Principles

Notations

Lets note

- E The true energy
- $\tilde{E}$ The reconstructed energy
- θ the zenith angle
- δ the off-axis angle (distance of the event to the center of the camera)

System Effective Area

For a given set of cuts, the system effective area is defined as the integral over the ground of the detection probability $\epsilon(\vec{r}, E, \theta, \delta)$:

- $\mathcal{A}(E, \theta, \delta) = \int dS \times \epsilon(\vec{r}, E, \theta, \delta)$

The effective area is calculated from simulation as function of the true energy, the zenith angle and the off-axis angle.

If you intend to calculate the system acceptance for your own set of cuts, see Spectrum Tables about how to do it.

Energy Resolution

The energy probability density function is defined as the **probability of observing a reconstructed energy $\tilde{E}$ for a given true energy E**.

- $PDF(E, \tilde{E}, \theta, \delta) = P(\tilde{E}|E, \theta, \delta)$

It depends on:

- The zenith angle
- The true energy
- The off-axis angle

Note:

It is in principle possible to add dependency on other observables, such as the reconstructed impact position or the reconstructed shower maximum altitude.

Flux Definition

The source flux depends on the true energy, not the reconstructed one. It is defined as the rate of events per energy interval and per surface unit on ground, and the shape modelisation depends on some parameters α_i:

- $\Phi(E) = \frac{d^3N}{dE\,dS\,dt}(E, \alpha_i)$

Various **spectral shapes** are already implemented in the spectrum module, however, it's not difficult to **implement yours**.

The folding method

The basic idea of the spectrum calculation is to compare the number of excess events in a **reconstructed energy** bin (number which is directly observable) with the expected number, given the spectral shape, energy resolution and system effective area.

Expected number of events

The expected number n_γ of gamma events in a reconstructed energy bin $[\tilde{E}_1, \tilde{E}_2]$ (and for a given zenith angle and off-axis angle) is calculated by the formula:

$$n_\gamma = \int_{\tilde{E}_1}^{\tilde{E}_2} d\tilde{E} \int_0^\infty dE \times \Phi(E) \times \mathcal{A}(E, \theta, \delta) \times PDF(E, \tilde{E})$$

This calculation is performed by the function **Spectrum::SpectrumBase::TheoricRate**.

Log-likelihood comparison

In a given bin in reconstucted energy, zenith angle and off-axis angle, assuming we have:

- N_{ON} the measured number of events in the ON dataset
- N_{OFF} the measured number of events in the OFF dataset
- T_{ON} and T_{OFF} the respective livetimes for the two datasets
- $\beta = T_{ON}/T_{OFF}$ the livetime normalisation
- n_γ and n_h the expected number of gamma and background events in the bin (the number n_γ is for example calculated from the spectrum folded through the detector response)

The probablity of observing N_{ON} and N_{OFF} events when we expect n_γ gamma events and n_h hadrons is given, in Poisson statistics, by the formula

$$P(N_{ON}, N_{OFF}|n_\gamma, n_h) = \frac{(n_\gamma + \beta n_h)^{N_{ON}}}{N_{ON}!} e^{-(n_\gamma + \beta n_h)} \times \frac{{n_h}^{N_{OFF}}}{N_{OFF}!} e^{-n_h}$$

This probability is calculated by the function MathUtils::OnOffFitter::GetLogLikelihood. (See the documentation on the mathutils module).

The best background estimate is given by the value of n_h which maximizes the log-likelihood $L = log(P)$

It's the solution of a second order linear equation in n_h :

$$\frac{\beta N_{ON}}{n_\gamma + \beta n_h} + \frac{N_{OFF}}{n_h} - (\beta + 1) = 0$$

whose solution is given by (MathUtils::OnOffFitter::GetBackground) :

$$\begin{aligned} C &= \beta \times (N_{ON} + N_{OFF}) - (1+\beta) \times n_\gamma \\ \Delta^2 &= C^2 + 4\beta(\beta+1) \times N_{OFF} \times n_\gamma \\ n_h &= \frac{C+\Delta}{2\beta(\beta+1)} \end{aligned}$$

The uncertainties on n_h can be calculated from the second derivative of the log-likelihood:

$$\Delta n_h = \sqrt{\left(\frac{\beta^2 N_{ON}}{(n_\gamma + \beta n_h)^2} + \frac{N_{OFF}}{n_h^2} \right)^{-1}}$$

Fit Procedure

The fit procedure is based on the Levenberg & Marquardt minimization algorithm, which is very efficient in terms of number of function calls (Here, each function call involves the convolution of effective area with energy resolution for each bin in reconstructed energy, zenith angle and off-axis angle and can therefore be very expensive).

It's possible to fit all spectrum parameters, or to **fix some of them**.

Equivalent Chi Square

The log-likelihood is normalised to 0 if the number of events is equal to the expected number of events in each bin.

Flux Uncertainties

The flux uncertainty at each energy is determined using the full covariance matrix of the spectrum paramaters.

Assuming:

- $W_{i,j}$ the error matrix (inverse of covariance matrix),
- $\Phi(E)$ the differential flux at energy E,
- α_i the spectrum parameters,

the flux uncertainty is determined by the equation

$$d\phi = \sqrt{\sum_{i,j} W_{i,j} \frac{\partial \phi}{\partial \alpha_i} \frac{\partial \phi}{\partial \alpha_j}}$$

Reference energy

The *Reference Energy* is the energy where the differential flux is calculated and quoted. It's also the energy E_0 used in the **spectrum shapes formula**. For instance, the spectrum index parameter in the **Curved power law spectrum** depends on the reference energy:

$$\frac{dN}{dE} = \Phi_0 \left(\frac{E}{E_0} \right)^{\alpha - \beta \log(E/E_0)}$$

By definition, in this case α is the spectrum index **at the reference energy** and a change from E_0 to E_1 as reference energy induces a change of parameter α :

$$\alpha_1 = \alpha_0 - \beta\beta \log(E_1/E_0)$$

The reference energy can be changed using the function **set_reference_energy**. It's safer to do it prior to spectrum fit, but one can also do it interactively after the fit is done via a call to **Spectrum::SpectrumBase::SetReferenceEnergy**. The spectrum parameters will be recomputed and be used at the next **Spectrum::SpectrumBase::Display** call.

Decorrelation energy

The *Decorrelation Energy* is the energy where the relative flux uncertainty ($d\phi/\phi$) is the smallest. At this energy, the spectrum parameters are usually mostly uncorrelated (*i.e.*, the error matrix is almost diagonal). As a

consequence, this is the energy where the uncertainty on the spectrum parameters is the smallest.

When performing a fit, we change the reference energy to the decorrelation energy to reduce the uncertainty on the spectrum parameters, unless the function **set_reference_energy** was called with the second argument set to *true*.

Spectrum Points

Methodology

Since the events are binned in **reconstructed energy** and not in **true energy**, the translation of spectrum parameters into points is not trivial.

For each reconstructed energy bin $[\tilde{E}_1, \tilde{E}_2]$, the average true energy is given by:

$$\langle E \rangle = \frac{\int_{\tilde{E}_1}^{\tilde{E}_2} E \times d\tilde{E} \int_0^\infty dE \times \Phi(E) \times \mathcal{A}(E,\theta,\delta) \times PDF(E,\tilde{E})}{\int_{\tilde{E}_1}^{\tilde{E}_2} d\tilde{E} \int_0^\infty dE \times \Phi(E) \times \mathcal{A}(E,\theta,\delta) \times PDF(E,\tilde{E})}$$

Similarily, the true corresponding true energy variance $\sigma_E = \sqrt{\langle E^2 \rangle - \langle E \rangle^2}$ can be easily calculated using the average squared energy:

$$\langle E^2 \rangle = \frac{\int_{\tilde{E}_1}^{\tilde{E}_2} E^2 \times d\tilde{E} \int_0^\infty dE \times \Phi(E) \times \mathcal{A}(E,\theta,\delta) \times PDF(E,\tilde{E})}{\int_{\tilde{E}_1}^{\tilde{E}_2} d\tilde{E} \int_0^\infty dE \times \Phi(E) \times \mathcal{A}(E,\theta,\delta) \times PDF(E,\tilde{E})}$$

This will give the **position and horizontal error bars of the spectrum points**.

The number of excess events is given by $N \pm dN$ whereas the expected number of gamma events is

$$n_\gamma(\tilde{E}_1, \tilde{E}_2) = \int_{\tilde{E}_1}^{\tilde{E}_2} d\tilde{E} \int_0^\infty dE \times \Phi(E) \times \mathcal{A}(E,\theta,\delta) \times PDF(E,\tilde{E})$$

The reconstructed flux $\tilde{\Phi}$ in the bin $[\tilde{E}_1, \tilde{E}_2]$ is therefore calculated by a scaling of the adjusted flux Φ with the ratio of observed excess events to expected one:

$$\tilde{\Phi}(\langle E \rangle) = \frac{N}{n_\gamma} \times \Phi(\langle E \rangle)$$

Note:

The uncertainties on the bin flux are calculated in a similar way. It has to be noted that the uncertainties on the spectrum parameters are not taken into account (i.e. *propagated*) into the spectrum points uncertainties, to keep the points as much uncorrelated as possible.
The points are almost uncorrelated, since they are calculated on disjoint groups of events.

Binning definition

Todo:

To be documented: **Binning definition**

Change binning:

```
void set_sigma_perpoint(float Nsigma = 3);
```

Lightcurve

See **Lightcurve determination**

Confidence Region

Todo:

Confidence Region not documented yet

To ask for confidence region determination:

```
void make_spectrum_confidenceregion(bool);
```

It is also possible to compute the confidence region for any pair of parameters (this uses the class **Spectrum::ConfidenceRegionMaker**), after having done the spectrum fit. The script function **ComputeConfidenceRegion** takes two arguments: the two spectrum parameters for which one wants to compute the confidence region. For example, in the case of a **Spectrum::SpectrumCurvedPowerLaw** "Curved Power Law" we have the following parameters:

- Flux normalisation
- **Spectrum** index at **reference energy**.
- Curvature

For producing the confidence region for curvature versus spectral index, one would type:

```
//  Fit spectrum
FitSpectrum("Crab_4_05_Combined_60pe");

// Compute Confidence Region
ComputeConfidenceRegion(1,2);
```

Note:

Since this is a quite slow procedure, it is often a good idea to increase the integration precision before computing the confidence region:

```
// Decrease integration precision to go faster
set_integration_precision(1e-2);
```

Cosmic Ray Spectrum

The OFF regions can be used to determine the spectrum of cosmic rays in cuts. This uses gamma-like acceptances and resolutions.

The algorithm is exactly the same that the usual spectrum determination, except than we we replace the ON data by the OFF data and set the OFF to zero:

$$P(N_{OFF}|n_{CR}) = \frac{n_C R^{N_{OFF}}}{N_{OFF}!} e^{-n_C R}$$

See example **make_crab_cr_spectrum::C**