Subject: GLAD positioning
Posted by lzanetti on Tue, 13 Jun 2017 15:12:20 GMT
View Forum Message <> Reply to Message

Hi there,

Apparently, the initial rotation in "$R3BROOT_HOME/macros/r3b/geo/create_glad_geo.C", defined at lines 79 to 86, has either no effect or is discarded... or something.
This is creating, in my simulations, a GLAD magnet which is centered in the correct (?) place but not rotated properly.
I tried to change the angles around (at first, just to see what happens), to no effect.

I'm already on the latest version of "dev", commit:
8f11ca90ed374612082cf5358672e411e5dd0175.

Attached, there's a screenshot from eventDisplay.C, illustrating the problem. That's a 750MeV proton, generated at the target's position (0,0,0) and propagating. The red box is there simply to highlight the origin, it has no particular meaning.

So far, the solution is to simply avoid GLAD's bulk in the simulation and just use its field, but eventually we'll need to know the effect the magnet's having. How can we solve this?

Thanks for your time.
Cheers,
L.

File Attachments
1) Screenshot from 2017-06-13 17:00:13.png, downloaded 275 times

Subject: Re: GLAD positioning
Posted by Dmytro Kresan on Wed, 14 Jun 2017 07:46:13 GMT
View Forum Message <> Reply to Message

Hi Lorenzo,

We have introduced a new feature: possibility to move and rotate the detectors directly in the macro. The values specified in the call to standard constructor overwrite the values stored in the geometry file. That is why you do not see the effect. An exception is done for GLAD, where the position and angle (taken from the actual placement in the cave) are hard-coded in R3BGladMagnet.cxx in the standard constructor, and they are not supposed to be changed from experiment to experiment.

In order to debug your problem, please post here:
1. Which geometry version are you using?
2. How does the call to R3BGladMagnet(...) look like in your case (in the macro)?
3. What are the values of position and angle in the R3BGladMagnet(...) implementation in your case?

Best regards,

Dima

Subject: Re: GLAD positioning
Posted by lzanetti on Wed, 14 Jun 2017 15:32:13 GMT
View Forum Message <> Reply to Message

Hi Dima, thank you for your reply.

Dmytro Kresan wrote on Wed, 14 June 2017 09:46
We have introduced a new feature: possibility to move and rotate the detectors directly in the macro. The values specified in the call to standard constructor overwrite the values stored in the geometry file.
That's nice --and useful to me, since that will be needed soon around here, I think. I'll look at the code.
As a side question: where I can find this kind of information? (except bothering you here...)

Dmytro Kresan wrote on Wed, 14 June 2017 09:46
That is why you do not see the effect.

Does that mean that the detector position has to be always passed to the constructors, now?

Dmytro Kresan wrote on Wed, 14 June 2017 09:46
An exception is done for GLAD, where the position and angle (taken from the actual placement in the cave) are hard-coded in R3BGladMagnet.cxx in the standard constructor, and they are not supposed to be changed from experiment to experiment.

Ok. So GLAD cannot be moved --and I don't want to: I tried because it looked like it was in the wrong position, or more specifically not rotated by its 14 degrees.

Dmytro Kresan wrote on Wed, 14 June 2017 09:46
1. Which geometry version are you using?
2. How does the call to R3BGladMagnet(...) look like in your case (in the macro)?
3. What are the values of position and angle in the R3BGladMagnet(...) implementation in your case?

The one included in the commit I posted above. More specifically: the new commits don't seem to update the geometry binaries (GLAD's was untouched since last December), so it's the last version of the geometry macro and the geometry file has been regenerated with that.It's just constructed by name. The executable I use calls the constructors just by name, as r3ball.C used to do --I see this is no longer the case and other arguments are passed to the GLAD constructor.
This is how my code calls the constructor:

R3BModule* mag = new R3BGladMagnet("GladMagnet");
There are none I'm passing at runtime. All there is is what's in create_glad_geo.C and is used

for the geometry binary. If you're asking what's in the source file for GLAD's constructor, it's exactly the one in the commit I posted before...

~~~

EDIT:
Found it: fGladAngle is not initialized in the init list of the "just by name" constructor --which has a default argument as the second constant string (ll 32-34):

```
R3BGladMagnet::R3BGladMagnet(const char* name, const char* Title)
    : R3BModule(name, Title)
{
```

This can be changed to:

```
R3BGladMagnet::R3BGladMagnet(const char* name, const char* Title)
    : R3BModule(name, Title),
      fGladAngle( 14. )
{
```

To solve my problem (I think, didn't actually try yet). And to preserve compatibility with old code.

I'm not supposed to be working right now (holidays...) so I won't touch it, if this one liner is included in a commit on "dev" before Monday.

Subject: Re: GLAD positioning
Posted by Dmytro Kresan on Thu, 22 Jun 2017 05:26:56 GMT
View Forum Message <> Reply to Message

Quote:As a side question: where I can find this kind of information? (except bothering you here...)

This information was sent to the mailing list of the Simulation and Analysis Working Group. If you want to be added - send a request to Hector Alvarez Pol (group leader).

If you do not pass position and angle arguments to the constructor - the values from the geometry file are taken (exception for glad - the values are hard-coded in the call to ctor of the base class).

I recommend usage of "glad_v17_flange.geo.root" geometry file - the closest to real geometry we have in the cave.

Here is how the constructor is called in run_sim.C (new simulation macro, was also announced through the mailing list):

```
new R3BGladMagnet("GladMagnet", "glad_v17_flange.geo.root", "R3BGlad Magnet")
```
~~~

and here is the part of its implementation with the default position and angle parameters:

```
R3BGladMagnet::R3BGladMagnet(const char* name, TString geoFile, const char* Title)
    : R3BModule(name, Title, geoFile, -42., 0., 308.8)
    , fGladAngle(14.)
{
}
```

And all this works properly in the central dev branch.

Best regards,
Dima

Subject: Re: GLAD positioning
Posted by lzanetti on Thu, 22 Jun 2017 11:02:38 GMT
View Forum Message <> Reply to Message

Hi Dima,

Dmytro Kresan wrote on Thu, 22 June 2017 07:26
If you do not pass position and angle arguments to the constructor - the values from the geometry file are taken (exception for glad - the values are hard-coded in the call to ctor of the base class).

Not in the case of GLAD, if the "by name" constructor is used. Because GLAD's rotation isn't read from the geometry, but from the variable fGladAngle, which is not initialized in with that particular constructor, as I mentioned above. BTW, I tried the solution I was proposing: it works.

Specifically, GLAD's rotation is created like this:

```
TGeoRotation* rot_glob = new TGeoRotation();
rot_glob->RotateY(90.0);
rot_glob->RotateZ(-90.0);
rot_glob->RotateY(fGladAngle); //<--- look here!
rot_glob->RotateZ(180.0);
```

and then applied.
This happens anyhow, and overrides the rotation read from the geometry file by R3BModule::ConstructRootGeomtry(), which is called a few lines above. Thus, if fGladAngle is 0, a rotation with 0 degrees along -Y is applied, discarding the one defined in the macro (and in the geometry file generated by it).
This can also lead to even funnier behaviour if the compiler doesn't bother default-initalizing fGladAngle to 0 when the "just by name constructor" is called --as for now, GCC does bother, but other compilers may not and I'm not really aware of any strong guarantee about GCC continuing to bother, but I didn't look.

Dmytro Kresan wrote on Thu, 22 June 2017 07:26
I recommend usage of "glad_v17_flange.geo.root" geometry file - the closest to real geometry we have in the cave.

Ok, thanks. Question: is that binary generated with the latest macro? I see that regenerating the geometry with "$R3BROOT_HOME/macros/r3b/geo/create_glad_geo.C" GLAD appears with its flange.

Dmytro Kresan wrote on Thu, 22 June 2017 07:26
Here is how the constructor is called in run_sim.C (new simulation macro, was also announced through the mailing list):
[...]
And all this works properly in the central dev branch.

Yes, I saw that constructor and yes, that does work, thanks.
But it will not work with code written before commit f3eb7099988ca632cfdd6c83a3aacbb659b4709a, which will call the "just by name" constructor which is, as of now, broken. One solution is to add the initialization of "fGladAngle" to that constructor as well (which, as I said, fixes it. Regardless of the specific geometry file being used), another is to remove the constructor altogether (and maybe declare it as deleted, so code that calls it will give a useful compilation error).

Thanks a lot for your help.
Best regards,
Lorenzo.