

Hi,
only today I can answer to this post.

Quote: I got the following results with the bump analysis qa macro provided by Dima:

First of all, which qa macro have you used to produce those plots? So that I can also try to reproduce the distributions.

Quote: My assumption is that something is not properly initialized in the new EMC geometry setup (e.g. EmcMapper, TwoCoordinateIndex, etc.) and that the bump splitting algorithm works fine.

When the new geometry for the endcap was put inside svn, Ola has showed plots with cluster residuals for energy and angular distributions, and they looked fine. For this reason I think that the mapping should be ok, considering that the clusterization depends on the map. Maybe there is something just after the cluster and before the bump, one should investigate.

Quote: 1. Is there a tool available with which one can easily check whether the crystal neighbour list, TwoCoordinateIndex, etc. are o.k.?

The EMC reconstruction code was substantially ported from the fast simulation framework, thus you could use all your tools to check the mapping in pandaroot, the classes are practically the same.

Which tools have you developed, so that we can also implement them in pandaroot? In general one can use the GetXPad()/GetYPad() functions and correlate them to the x/y/z. But a real "tool" was never developed by us.

Quote: 2. All the geometry information will be handled via the singleton objects...
[cut]

Could you please show an example on how the structure could be messed up? In theory the mapper is needed only for the reconstruction tasks, when looping inside the data after the reconstruction the mapper could only screw the TwoCoordinateIndex objects, but at this stage they are not needed anymore. So maybe I have not well understood which could be the possible problems. And if I remember well, the Instance(0) are called only when no previous mapping is defined and then the code does not know what to use. Maybe Dima can comment on it.

Quote: 3. Would it make sense to introduce an EMC initialization sequence where everything will be initialized centrally at one place.

In theory the function `static PndEmcMapper* Instance(Int_t, TString geoName="");` is done with this purpose, to setup the environment only one time. Or am I missing something?

Quote: 4. Does it make sense to introduce a global Emc environment class where all global emc environment objects are collected?

I have still not understood well what do you call "environment objects". In theory there is just

the map and all the correlated classes. I know the structure of the code is a bit complicated, but we have preferred to just port the code and to not modify it a lot.

Which are exactly your suggestions?
