

# PySpi for Persistent Sources

Möller, Julius

May 2023

## Contents

<b>1</b>	<b>General Procedure</b>	<b>2</b>
<b>2</b>	<b>Spimodfit Procedure</b>	<b>3</b>
<b>3</b>	<b>Results: Crab Nebula</b>	<b>4</b>
<b>4</b>	<b>Results: Simulated Source</b>	<b>4</b>

# 1 General Procedure

In contrast to previous approaches to working with INTEGRAL SPI, this method does not attempt to fit the background via physical models, and instead uses a profile likelihood to eliminate its effect in the source model fit. The underlying assumption that allows this to work is that background count rates in each energy bin remain constant for temporally and spatially near SPI Science Windows (SCWs). Here we distinguish between source counts, which describe all detected events directly originating from astronomical objects considered in our source model and background counts, which describe all other detector events.

This works in the following way: once we have split our set of SCWs into clusters within which we claim the assumption of constant background to be true and have defined a source model, a likelihood value can be calculated for each energy bin of each detector in each cluster. If one energy bin of one detector in one SCW with measuring time  $t$  has a source count rate  $s$  and a background count rate  $b$ , the probability  $P$  of measuring  $C$  counts follows a Poisson distribution:

$$P(C|b, s, t) = \frac{(t(b + s))^C \exp(-t(b + s))}{C!} \quad (1)$$

Equivalent energy bins in the same cluster are assumed to have the same background count rate  $b$ , so that the probability of measuring  $C_1$  counts in one SCW and  $C_2$  counts in another SCW is simply:

$$P(C_1, C_2|b, s_1, t_1, s_2, t_2) = P(C_1|b, s_1, t_1) \cdot P(C_2|b, s_2, t_2) \quad (2)$$

The likelihood  $\mathcal{L}$  for any energy bin of a detector within a cluster depends on all equivalent energy bins in that cluster, i.e. one term for each SCW in the cluster. If we have cluster of size two, the likelihood is described by:

$$\ln \mathcal{L}(C_1, C_2, t_1, t_2|b, s_1, s_2) = \ln P(C_1|b, s_1, t_1) + \ln P(C_2|b, s_2, t_2) \quad (3)$$

This expression allows us to solve for the probability distribution of  $b$ . A full and mathematically correct treatment of the background parameter  $b$  turns out to be very cumbersome with only marginal benefits. Hence, the profile likelihood is used by simply solving for the background count rate value  $b_M$  which maximizes the likelihood. This is easily done by setting the derivative of equation 3 to zero and solving for  $b$ . For a cluster size of two one finds:

$$b_M = \frac{1}{2} \left[ \frac{C_t}{t_t} - s_t + \sqrt{\left( \frac{C_t}{t_t} - s_t \right)^2 + 4 \left( \frac{C_1 s_2 + C_2 s_1}{t_t} - s_1 s_2 \right)} \right] \quad (4)$$

where  $C_t = C_1 + C_2$ ,  $s_t = s_1 + s_2$ , and  $t_t = t_1 + t_2$ . Using this maximum likelihood background  $b_M$  finally allows us to compute a likelihood value that is only dependent on the measured counts and the source model:

$$\ln \mathcal{L}(C_1, C_2, b_M, t_1, t_2|s_1, s_2) = \ln P(C_1|b_M, s_1, t_1) + \ln P(C_2|b_M, s_2, t_2) \quad (5)$$

The process for clusters with sizes larger than two is analogous. One such likelihood value exists for every energy bin of every detector in every cluster, such that the total likelihood value for any given source model is simply the sum of the logarithmic likelihoods over all energy bins, detectors, and clusters. Thus the source model may be fitted to the count data by applying an algorithm to maximize the likelihood.

## 2 Spimodfit Procedure

In order to be able to compare the results obtain using PySpi and the method described above, spimodfit will be used with the same dataset. For the sake of limiting sources of potential errors, the procedure for obtaining results using spimodfit will be described in more detail than might otherwise be deemed necessary. In general, the spimodfit cookbook found [here](#) was followed very closely.

- 1.

### 3 Results: Crab Nebula

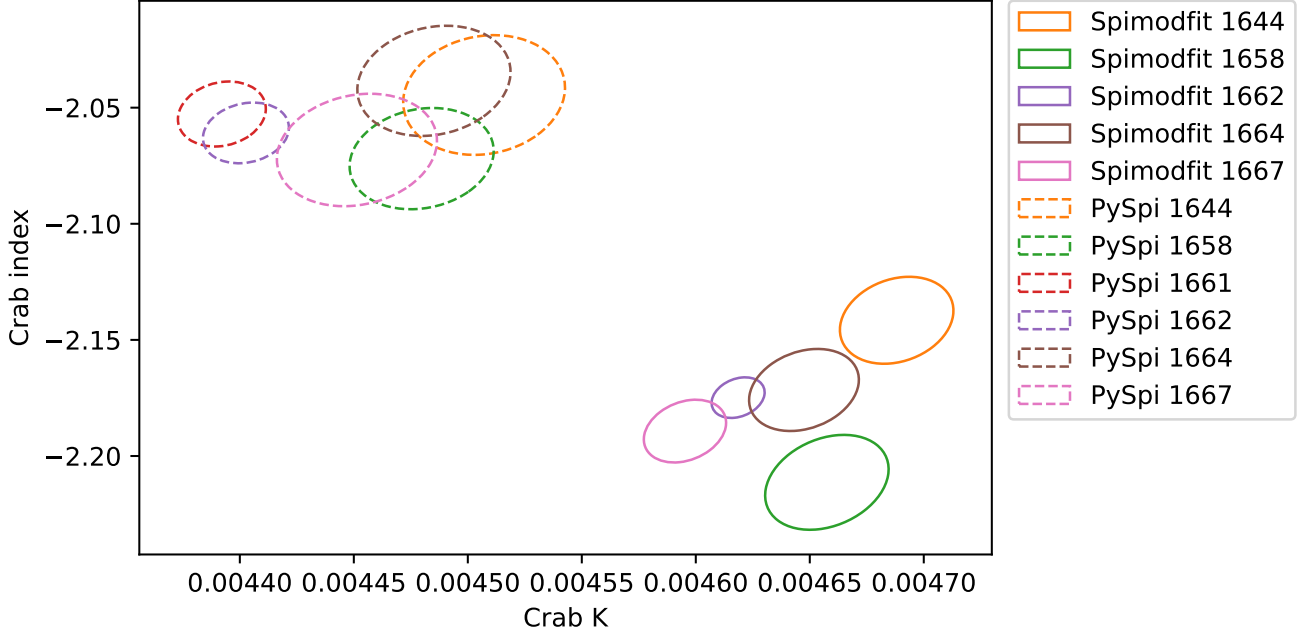


Figure 1

Figure 1 shows one standard deviation confidence intervals of several near SPI Revolutions, as fitted using PySpi and Spimodfit. Revolution 1661 was only fitted using PySpi since spimodfit rejected all SCWs due to "General Expression". The source model is a simple powerlaw

$$f(x) = K \frac{x^{index}}{piv} \quad (6)$$

where  $K$ , measured in  $\text{keV}^{-1}\text{s}^{-1}\text{cm}^{-2}$  is the differential flux at the pivot value  $piv = 40\text{keV}$ ,  $x$  is the photon energy in keV, and  $index$  is the index of the powerlaw.

Both methods used the same energy range of 20-81.5keV, although it is worth pointing out that they use independent algorithms to select which SCWs are used in the fit. PySpi used a cluster size of two.

Although the two methods obtain different values for the index and normalization, we see comparable sizes of uncertainties as well as similar levels of compactness within the values from different revolutions.

### 4 Results: Simulated Source

The downside to fitting a real astronomical object such as the Crab nebula is that one can never be exactly sure what the true source parameter values should be. This is why it can be advantageous to simulate a source, feed the simulated data back into the fit, and compare the produced source parameters to the now known true values.

However, not only a simulated source is required, but a background as well. There are different approaches one may take to realize this. The first approach is to use an entirely real dataset from a SPI revolution as background, and add a simulated source on top:

1. Choose real data set to use as background. In this case single SPI revolutions were used. For this to work well, there should be as few bright sources within the field of view as possible, since these will interfere with the fit if not included in the source model.
2. Define spectrum of the source to be simulated, as well as position. In this case powerlaws as shown in equation 6 are used, and the position of the sources is chosen to be central to as many SCWs in the revolution as possible.
3. Use the SPI Instrument Response Function (IRF) to calculate the expected source count rates in each energy bin in each detector for each SCW. Multiply by these be the respective lifetime (effectively the active measuring time) of the detector in the SCW, and finally draw a sample of a Poisson distribution with the

respective mean to attain the measured counts of the simulated source for each energy bin in each detector in each SCW.

4. These source count rates are then added to the real measured data we are using as background.

This dataset can then be used to fit the simulated source with both PySpi and Spimodfit. Although this method of generating the background counts produces a very realistic background, in some cases it may be too realistic by including unwanted sources in the field of view that interfere with the fit. Unlike PySpi, Spimodfit does use a background model, which can also be used in combination with the simulated source:

1. When