DESCRIPTION OF THE EXPOSURE MAP GENERATOR, `AG_expmapgen`

ANDREW CHEN, TOMASO CONTESSI

1. Change History

2009-07-22, first issue of this document.

2. Introduction

`AG_expmapgen` is the program that, given an index of log files, a time interval, an energy range, and map projection parameters, generates an exposure map readable by FITS image display routines and usable for AGILE analysis software. *Exposure* is the cumulative product of effective area, time, and solid angle of the instrument (AGILE GRID) in each map pixel within the given time period and energy range.

3. Effective area

The effective area of the instrument depends not only on its geometric area but also on the efficiency with which photons of a given energy are detected and recognized by the standard analysis as photons. This depends on the filter used as well as the direction from which the photon enters the instrument. The effective area for gamma-ray sources assumed to have a power-law spectrum with a given index in a given energy range is calculated as a weighted average of the effective areas in each energy channel. Two methods have been developed, one which takes into account the energy dispersion (EDP) and one which does not.

The software currently released, version BUILD19, is not taking into account the energy dispersion.

3.1. Without EDP.

In an energy interval between E_{min} and E_{max}, assuming a spectrum of $E^{-\alpha}$, the area A_{EFF} will have the form:

$$A_{EFF} = \frac{\sum_{i=i_{min}}^{i_{max}} A_i W(E_i)}{\sum_{i=i_{min}}^{i_{max}} W(E_i)}$$

where $W(E_i) = E_i^{1-\alpha} - E_{i+1}^{1-\alpha}$
or $W(E_{i_{max}}) = E_{i_{max}}^{1-\alpha}$ if i_{max} is the last available index.

3.2. **With EDP.** Taking into account the energy dispersion,

$$A_{EFF} = \frac{\sum_{j=0}^{j_\infty} \sum_{i=i_{min}}^{i_{max}} A_j W(E_j) EDP(E_j, E_i)}{\sum_{i=i_{min}}^{i_{max}} W(E_i)}$$

where $W(E_i)$ is defined as above and $EDP(E_{true}, E_{obs})$ is the fraction of photons in energy channel E_{true} with reconstructed energy in channel E_{obs}.

3.3. **Determination of i_{min} and i_{max}.** We impose the requirement that the minimum energy of the user-specified energy interval must coincide with one of the pre-defined energy channel boundaries. The maximum energy must either also coincide with one of the pre-defined energy channel boundaries or be greater than the highest energy channel boundary, in which case the last energy channel, which has upper bound of ∞, will be used. i_{min} will be the index of the channel whose lower bound is E_{min}, while i_{max} will either be the channel whose upper bound is E_{max} or the last energy channel if E_{max} is higher than the highest energy channel boundary.

If the user specifies energies which fail to coincide with any of the energy channel boundaries, the program will issue a warning and attempt to find the nearest energy channel boundary using the following algorithm: If the energy channels have boundaries $\{E_i\}$, and the energy interval to be calculated is $E_1..E_2$, the indexes i_{min} and i_{max} are such that E_{min} and E_{max} are as close as possible to E_1 and E_2 in the geometrical sense. In other words, if $E_i < E$, the distance between them is $\frac{E_i}{E}$. If E_{max} is greater than the highest energy channel boundary, then i_{max} is set to the last energy channel as mentioned above. Note that the program will insure that at least one energy channel, corresponding to i_{min}, is used, even if $i_{max} < i_{min}$.

4. Time

The program selects the rows from the log files listed in the log index files according to the following criteria:

(1) The time falls within the time interval selected by the user, from `tmin` to `tmax`;
(2) $LIVETIME > 0$, $LOG\_STATUS == 0$, and $MODE == 2$;
(3) The orbital phase is among those allowed by `phasecode`.

The program determines the optimum block increment using `fits_get_rowsize()`. Within each block increment, the program determines the rows for which any of the following changes occur:

- The orbital phase changes;
- The pointing direction changes by an amount greater than `y_tol`;
- The rotation of the instrument about the pointing direction changes by an amount greater than `roll_tol`;
- The direction of the earth relative to the instrument changes by an amount greater than `earth_tol`;

For each period within which none of these changes occurs the effective area for a each bin is calculated (unless `step` > 1; see below) according to its direction with respect to the instrument axes. This area is multiplied by the number of rows within the period, the livetime of each row, and the area of the pixel.

5. Solid angle

The solid angle of each bin is calculated according to the projection (ARC or AIT) used and multiplied into the exposure.

6. Interpolation

In order to speed up the calculation, `AG_expmapgen` calculates the exposure only for every nth bin of each row and column, where n is given by the `step` input parameter, and estimates the intervening values by interpolation.

7. FITS image

Once all the bins of the exposure map are filled, the result is written to a FITS file with the name provided by the user.