Subject: Re: Changing t distribution slopes
Posted by Ingo Froehlich on Sun, 14 Aug 2011 20:34:31 GMT
View Forum Message <> Reply to Message

OK, I played a little bit with a new source code, which I uploaded here:

http://www-linux.gsi.de/~hadeshyp/pluto/v5.38.1/pluto_v5.38.1.tar.gz

It is a test version, not yet finalized, and without warranty, because I had to change some parts in the scripting class, which needs some severe testing from my side. But for a first test it should be fine.

Below I attached a macro for your purpose with some comments (it works only with the new patched version):

Toggle Spoiler

```
   //First, we create the general purpose distribution
   //model:
   PAnyDistribution* decay =
      new PAnyDistribution("t_slope",
      "A function to add a new t-slope");
   decay->Add("q,      parent");
   decay->Add("p,      daughter");
   decay->Add("rho0,  daughter");
   //This is the cache for the undistorted data
   //It is needed because the mandelstam t is a non-uniform
   //distribution. The size and binning of the cache must
   //be chosen such, that during runtime (better: during Preheating) the statistic
   //is sufficiently filled
   TH1F * cache  = new TH1F ("cache","Rho0 t cache",400,-4.0,.0);


   //For the following we we have to know that all particles (but the daughters) are in the parent rest frame
   //the daughters are in their rest frame (i.e. the parent)
   //therefore we have to boost into the parent frame
   //the parent is indicated by "_parent"
   //N.B.: "t" is reserved in TFormula, do NOT use it
   decay->AddEquation(cache,"beam = _parent->GetBeam(); beam->Boost(_parent) ; t1 = (beam - [rho0])->Mag2(); _x = t1;");

   //This is the final equation. The distribution (the probability function)
   //must be stored in "_f"
   decay->AddEquation("_f = exp( 8*t1 );");

   //Remember, AnyDistribution is a rejection method. Therefore
   //it can happen, that parts of the phase space, where _f has a
   //large probability, is not well populated by the generated events
   //In this case, the event loop will run forever, as Pluto tries
   //to match the shape defined by _f.
```

```
//The following factor is the maximum enhancement factor to avoid such
//deadlocks.
//N.B.: It directly scales with the computing time!!!
decay->SetMaxEnhancementFactor(10);
//Hint: in this configuration, the sampling of 100kEvent takes ~30min

//Add this model to the Pluto data base:
makeDistributionManager()->Add(decay);

//Construct the reaction, as usual:
PReaction my_reaction("_T1 = 2.2","g","p","p rho0 [pi+ pi-]");

TH1F * histo1 = new TH1F ("histo1","rho0 t",100,-4,0);
TH1F * histo3 = new TH1F ("histo3","cos theta of rho0",50,-1.,1.);

my_reaction.Do(histo1,"beam = [g+p]->GetBeam(); t1 = (beam - [rho0])->Mag2(); _x = t1;");
my_reaction.Do(histo3,"_rho=[rho0]; _rho->Boost([g+p]); _x= cos(_rho->Theta())");


my_reaction.Print();   //The "Print()" statement is optional

//Make a dummy loop to fill the AnyDistribution with some statistics:
my_reaction.Preheating(100);

my_reaction.Loop(1000);
```

So the basic idea is that one is able to add an individual equation.

Before I move on, I would like to know your opinion.

Greetings, Ingo