

Greetings,

I had ran a code of mine that originally used v5.40 on v5.40.5, and did not get the same behavior for PAnyDistribution.

I see no reason for it in the .cc or .h files. I also re-compiled v5.40.5 and got the same result.

Below is v.5.40 t-distribution.

Below is v.5.40.5 t-distribution with same code.

Here is the code used to generate

```
//#include "loadPluto.h";  
//Program to generate multiple PLUTO root file  
//Input of number of entries for specific energy is required (see list.dat)  
//list.dat is a bremstrahlung generated spectrum  
//Author Michael C. Kunkel  
  
#include "/u/home/mkunkel/PLUTO/pluto_v5.40.5/src/PParticle.h"  
#include "/u/home/mkunkel/PLUTO/pluto_v5.40.5/src/PReaction.h"  
#include "/u/home/mkunkel/PLUTO/pluto_v5.40.5/src/PBeamSmearing.h"  
#include "/u/home/mkunkel/PLUTO/pluto_v5.40.5/src/PAnyDistribution.h"  
  
/*  
#include "/w/hallb/clasg12/mkunkel/PLUTO/pluto_v5.40/src/PParticle.h"  
#include "/w/hallb/clasg12/mkunkel/PLUTO/pluto_v5.40/src/PReaction.h"  
#include "/w/hallb/clasg12/mkunkel/PLUTO/pluto_v5.40/src/PBeamSmearing.h"  
#include "/w/hallb/clasg12/mkunkel/PLUTO/pluto_v5.40/src/PAnyDistribution.h"  
*/  
void Rho_T(Double_t energy, Int_t entries){  
    //gSystem->Load("/u/home/mkunkel/PLUTO/pluto_v5.40.5/libPluto.so");  
  
    cout.precision(2);  
    char nam1[5] = "Rho_";  
    char nam2[3] = "_t";  
    char nam3[8] = "_T1 =";  
    char c[10];  
    sprintf(c, "%3.2f",energy);  
    char name[60];  
    char setE[60];  
    sprintf(name,"%s%s%s",nam1,c,nam2);
```

```

sprintf(setE, "%s%s", nam3, c);
cout<<name<<"\t"<<setE<<endl;

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");

TH1F * cache = new TH1F ("cache", "Rho0 t cache", 400, -4.0, .0);

decay->AddEquation(cache, "beam = _parent->GetBeam(); beam->Boost(_parent) ; t1 =
(beam - [rho0])->Mag2(); _x = t1;");

decay->AddEquation("_f = exp( 7.1*t1 );");

decay->SetMaxEnhancementFactor(10);

makeDistributionManager()->Add(decay);

//Construct the reaction, as usual:

PReaction my_reaction(setE, "g", "p", "p rho0 [pi+ pi-]", name);

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, "beam2 = [g+p]->GetBeam(); t12 = (beam2 - [rho0])->Mag2(); _x =
t12;");
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(entries);
}

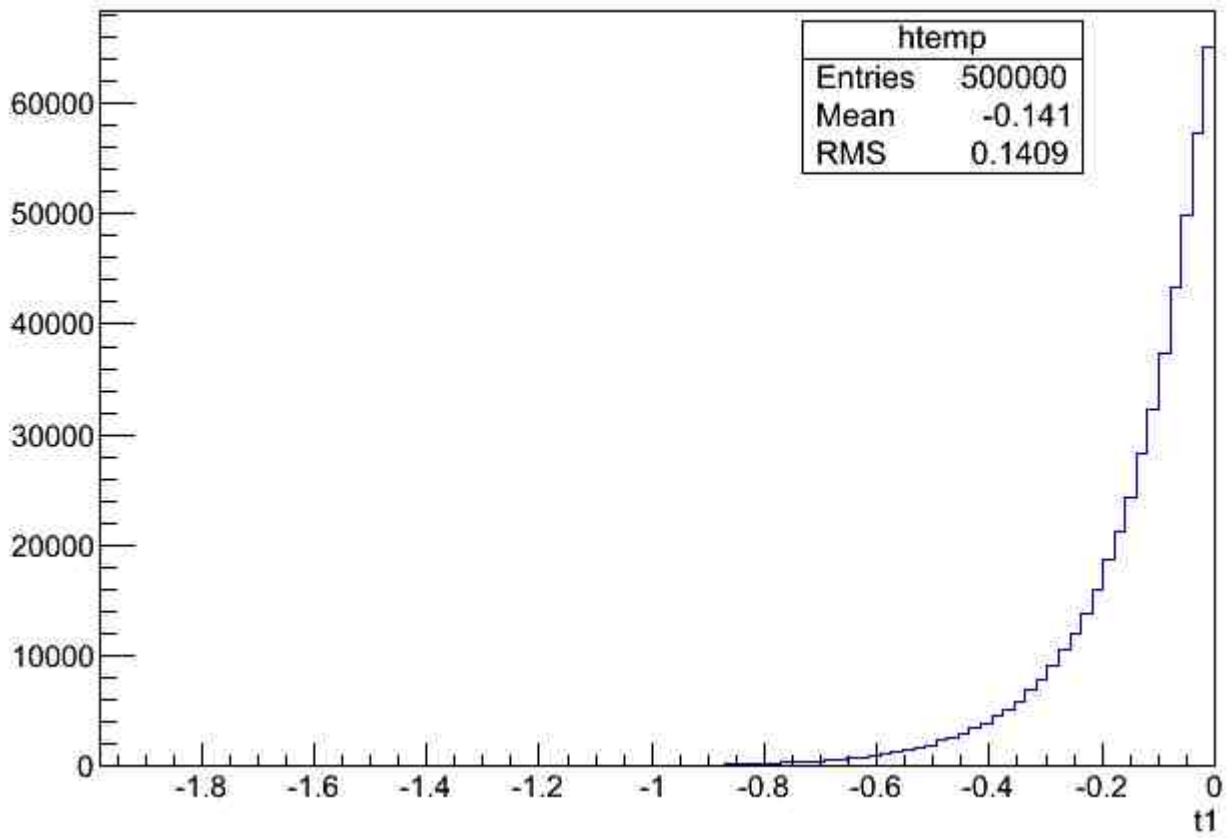
```

---

## File Attachments

1) [t\\_v\\_5\\_40.jpg](#), downloaded 1037 times

t1



2) [t\\_v\\_5\\_40\\_5.jpg](#), downloaded 1099 times

t1

