HowTo Draw Confidence Regions

Daniel Wegner

June 13, 2011

1 Standard use

In order to draw contour lines representing 90% confidence regions for single EDW events one has to do the following steps:

1. Load the kqpa library.

   ```
   gSystem->Load("$KDATA_ROOT/lib/libkqpa.so");
   ```

2. Make a KQContourPointList object.

   ```
   KQContourPointList aList("QErecoil");
   ```

   or

   ```
   KQContourPointList aList("IonHeat");
   ```

3. Fill the list with points (Q, E_{Recoil}, $\sigma_{E_{ion}}$, $\sigma_{E_{heat}}$) or (E_{ion}, E_{heat}, $\sigma_{E_{ion}}$, $\sigma_{E_{heat}}$)

   (a) Read an ASCII file in the current working directory with lines of the form
   $<Q> <E_{recoil}> <\sigma_{E_{ion}}> <\sigma_{E_{heat}}>$ for mode "QErecoil"
   $<E_{ion}> <E_{heat}> <\sigma_{E_{ion}}> <\sigma_{E_{heat}}>$ for mode "IonHeat"

   ```
   aList.ReadASCIIFile(aFileName);
   ```

   (b) Add events manually

   ```
   aList.AddPoint(aQvalue,anEnergyRecoil,aSigmaIon,aSigmaHeat);
   ```

   for mode "QErecoil"

   ```
   aList.AddPoint(anEnergyIon,anEnergyHeat,aSigmaIon,aSigmaHeat);
   ```

   for mode "IonHeat"

4. Draw the events in an empty frame

   ```
   aList.Draw(anOption = "");
   ```

2 Additional features

The default values of the empty frame ($E_{recoil} = 0..1000 keV$, $Q = 0..2$) can be changed with set methods

```
aList.SetQvalueMax(aNewQvalueMin);
aList.SetQvalueMin(aNewQvalueMin);
aList.SetEnergyRecoilMax(aNewEnergyRecoilMax);
aList.SetEnergyRecoilMin(aNewEnergyRecoilMin);
```

The list of points can be cleared by

```
aList.ClearPoints();
```

and single events can be removed by

```
aListRemovePoint(anIndex);
```

In order to find valid indices the size of the list can be retrieved by

```
UInt_t aSize = aList.GetEntries();
```

Also single events can be created and drawn:

```
KQContourPoint anEvent(aQvalue,
                       anEnergyRecoil,
                       "QERecoil",
                       aSigmaIon,
                       aSigmaHeat,
                       aSigmaIonHeat,
                       aConfidenceLevel,
                       aVoltageBias,
                       anEpsilon,
                       aNumBinsX,
                       aNumBinsY,
                       aNumSigmas;
anEvent.Draw(anOption="");
```

or

```
KQContourPoint anEvent(anEnergyIon,
                       anEnergyHeat,
                       "IonHeat",
                       aSigmaIon,
                       aSigmaHeat,
                       aSigmaIonHeat,
                       aConfidenceLevel,
                       aVoltageBias,
                       anEpsilon,
                       aNumBinsX,
                       aNumBinsY,
                       aNumSigmas);
anEvent.Draw(anOption="");
```

aNumBinsX and aNumBinsY are the dimensions of the distribution histogram from which the contour for the specified confidence level is determined. The default values (1000,1000) should and seem to be sufficient in most cases, but can be increased if needed.

aNumSigmas (default 10) is the number of sigmas calculated by uncorrelated error propagation in E_{recoil} and Q to determine the ranges ($\overline{x} \pm n_{sigmas} \cdot \sigma_x, \overline{y} \pm n_{sigmas} \cdot \sigma_y$ whereas for means the specified or calculated center values for E_{recoil} and Q are assumed) for the contour function (also the draw range if single draw).

In case that there might be changes necessary on the empty frame or the contour function, they can directly be retrieved by

```
TF2* aFunction = aList.GetEmptyFrame();
TF2* aFunction = anEvent.GetFunction();
```