



Butterfly Status Report

2009 Oct 01

Fabricio Ferrari

The purpose of this document is to describe the current status of the Butterfly software system. At the moment, Butterfly is composed of the following independent systems

- Illusion – BTFI data synthesizer (Bruno)
- ITC – Integration time Calculator (Sergio)
- Gollum – Ring fitting algorithm (Fabricio)
- BatFly – Phase finding and correcting (Fabricio)

Here we will focus on **Gollum** and **BatFly**. Illusion and ITC will have its own reports.

All routines are coded in a modular and object oriented fashion, making it easy to integrate with any other Python system, for example that which will communicate with the rest of the system. They are command line driven and no runtime user interaction is needed.

Gollum (Feb/2009)

Gollum is designed to fit an analytical 2D function to an image with a ring or part of a ring, such as those produced by a Fabry-Perot instrument. The fitting function¹ is not the Airy profile, but the output fitted parameters (except σ) would be the same. Eventually, the function can be the Airy function with little work, with a penalty in the algorithm speed.

The input parameters might be the ring peak intensity I , the ring center (x_0, y_0) , the ring radius R and the ring width σ . These input serve as first guesses for the fitting algorithm. The outputs are the same of the inputs

$$I \exp \left\{ - \frac{[(x-x_0)^2 + (y-y_0)^2 - R^2]^2}{2\sigma^4} \right\}$$

plus a residual image builded by subtracting the 2D function from original image. From this residual image the fit quality are estimated, by measuring the mean and the standard deviation of the residual, the last is the χ^2 of the fit.

Gollum was stopped at this stage for it was not considered an urgent task at the moment.

BatFly v1.2 (Aug/2009)

BatFly is the phase correction package. It routines currently include:

1. ReadFits – reads and save important information about the data cube, such as dimensions, central spectra, to be used by the processing routines.
2. PeakFFind – routine that finds the peaks in the spectra by smoothing the 1st derivative and verifying where it changes signal (a peak).
3. PhaseFind – does the actual Gaussian fitting and saves the results on maps (Intensity, center, sigma, phase, FSR). It takes the peaks parameters found by PeakFFind as input for the fit.
4. PhaseCorrect – takes the phase map measured by PhaseFind() and then calls ShiftSpectra to do the displacement in each spectra in order to have the cube phase corrected.
5. ShiftSpectra – it begins by resampling the spectra with a number of points a 100 times that of the original spectra. Then it cuts the spectra at 1 FSR (measured by PhaseFind); then, it cuts a portion of the spectra at its end with size equal to the shift to be done; this slice is put back in the beginning of the spectra. With this sequence of operations, the spectra is shifted by the given amount. If the shifted amount is negative the piece is taken from the beginning and put at the end. Finally, it downsamples the spectra to the original sampling and returns it.
6. SynPhaseMap – takes the measured phase map and fits a paraboloid folded by a remainder operation (*mod()*) in order to represent its folding. Then builds an unfolded synthetic phase map from that function.

So, in a typical situation, one would call the PeakFind(), PhaseFind() and then SynPhaseMap on a calibration file in order to build the phase

map for the instrument. Then, subsequently will apply PhaseCorrect() on the science cube with the given synthetic phase map. If the FSR of the science cube is different from that of the calibration cube, then it must be supplied.