Subject: Genfit Reco Task
Posted by StefanoSpataro on Thu, 18 Feb 2010 16:42:43 GMT
View Forum Message <> Reply to Message

Dear all,
I have done some work around genfit, in order to unify all the existing kalman tasks, to have all the genfit stuff in one package separated from the detectors.

I have created a folder GenfitTools/recotask, which creates the library libRecoTasks.

In this folder, at present, there are two classes:

PndRecoKalmanFit -> single track fitter
PndRecoKalmanTask -> task for track fitting

PndrecoKalmanTask substitutes PndLheKalmanTask, which soon will disappear.
PndrecoKalmanTask is fitting a TCA of PndTrack, creating a new TCA of PndTrack. The user can set the names of the input TCA and of the output. This can be used with all the possible tracking code using PndTrack and the standard definition of detectors in PndDetectorList (mvd tpc stt gem dch mdt).
Macros in macros/pid are already updated. You can check there how to use it.

PndRecoKalmanFit is able to do the kalman fit track by track.
In order to use it, you have to create it, init, and then to fit a PndTrack object:

```
PndRecoKalmanFit *fitter = new PndRecoKalmanFitter();
if (!fitter->Init()) continue;

for(Int_t itr=0;itr<ntracks;++itr)
 {
   PndTrack *prefitTrack = (PndTrack*)fTrackArray->At(itr);
   Int_t  fCharge= prefitTrack->GetParamFirst().GetQ();
   Int_t PDGCode= -13*fCharge;

   PndTrack *fitTrack = new PndTrack();
   fitTrack = fFitter->Fit(prefitTrack, PDGCode);
   if (fitTrack==NULL) continue;
   ...
 }
```

This is a way on how to use the code with a muon hypothesis. PndRecoKalmanTask is just a for loop retrieving PndTrack objects and fitting them by PndRecoKalmanFit.

What is important is that:
a) nobody should take care of the genfit conversions and of the recohitfactory anymore;
b) one can run a single track fitter, without being forced to process all the tracks inside a task (the job of the PneRecokalmanTask;
c) one can change the particle hypothesis, even refitting according to something different from the hardcoded muons, i.e. after PID.

Once all the other macros will be updated, I will remove the old PndLhekalmanTask.

I wait for some feedback.

Subject: Re: Genfit Reco Task
Posted by Sebastian Neubert on Thu, 18 Feb 2010 17:46:45 GMT
View Forum Message <> Reply to Message

Hi Stefano!

That sounds great!
I'll have a look at that on the next occasion.

Cheers! Sebastian.

Subject: Re: Genfit Reco Task
Posted by StefanoSpataro on Thu, 18 Feb 2010 17:52:02 GMT
View Forum Message <> Reply to Message

Thanks,
any comment could be helpful.

Subject: PndRecoMultiKalmanTask
Posted by StefanoSpataro on Mon, 15 Mar 2010 17:45:22 GMT
View Forum Message <> Reply to Message

Dar all,
in GenfitTools/recotasks you can find a new kalman task, PndRecoMultiKalmanTask.

This is working like the PndRecoKalmanTask, but it is fitting the track according all the 5 particle hypothesis, and filling 5 different TCloneArrays of PndTrack.

At the moment the task is meant for tests, not completely embedded in the reconstruction framework. Tomorrow some diagnostic plots.

Subject: Re: PndRecoMultiKalmanTask
Posted by StefanoSpataro on Wed, 17 Mar 2010 11:25:20 GMT
View Forum Message <> Reply to Message

In order to check PndRecoMultiKalmanTask, I run 500 kaon events at 500 MeV/c.

The following plot shows the reconstructed momentum distribution, under different particle hypothesis (MVD+TPC+GEM):

The red dotted line is the "real" value. You can see that the kaon hyp gives the best results, the pion/muon (almost identical) are missing the momentum of a bit, and have worse resolution, electrons and protons have very bad results.
From this plot it seems, then, that we need to refit properly the track with the correct particle hyp, at least at 0.5 GeV/c.

I have thought that maybe the chi2/ndf distribution could be helpful trying to find the best particle hypothesis, then I have plotted chi2/ndf for the same events:

You can see that:
a) The distributions are almost the same, no clear difference between different particles, even if we have seen protons and electrons are very bad! I would expect a worse value for the wrong hypothesis
b) the distributions are peaked at 0.5, and not at 1 ?
c) Running 500 events, the code reconstructs 750 tracks, but there are almost 200 tracks with NDF==0. I can understand that they could be failures, but the track is filled and not empty, and they have "reasonable" momentum. I am wondering why NDF is 0 and why, eventually if bad, the tracks are not rejected. The flag of those tracks is always zero, then I suppose there is no propagation of fit "quality" in the genfit adapter.

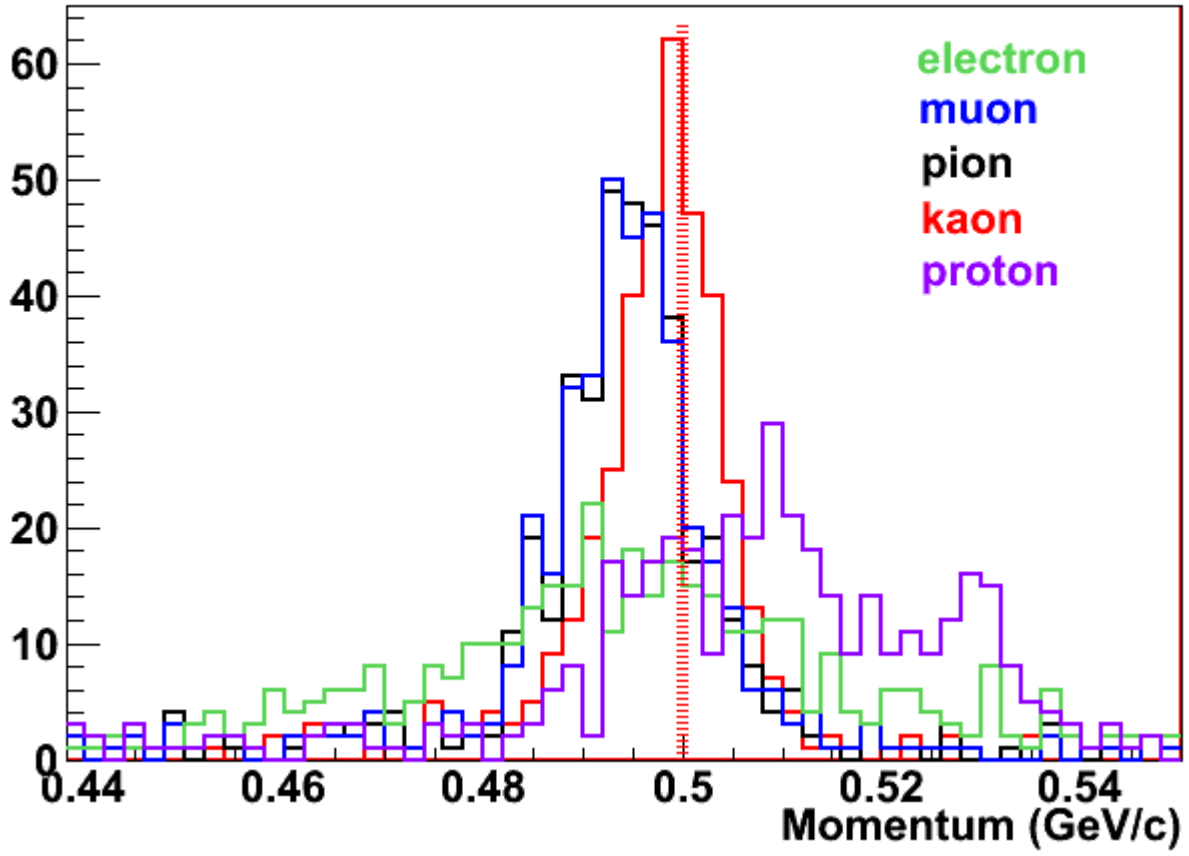
Trying to summarize:

A wrong particle hypothesis provides bad resolution at 500 MeV, and this could also give problems in the correlation to the other pid detectors.
It is not clear to me how reliable is the chi2 calculation from genfit, because of the wrong position, the absence of differences under different particle hypothesis and even the number of degrees of freedon equal to zero for tracks which seems almost good.

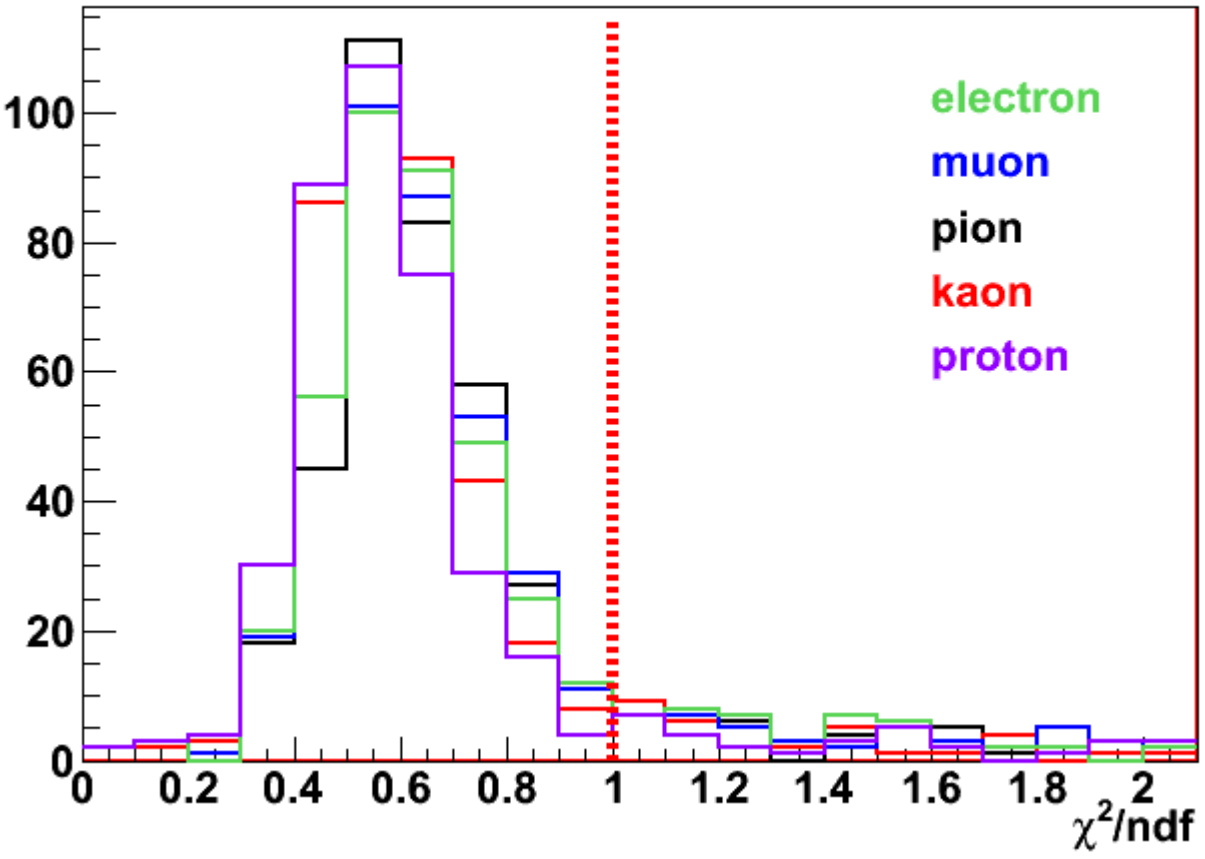
Comments from experts are welcome.

File Attachments

1) mom_pid.gif, downloaded 1546 times



2) chi2ndf_pid.gif, downloaded 1385 times



Subject: Re: PndRecoMultiKalmanTask
Posted by Anonymous Poster on Wed, 31 Mar 2010 15:31:09 GMT
View Forum Message <> Reply to Message

Hi Stefano and others,

I spent a few hours getting up to speed on what is going on here. First of all: I really appreciate your effort in making a common fitting task for everybody!

Concerning your results:

- The fact that the chi2/nfd distributions peak at 0.5 is due to the fact that we do not estimate the resolution on the TPC (and possibly others) reco hits correctly. The smearing of TPC clusters depends on the full realistic digitization chain. The exact results for the point resolutions are not known at the moment. However, I can verify that scaling the point resolution by a constant factor of 0.8 makes the distribution peak at ~1. At this point this can not be considered as a major problem, but it will be subject to optimizations by all detector code developers, responsible for putting the right measurement errors into the corresponding reco hits. I can say that in the TPC it looks like we are already within ~20%, which is not so bad.

- The tracks which have NDF==0: I could reproduce this effect. I found out that it is due to the fact that all (or almost all) hist in these track failed int he Kalman filter, meaning in particular that the GEANE extrapolation had failed for all hits. You can see that if you get rid off the GFException::quiet() switch. The way to catch this, is to evaluate the number of failed hits. You can access it by

GFTrack::getFailedHits(int repId)

If you do not pass a repId, it defaults to the cardinal track representation. You might want to pass this info into PndTrack and to the adapter function.

I didnt check, why all the hits fail for these tracks. But I guess it could be due to the fact that the starting values are really wrong or that the starting point maybe is already much behind the first hit or so. If you have further problems in finding out what the problem is, I can try to help.

- about separating different particle hypothesis from chi2/ndf: This would be very interesting for PID purposes. I am not surprised that the chi2/ndf distributions look similar. The difference in chi2 must be very small for each fit. But it would be very interesting to know:
Is the chi2/ndf for the right hypothesis (if only slightly) better that for the others? Maybe you could plot (chi2/ndf_kaonHyp - chi2/ndf_pionHyp) for you fits. I am eager to see that!
Maybe it might be better to actually use the chi2 probability for cuts or evaluation instead of chi2/ndf. You can get it in ROOT via TMath:Prob(chi2,ndf). It should be better especially if ndf differs a lot between fits.

Cheers, Christian

Subject: GFTrack NDF == 0
Posted by StefanoSpataro on Tue, 20 Apr 2010 15:11:00 GMT
View Forum Message <> Reply to Message

Hi,

I have modified the adapter and the fit class in order to use a "flag" convention PndTrack::GetFlag() :

flag 1 - good track
flag 0 - (default value from constructor)
flag -1 - NDF==0
flag -2 - failure in PndGenfitAdapter
flag -10 - pz < 1e-9 (no propagation possible)

If flag = -10 or -2, the results of the fit is the "prefit" track. If flag = -1, then the result is the already fitted track.

I have simulated 500 muons at 1 GeV, tpc+mvd from 20° to 120°.
This is the flag distribution:

Many tracksat 1, but tracks also at -1 and -2.

Here I plot the flag distribution versus number of hits belonging to the track:

As you can see, -1 (ndf==0) tracks have a high number of hits, they seem "good" tracks, and not just bad tracklets.

I plot here the momentum distribution of the "prefit" value for tracks with flag==-1:

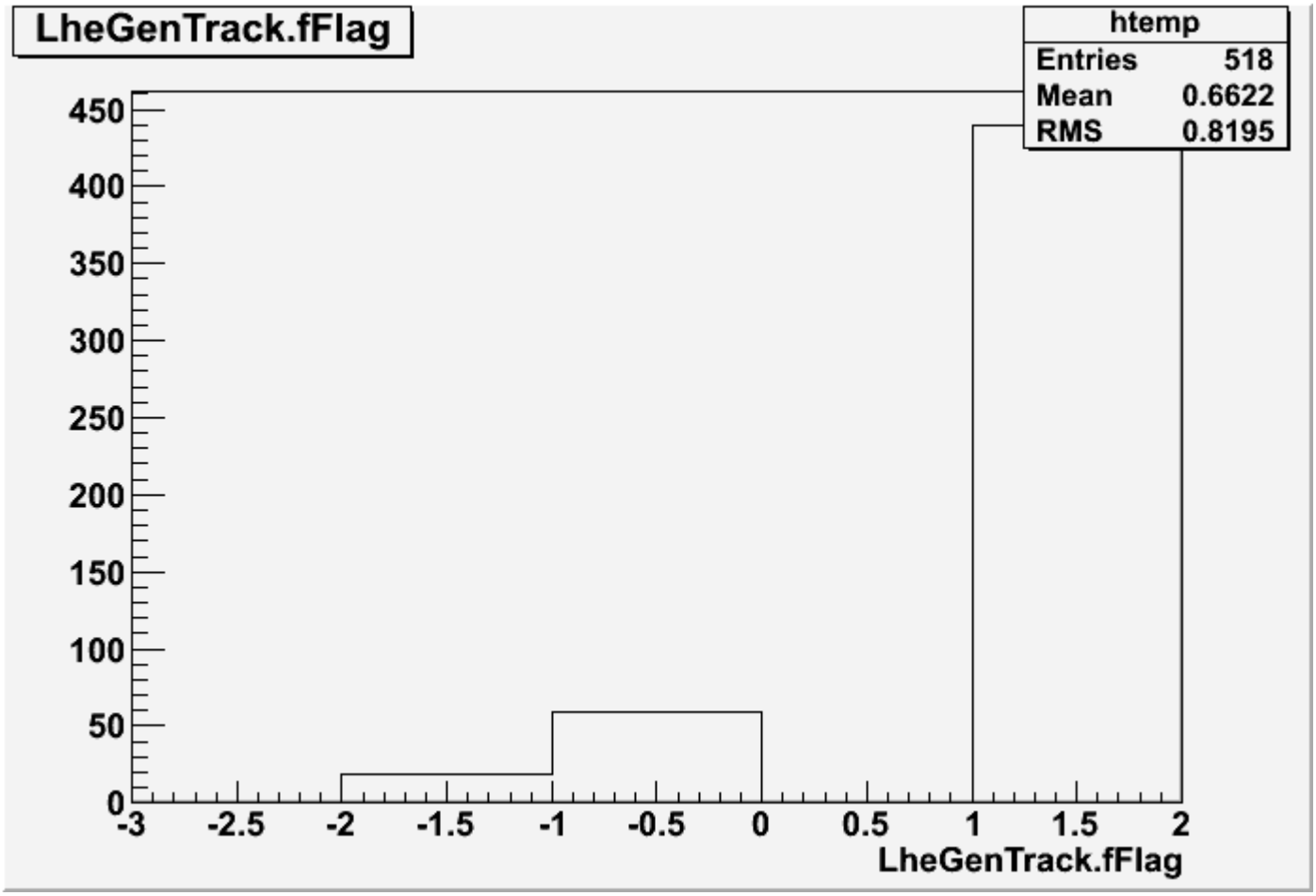
You can see these tracks have a correct momentum, and even theta seems reasonable. This means that, maybe, the ndf==0 is not connected to wrong initial parameters given by lhe.

A systematic study about the source of this problem should be done, and also the "volunteer" should be found. Maybe they are simply bad tracks, but it could be also possible that we are losing good tracks simply fitting them.

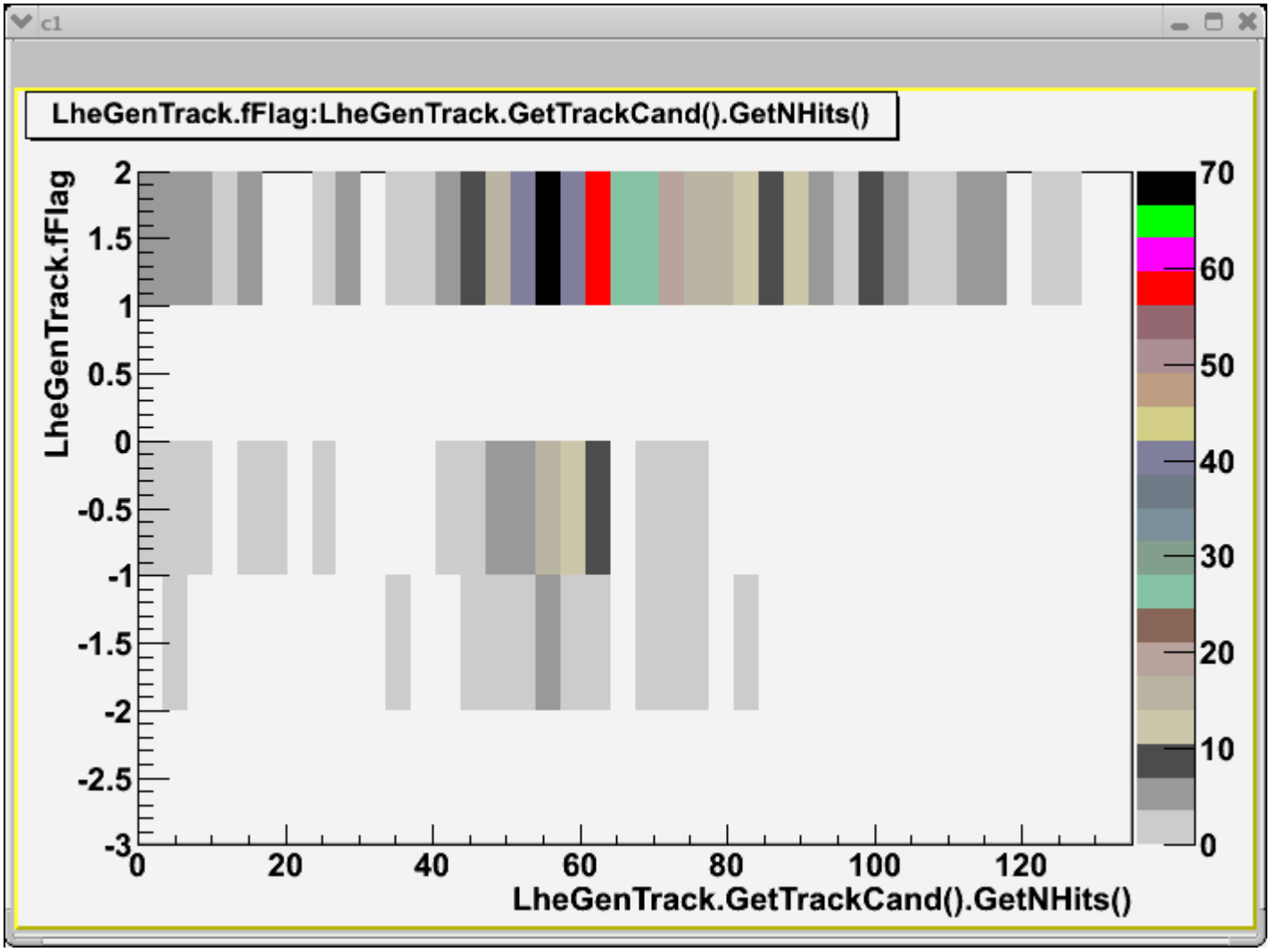
it is also not clear to me why I have -2 tracks, why the adapter is failing, but this is a second order problem, I think.

File Attachments

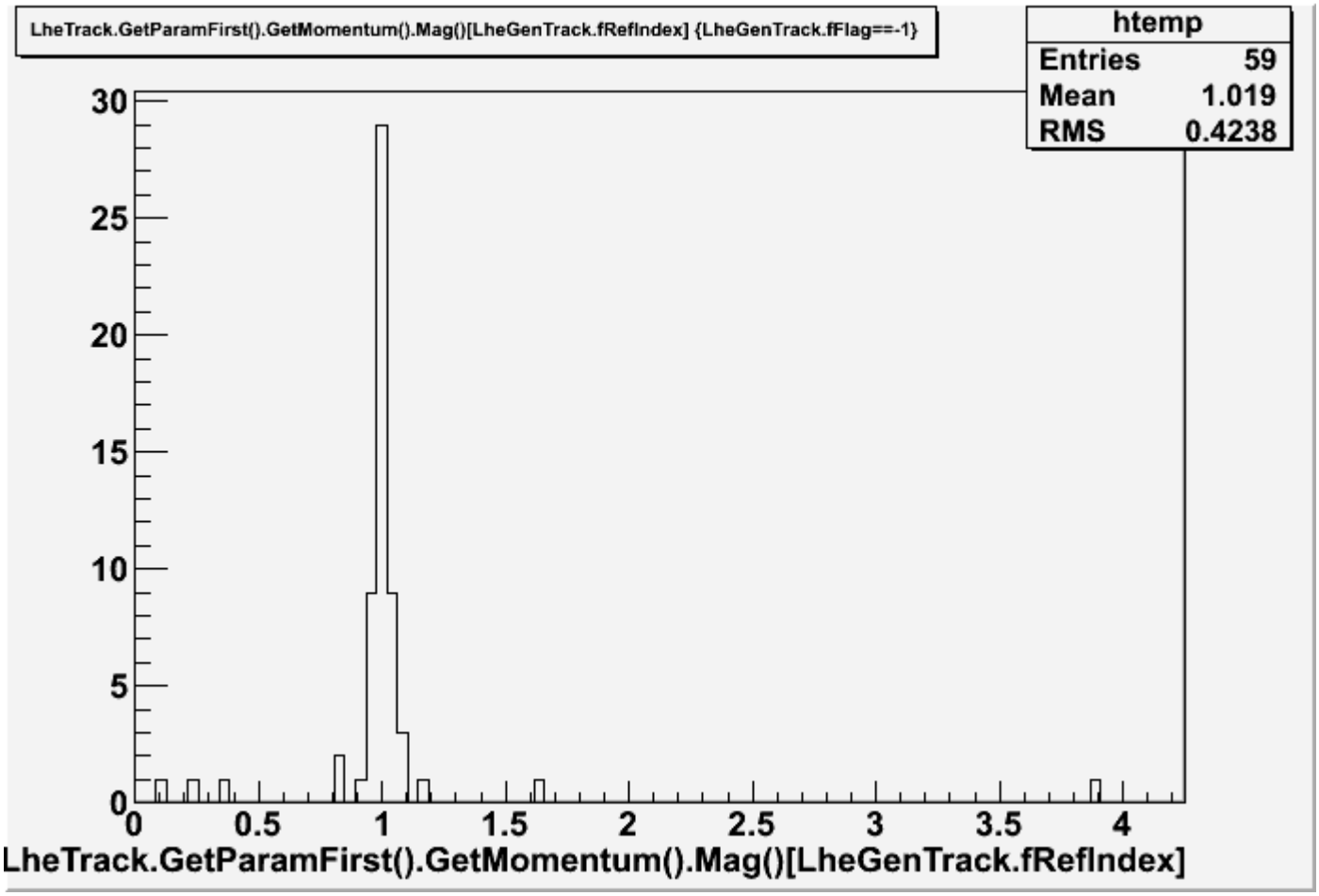
1) gen_flag.gif, downloaded 1299 times



2) gen_flaghits.gif, downloaded 1323 times



3) lhe_flag-1_prem.gif, downloaded 1234 times


LheTrack.GetParamFirst().GetMomentum().Mag()[LheGenTrack.fRefIndex] {LheGenTrack.fFlag==-1}
htemp
Entries 59
Mean 1.019
RMS 0.4238
30
25
20
15
10
5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
LheTrack.GetParamFirst().GetMomentum().Mag()[LheGenTrack.fRefIndex]