;
and I see GEANE does not crash anymore.

Ciao,
Lia.

Subject: GenaeTrackRep DetPlane
Posted by [asanchez](#) on Mon, 25 May 2009 09:34:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Lia and Susanna,
I have one more question related to how one should define the
DefPlane in order to use geanetrakrep.

So far i know (please correct me if i'm wrong)
In Geane coordinates system, the direction of the
propagation is the X-axis, right??.

I have seen that in (stt case) PndSttPatternRecoTask2.cxx
class, you define the Detector plane in the Y--Z projection
so as to have the beam in the x--direction.

```
176      TVector3 u(0.,1.,0.);
177      TVector3 v(0.,0.,1.);
178      DetPlane pl(StartPos,u,v);
179
180      AbsTrackRep* rep = 0;
181      GeaneTrackRep *grep = new
GeaneTrackRep(fPro,pl,StartMom,StartPosErr,StartMomErr,fCharge,pdg)
;
```

Is that general for all detectors, i mean, if one wants to use
GeaneTrackRep representation, should the detplane be defined
also in the Y--Z projection?

I say that, because for example, in the LSLTrackRep, the detplane
is defined as beeing in the X--Y projection.

```
TVector3 u(1.,0.,0.);
TVector3 v(0.,1.,0.);
```

Thank you in advance,
ALicia Sanchez.

Subject: Re: GenaeTrackRep DetPlane
Posted by [Anonymous Poster](#) on Mon, 25 May 2009 09:51:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Alicia,

could you please explain shortly what the geometry of your detector/hits is? Is it measuring in some plane, is it some wire, ...? Sorry for not knowing this. If you give me a little info, I am sure I can also help.

Cheers, Christian

Subject: Re: GenaeTrackRep DetPlane
Posted by [Lia Lavezzi](#) on Mon, 25 May 2009 10:15:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Alicia,
in GEANE when you define the SD (detector system) frame, which is the one we use with detector planes, you have a u, v, w system of axis and the GEANE plane (let's call it this way) is the vw (so, yz plane of the frame), with the u axis orthogonal to it (I guess this is what you mean, with

Quote: So far I know (please correct me if I'm wrong)

In Geane coordinates system, the direction of the propagation is the X-axis, right??.

is this correct?), but I think that with genfit you don't have to care about this, this is handled directly by GeaneTrackRep (when FairTrackParP is created), you just have to define as DetPlane your detector plane (with any orientation in space).

The only thing you have to pay attention for is that your track momentum must not lie on the plane, but it must have some angle with respect to it, otherwise you have some kind of divergence...

In the Stt case, the yz plane you mention is just a starting plane, in this case it is set in the origin, where no real plane is present, so in principle we could have chosen xy plane as well. If you want to choose the xy plane as starting plane you can do it! ... if you have problems originating from the det plane choice, please let me know

Ciao,
Lia.

Subject: Re: GenaeTrackRep DetPlane

Posted by [asanchez](#) on Mon, 25 May 2009 10:23:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Christian,
In principle my detector
is similar to mvd, in the sense, that
is also consisting out of silicon layers(strips)
, so that means, planar geometry.

For the track representation i ('m still)was using SpacePointHitPolicy and
Planer?HitPolicy.(so DetPlane(point,U(1,0,0),V(0,1,0))

My question came due to the fact, i have observed that the stt was using a different projection
to define its DetPlane.

And i was wondering wether the plane should be initialated
in another way for geane(i mean taking .U and V at the Y--Z
projection.

Sorry maybe it is a stupid question.

cheers

Alicia

Subject: Re: GenaeTrackRep DetPlane

Posted by [asanchez](#) on Mon, 25 May 2009 10:27:13 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Lia,
Thanks a lot for the explanation

ALicia.

Subject: Re: GenaeTrackRep DetPlane

Posted by [Anonymous Poster](#) on Wed, 27 May 2009 08:32:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

sorry for not replying for such a long time: Are your planes of silicon always perpendicular to
the z-axis?

Cheers, Christian

Subject: Re: GenaeTrackRep DetPlane

Posted by [asanchez](#) on Wed, 27 May 2009 09:16:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Christian,
the planes(layers) are placed around the Z-axis,

like the mvd case. So yes let say they are parallel to the Z-axis.

Maybe we can discussed more in detaile during the panda meeting in turin. Of course if you have time .

Cheers
Alicia.

Subject: Re: GenaeTrackRep DetPlane
Posted by [Anonymous Poster](#) on Wed, 27 May 2009 10:12:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

hmm us bad kids writing in the forum during the PandaROOT meeting

Yes, I will have time in Torino. Maybe I can give you a little help even now. You can take a look to a class called StripHit.h,cxx} in genfit/benchmark.

I define a strip detector hit here, which is perpendicular to the z-axis. But changing this should be easy for you.

What you need to do:

- If it is 1D, you can leave the HMatrix like it is, if it is 2D, add another 1 at `_HMatrix[0][4] = 0`. Also you need to set the NParHitRep to 1 or 2 accordingly.

- make a contructor, that accepts an object of the class you use for representing the hit (a digi or so)

- in this constructor construct the DetPlane (which you then set with `setDetPlane()` after it is complete) so that the span vector U points along the coordinate which your detector measures (and V of course perpendicular to it, and it measures the second coordinate in case you have one) Of course also you have to define the origin of the plane.

- then you need to fill `_hitCoord[i][j]` and `_hitCov[i][j]`. If it is 1D, then hitCov is just σ^2 , if it is 2D, hitCoord has two entries and hitCov is a 2x2 matrix. Most likely you just set the diagonal to the σ^2 of the projections. If there are correlations you can put them, and they will be used correctly.

If you have more questions, please ask.

Cheers, Christian

Subject: Re: GenaeTrackRep DetPlane
Posted by [asanchez](#) on Wed, 27 May 2009 10:39:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Christian,

thank you very much for your advice.
Actually i have taken your DemoSPRecoHit. class
as example. And if i use LSLtrackrep it is working well.

If instead of SpatialHitPolicy i use the PlaneHitPolicy
then i'm getting huge values for the chi square.
But that is something what can still be investigated.
Nevertheless i appreciate very much your advices, .

I will take a look into the StripHit.h,cxx, and i will let you know.

by best regards
ALicia.

Subject: Re: GenaeTrackRep DetPlane
Posted by [Anonymous Poster](#) on Wed, 27 May 2009 21:55:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

if you like I can take a look at the code and check if it looks fine...

Cheers, Christian

Subject: Re: GenaeTrackRep DetPlane
Posted by [asanchez](#) on Thu, 28 May 2009 16:23:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Christian,
sorry i was today busy with
some seminars.

Of course, i will be glad if you can have a look
into my code.

My recohits(Plane(MVD--like) and spacePoint) are at hyp/hypData.

i will wait for your remarks

best regards
Alicia.
