Subject: PndBoxGenerator
Posted by StefanoSpataro on Tue, 22 May 2007 16:59:16 GMT
View Forum Message <> Reply to Message

After some discussions I wrote a new box generator inside pgenerators directory: PndBoxGenerator.

PndBoxGenerator is like CbmBoxGenerator, but I put one function in order to set uniform distributions in cos(theta), and not in theta as it is done by default.

Example:

If you want to have a unifor distribution i theta, you have to type in your simulation macro:

```
PndBoxGenerator* boxGen = new PndBoxGenerator(13, 1);
boxGen->SetPRange(1.,1.); // GeV/c
boxGen->SetPhiRange(0., 360.); // Azimuth angle range [degree]
boxGen->SetThetaRange(0., 180.); // Polar angle in lab system range [degree]
boxGen->SetXYZ(0., 0., 0.); // vertex coordinates [cm]
primGen->AddGenerator(boxGen);
```

IF you want to have a unifor distribution in cos(theta):

```
PndBoxGenerator* boxGen = new PndBoxGenerator(13, 1);
boxGen->SetPRange(1.,1.); // GeV/c
boxGen->SetPhiRange(0., 360.); // Azimuth angle range [degree]
boxGen->SetThetaRange(0., 180.); // Polar angle in lab system range [degree]
boxGen->SetCosTheta(); // Set uniform ditribution in cos(theta)
boxGen->SetXYZ(0., 0., 0.); // vertex coordinates [cm]
primGen->AddGenerator(boxGen);
```

And that's all.
Enjoy...

Ste

Subject: Re: PndBoxGenerator -> distributions uniform in cos(theta)
Posted by StefanoSpataro on Tue, 22 May 2007 17:11:57 GMT
View Forum Message <> Reply to Message

To show that the PndBoxGenerator works...

Comparison plots with the two methods:
RED - uniform theta

BLUE - uniform cos(theta)

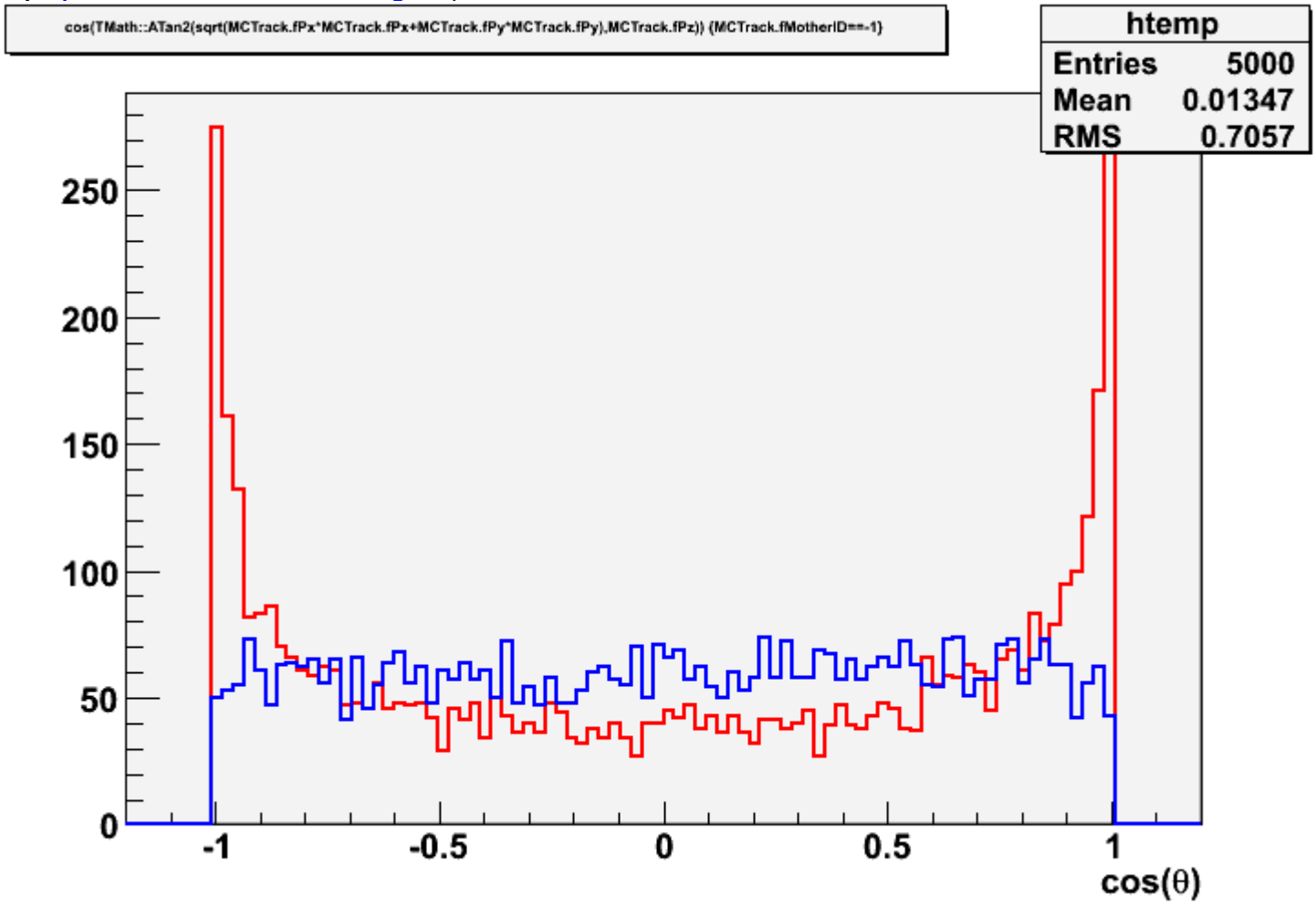
Theta distributions

Cos(Thata) distributions

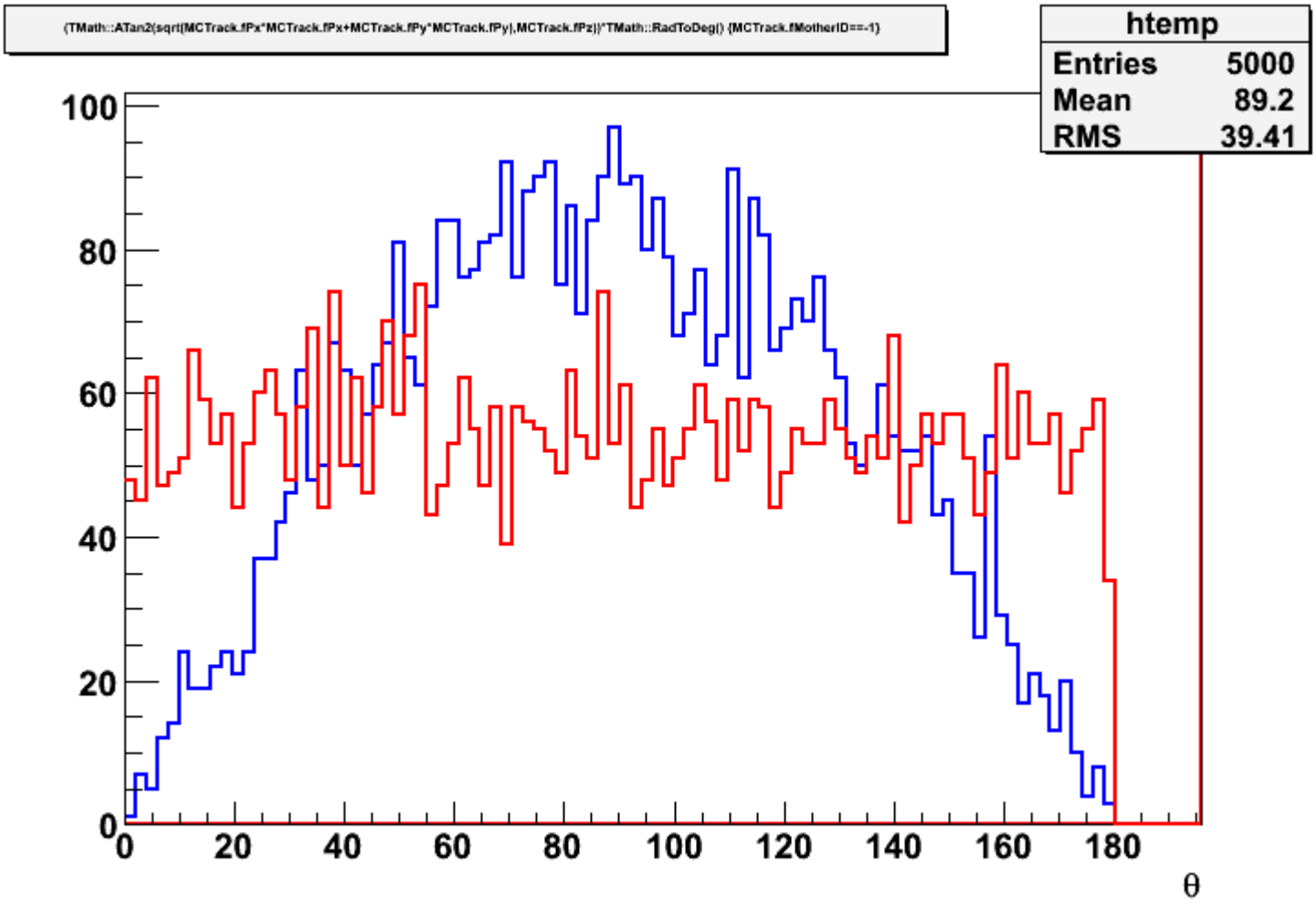
Yes, it works!

File Attachments

1) pndbox_costheta.gif, downloaded 1492 times



2) pndbox_theta.gif, downloaded 1484 times



Subject: Re: PndBoxGenerator -> distributions uniform in 1/p
Posted by StefanoSpataro on Fri, 20 Jul 2007 11:59:10 GMT
View Forum Message <> Reply to Message

Hello,
I added in PndBoxGenerator a new option, in order to have distributions uniform in 1/p (sometimes is can be useful).

It works as the CosTheta function, so (example):

```
PndBoxGenerator* boxGen = new PndBoxGenerator(13, 1);
boxGen->SetPRange(0.1,15.);
boxGen->SetPhiRange(0., 360.);
boxGen->SetThetaRange(1., 12.);
boxGen->SetXYZ(0., 0., 0.);
primGen->AddGenerator(boxGen);
```

generates particles with a uniform distribution opver p in the fixed range, while if you add the boxGen->SetInverseP() function:

```
PndBoxGenerator* boxGen = new PndBoxGenerator(13, 1);
boxGen->SetPRange(0.1,15.);
```

```
boxGen->SetInverseP();
boxGen->SetPhiRange(0., 360.);
boxGen->SetThetaRange(1., 12.);
boxGen->SetXYZ(0., 0., 0.);
primGen->AddGenerator(boxGen);
```

you will have a distribution in the same range BUT uniform in 1/p. It works even with pt range (so uniform in 1/pt).

Enjoy.

P.S. meanwhile I corrected a missing initialization in the constructor, ad sone for CbmBoxGenerator.
