Subject: Re: Bear Smear and Cross Sections
Posted by Ingo Fröhlich on Tue, 28 Aug 2012 06:35:40 GMT
View Forum Message <> Reply to Message

Michael Kunkel wrote on Tue, 28 August 2012 01:32
I wanted to say

Is _f the density function? If so, wouldn't using Input : _x s cos(theta), _y is differential cross section
Output : _f cross section suffice?

No, that's not correct. _x is cos(theta), and _y is the c.m. energy. It's a 2-dimensional function. _f is the results, and this is (dsigma/d(cos(theta)))(q), i.e. the differential cross section as a function of cos(theta) and q

Michael Kunkel wrote on Tue, 28 August 2012 01:32
I have 64 models I will be using. I was assuming I could implement this as

```
  model1->SetRange(1.77,1.8);
...
...
...
  model64->SetRange(2.56,2.6);

  model1->AddHistogram(example1,"value = Eval(_x); _f =_y * value");
  makeDistributionManager()->Add(model1);
  ...
  ...
  ...
  model64->AddHistogram(example64,"value = Eval(_x); _f =_y * value");
  makeDistributionManager()->Add(model64);
```

No, this will not work. Pluto is a sampling event generator. If you use it like this, the first model samples theta and q, the second model overwrites that, and so on...

Michael Kunkel wrote on Tue, 28 August 2012 01:32
In the above snipet I use 1 histogram for each model. Each histogram is derived from published data with

_x = Cos(theta)
_y = Differential Cross section

You are using a different convention, this is part of the confusion. _y is the c.m. energy in a 2-dimensional function. If you are using a 1-dimensional histogram, the results should be still mapped on _f, not _y.

The only thing you have to implement is a function _f = F(_x,_y) = F(cos(theta),q)