Subject: [CLOSED] Delta from Krivoruchenko
Posted by Ingo Fröhlich on Fri, 07 Oct 2011 08:47:52 GMT
View Forum Message <> Reply to Message

Dear all,

please update to v5.38.2. In the previous version, the Krivoruchenko model was not enabled by using the following command:

makeDistributionManager()->Exec("dalitz_mod: krivoruchenko");

It is very important to check the output of the reaction, it should print:

[D+_krivoruchenko] dgdm from Krivoruchenko {/}

some more word to the calculation of branching ratios. By default, Pluto does not calculate the branching ratio, but takes as a weight 1/N_ev * BR, where BR is the static branching ratio. This leads of course to incorrect results at the pole, if the branching ratio shows a large mass-dependency.

The only way to let Pluto calculate the branching ratio, is the "Monte-Carlo-Integration method". In this case one has to use a flat dilepton mass generator:

makeDistributionManager()->Exec("dalitz_mod: static_br_thresh=0.100 ; flat_generator");

By using this, the another model is added:

[D+_generator_p_dilepton] Dilepton generator {/generator}

I attached below a macro which uses this method.

Toggle Spoiler

```
   makeDistributionManager()->Exec("elementary");
   makeDistributionManager()->Exec("dalitz_mod: krivoruchenko");
   makeDistributionManager()->Exec("dalitz_mod: static_br_thresh=0.100 ; flat_generator");

   PReaction *my_reaction =
new PReaction("3.5","p","p","p D+",NULL,1,0,0,0);

   TH1F * pp_sum =
new TH1F ("pp_sum","pp DiLepton mass (coherent sum)",100,0.,0.6);
   pp_sum->Sumw2();
   TH1F * delta =
new TH1F ("delta","D mass ",100,0.9,1.6);
   delta->Sumw2();


   my_reaction->Do(pp_sum,"_x=[dilepton]->M()");
   my_reaction->Do(delta,"_x=[D+]->M()");
   my_reaction->Print();
```

```
   my_reaction->Preheating(100);
   my_reaction->Loop(100000);

   PReaction *my_reaction2 =
new PReaction("3.5","p","p","p D+ [dilepton p]",NULL,1,0,0,0);

   TH1F * pp_sum2 =
new TH1F ("pp_sum2","pp DiLepton mass (coherent sum)",100,0.,0.6);
   pp_sum2->Sumw2();
   TH1F * delta2 =
new TH1F ("delta2","D mass ",100,0.9,1.6);
   delta2->Sumw2();

   my_reaction2->Do(pp_sum2,"_x=[dilepton]->M()");
   my_reaction2->Do(delta2,"_x=[D+]->M()");
   my_reaction2->Print();

   my_reaction2->Preheating(100);
   my_reaction2->Loop(100000);

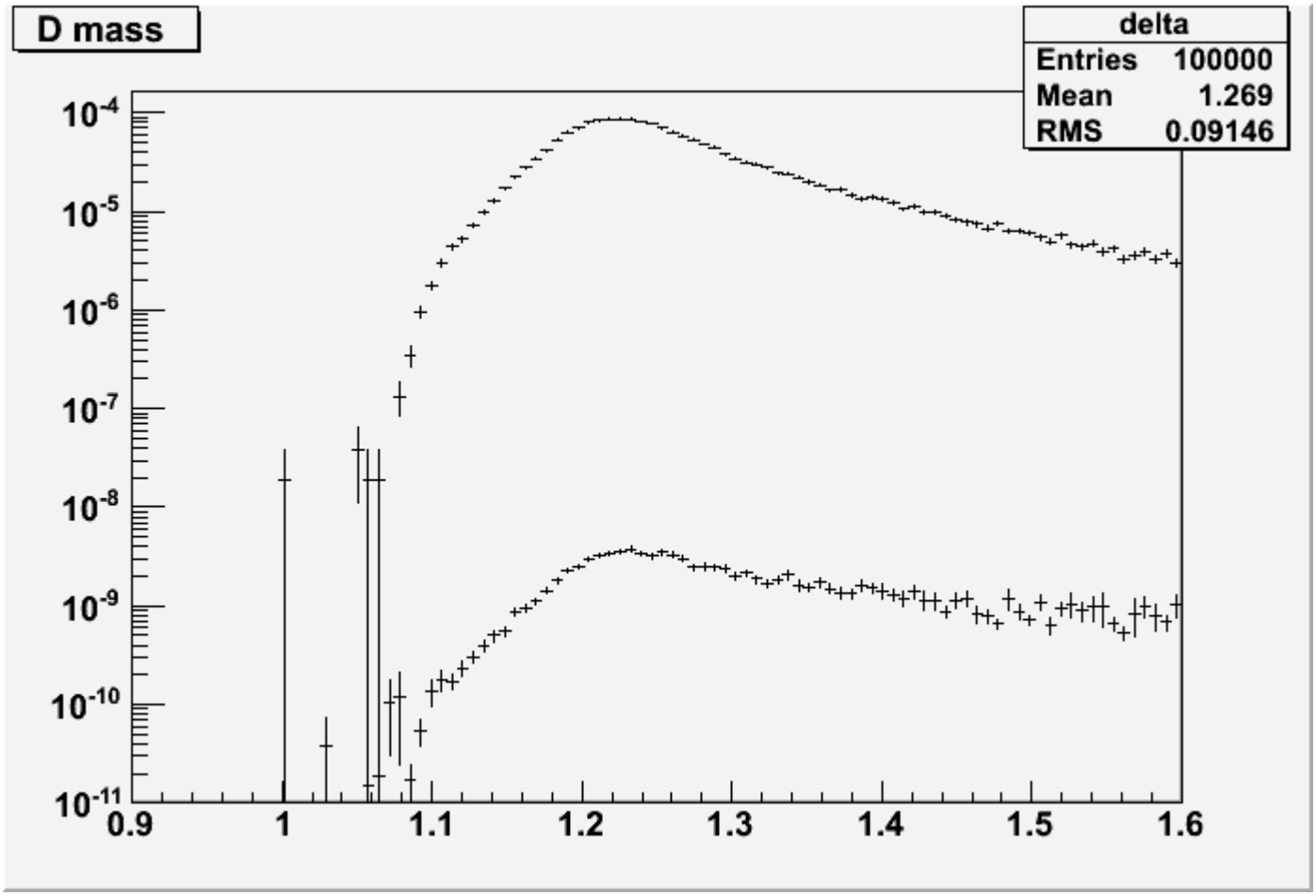
   //    delta2->Divide(delta);
   delta->Draw("");
   delta2->Draw("same");[/QUOTE]
```

The result is shown below:

and this is the branching ratio which I obtain:

File Attachments

1) kriv.png, downloaded 2031 times



2) c1.png, downloaded 1990 times

