

Fitter update



Marcin Chrzaszcz
mchrzasz@cern.ch

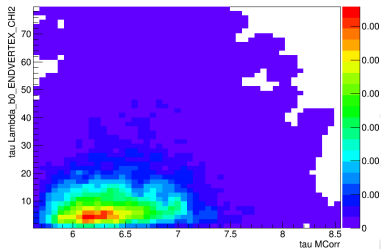
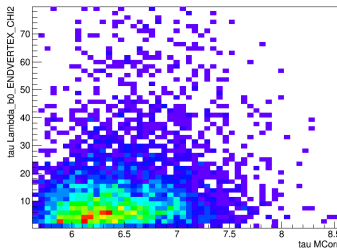
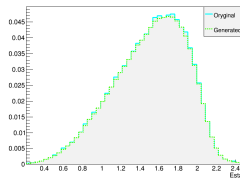
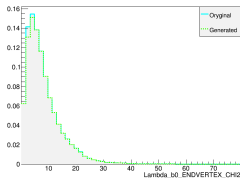
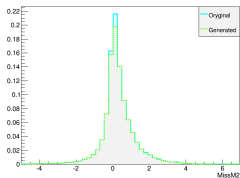


University of
Zurich^{UZH}

$R(\Lambda_c^*)$ meeting, CERN
June 20, 2016

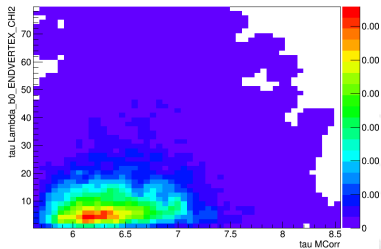
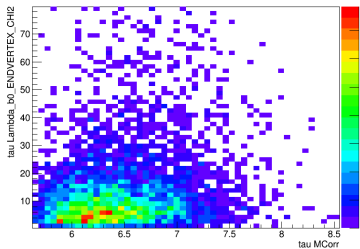
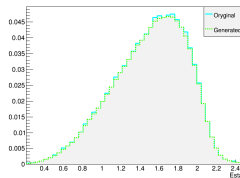
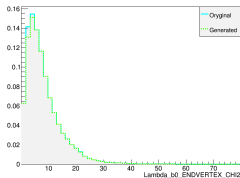
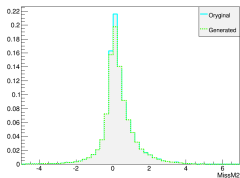
Reminder

⇒ We use the Kernel Density to get the pdf to simulate 50M events for each sample.



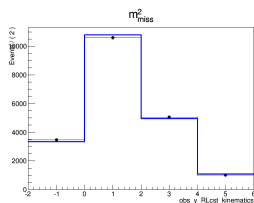
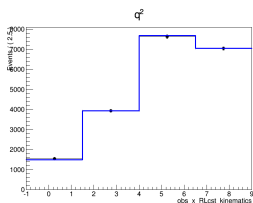
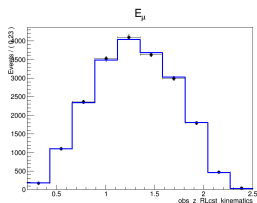
Reminder

⇒ We use the Kernel Density to get the pdf to simulate 50M events for each sample.



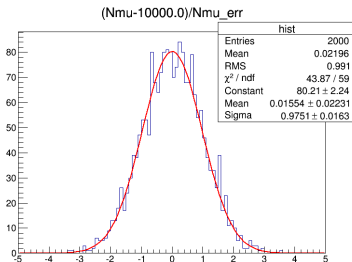
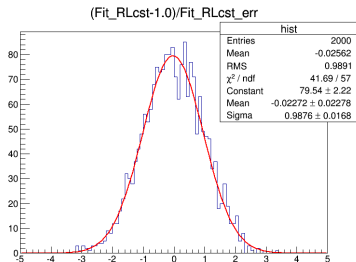
Initial parameters

- ⇒ Template size : 1000000 (for each μ and τ samples).
- ⇒ Each toy: 10000 (for each μ and τ samples).
- ⇒ Because of this obviously: $R(\Lambda_c^*) = 1$
- ⇒ Fits are done in 3-dim: $M_{\text{miss}}^2, E_{\mu}^{\text{CM}}, q^2$
- ⇒ The number of bins: 10, 10, 4 (ad hoc).



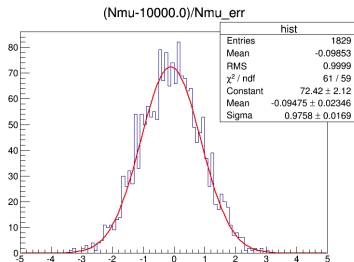
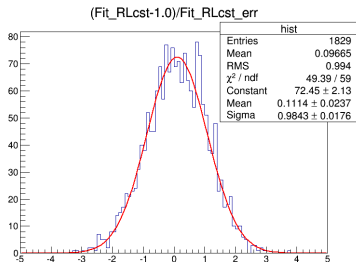
Results 1

⇒ If I use a binned template to generate histograms:



Results 2

⇒ If I use a kernel density pdf to generate toys:



Conclusions

- ⇒ So fit works!
- ⇒ The observed bias can be because of three things:
 - The bins are biasing the results (will study different binning options and observed diff).
 - Technicality: Because of the stupid `HistFactory` structure I am supplying the data in a different way. This I can change now.
 - The kernel density generation introduces a bias (hope now as I don't want to debug the internal scipy library).
- ⇒ I wrote this quite efficiently: 3k events < 5 min.

