Subject: charged geantino
Posted by donghee on Wed, 13 Oct 2010 08:07:58 GMT
View Forum Message <> Reply to Message

Dear collegues,

Does anyone know how charged geantino with positive and negative charge can be introduced serarately in the simulation?

The CbmParticle doesn't exist and is replaced with FairParticle.
How can I put this FairParticle into Boxgenerator?

Someone suggests that I can use TParticlePDG via...
Quote:
Int_t pdg;
TDatabasePDG *db= TDatabasePDG::Instance();
TParticlePDG *p=0;
p=db->GetParticle("chargedGeant");
if(p) pdg = p->PdgCode();

But pdg is always equal to zero 0, and just points to rootino.
and doesn't contain any charge info with this method.

Do you have some experience for usage of charged/neutral geantino in pandaroot and can surely distinguish from rootino with certain method?

Have a nice day!

Best regards,
Donghee

Subject: Re: charged geantino
Posted by Prometeusz Jasinski on Wed, 01 Feb 2012 15:21:56 GMT
View Forum Message <> Reply to Message

Hi,

still no solution for this after so many years?!

Subject: Re: charged geantino
Posted by StefanoSpataro on Thu, 02 Feb 2012 08:23:48 GMT
View Forum Message <> Reply to Message

Dear Promme,
The problem is not solved. If you are interested to solve it you could check in root and in TParticlePDG how it is defined, if it is defined. Or you could take a charged particle and disable all the physics lists, probably this should also do the job

---

Subject: Re: charged geantino
Posted by Prometeusz Jasinski on Thu, 02 Feb 2012 08:34:16 GMT
View Forum Message <> Reply to Message

Unfortunately I do not have the overview yet as root, pandaroot, Geant3 and Geant4 do have their own lists. That means that, as far as I can see it, creating a charged rootino is not mapped correctly to a Geantino in Geant4. Too many dependencies to figure it out on a short way. Let's see if I can find the time to figure it out by my self.

Cheers

Subject: Re: charged geantino
Posted by StefanoSpataro on Thu, 02 Feb 2012 10:00:16 GMT
View Forum Message <> Reply to Message

This is the answer from Ivana about:

Quote:Hi Stefano,

As Root has only Rootino and no ChargedRootino, both geantino and chargedgeantino are mapped to Rootino.
And Rootino should be mapped to geantino, but I have realized now that it is mapped by mistake to chargedgeantino.
So you can just define kRootino in your application and it will be interpreted as chargedgeantino.

But as this interpretation is not correct, after I will fix it in geant4_vmc, there will be no other way how to define it than to modify the mapping in the geant4_vmc code.
I can ask Root team if we can add ChargedRootino in Root PDG table and then map both particles properly.

Cheers,

Ivana

On 02/02/2012 10:11 AM, Stefano Spataro wrote:
> Dear Ivana,
> I have a short question. Is the charged geantino currently mapped into virtual mc? I mean, if I need to shoot some charged geantino, which PDG code I should use? I have tried to search for this info but I was not able to find it, or maybe I should just take some charged particle and switch off all the interactions?
>
> Many thanks in advance
>

> Ste

Hope this helps.

Subject: Re: charged geantino
Posted by asanchez on Thu, 02 Feb 2012 10:45:07 GMT
View Forum Message <> Reply to Message

Dear Stephano and Promme, actually one can use
directly the class PndParticle and define the particle
with DefineParticle constructor as you like.

best regards
ALicia.

Subject: Re: charged geantino
Posted by Prometeusz Jasinski on Thu, 02 Feb 2012 10:52:16 GMT
View Forum Message <> Reply to Message

you mean as discussed in this thread but was not working there. I should give it a try.

http://forum.gsi.de/index.php?t=msg&th=1722&rid=1777&S=bf478cf8209908f4f20869f569eb94ff#msg_5844

oh, nope, there you have proposed something else. Sorry I was wrong.

Subject: Re: charged geantino
Posted by asanchez on Thu, 02 Feb 2012 11:12:33 GMT
View Forum Message <> Reply to Message

Hi again, Ok now it is called FairParticle at the fairbase/base directory,
the general constructor is

```
FairParticle( Int_t pdg , const TString name, TMCParticleType mcType, Double_t mass,
Double_t charge,
        Double_t lifetime, const TString pType="Ion",  Double_t width=0, Int_t iSpin=0, Int_t
iParity=0,
    Int_t iConjugation=0, Int_t iIsospin=0, Int_t iIsospinZ=0, Int_t gParity=0, Int_t lepton=0,
    Int_t baryon=0,Bool_t stable=kFALSE);
```

In your simulation macro (first approach),

you should define your particle (for example in my case it was a double hypernucleus)

```
FairParticle  *B13LL = new FairParticle("B13LL", 5,13, 2,12.45659,5,kFALSE,0.200e-9);
   fRun->AddNewParticle(B13LL);
```

you have to know what is the pdgcode assigned to your particle,
(so that you can generate it afterwards) by doing

```
 Int_t B13LLPDG;
      p= db->GetParticle("B13LL");
    if(p) B13LLPDG=p->PdgCode();

     cout<<B13LLPDG<<endl;
```

for the physical processes , const TString pType="Ion", you should see that you get something similar to the rootino/geantino family.
instead of "ion".

I hope it works

cheers
Alicia.
