Count Rate Modeling Strategy

In the **unpixelized** form of *Forward Fitting* we will compute a model count rate profile directly from the parameters of the basic Gaussian sources. To do this, we need to get the following quantities of the *calibrated event list* for:

- Φ = phase at map center
- τ = gridtran
- T = livetime
- θ = roll angle
- A = modamp

Each one of these is a vector of length equal to the number of timebins used for the map. We run the simulation of the *calibrated event list* exactly **once** for a point source at the center of the map, throw away extraneous tags (such as the counts), and save the above vectors for subsequent iterations.

For each of the Gaussian sources, we have 6 free parameters:

- a_0 = Flux
- a_1 = Long axis width
- a_2 = short axis width
- a_3 = Tilt angle
- a_4 = X offset
- a_5 = Y offset

Then we use these and the *calibrated-eventlist* parameters together to compute the model count rate profile for each Gaussian and each subcollimator. The count rates can be written as a function of the *phase at map center* Φ:

$$f_{mod} = a_0 \tau (1 + c_1 \ cos\Phi_m \quad + \text{higher harmonics}) \tag{1}$$

The phase term Φ_m is the sum of the *phase at map center* (Φ) and the phase offset of a point source at a_4, a_5 relative to map center:

$$\Phi_m = \Phi + 2\pi \ (a_4 \ cos\theta \ + \ a_5 \ sin\theta)/p, \tag{2}$$

where p is the angular pitch of the particular subcollimator. In practice, Φ need be computed only once, while the offset terms change during the iteration and must be recomputed for each trial source position. The fundamental amplitude term c_1 for the Gaussian source is computed from *modamp* (A) and a_1, a_2 and a_3 using:

$$c_1 = A \cdot e^{-\pi/p)^2\{cos^2(\theta-a_3)a_1^2+sin^2(\theta-a_3)a_2^2\}}, \tag{3}$$

At each step of the iterative search for a model which matches the data, we compute the model count rates f_{mod} for each Gaussian using equation (1), and add them up to produce the total count rate. This is done for each subcollimator separately.

Questions

In applying this to a real program, several questions arise:

- How do we keep f_{mod} positive at large radii? (3 harmonics?)
- Is there a way to use c_1 alone and not f_{mod}?
- How do we handle low count rates?
  - Model-Ordered Statistics?
  - Cash Statistic?
  - Regularization?
- What do we do about extraneous local minima? (Simulated annealing?)