MIT Kavli Institute **Chandra X-Ray Center**

MEMORANDUM
November 4, 2015

To:	File
From:	David P. Huenemoerder, Glenn E. Allen
Subject:	Description of enhancements to HETG/ACIS CC-mode processing
Revision:	1.0
URL:	`http://space.mit.edu/cxc/docs/docs.html#hetgcc`
File:	`~dph/CXC/Grating_CC_mode/Doc/Memo_tgre/grating_cc_2015-01.tex`

1 Overview

We have implemented changes in `acis_process_events` and `tg_resolve_events` which improve the order-sorting of HETG data taken in CC-mode. Order-sorting requires accurate scaling of CCD pulse-heights to energy (`CTI`/gain corrections). However, in CC-mode, the event coordinates in the parallel read direction, `CHIPY`, are not known. Prior to now (CIAO 4.8, or DS 10.4.2.1), corrections were done for either the projected source location on the CCD (zeroth order position), or for the center of the array if the zeroth order was off the array. This could result in CCD `ENERGY` values which are too high or too low, depending on the events' grating arm (HEG or MEG) and position along the spectrum (determined by diffraction order and wavelength). Order sorting is ideally done through a lookup in the Calibration Database (`CALDB`) of CCD-response-derived `ENERGY` values as a function of `CCD_ID`, `CHIPX`, `CHIPY`, and `ENERGY` (the "`OSIP`" file). If the energy-dependent corrections are in error, then diffracted photon events can fall outside these boundaries and be erroneously rejected. To mitigate this, we have often used the `tg_resolve_events` option of using a "flat" (in wavelength vs. order coordinates), wide order-sorting region (`osipfile="none", osort_lo=0.3, osort_hi=0.3`). This is forgiving of `CTI`/gain correction errors, but has the side effect of allowing much more background, particularly at the shortest (accepting zeroth order scattering halos) and longest wavelengths (were source signal drops due to decreasing effective area or interstellar absorption).

Since `tg_resolve_events` estimates a `CHIPY` value in order to do the `OSIP` lookup, we have implemented the ability to do a second pass through `acis_process_events` which uses this value to update the ENERGY values, and then a second pass through tg_resolve_events will re-apply order-sorting. These changes have required some new parameter settings and header keywords. Proper use also requires a specific procedure to execute the two passes properly.

The enhancements also inlude corrections to the event arrival time which depend on the CHIPY position. In principle, light curves derived from HEG or MEG events will be more accurate.

We document the changes, procedures, and test cases in detail below.

2 Summary of Recommended Procedure

acis_process_events **first pass** as usual (with appropriate CC-mode "eventdef" parameter).

tgdetect **(or** tgdetect2**),** tg_create_mask**:** grating souce detection and mask creation, as usual.[1]

tg_resolve_events **first pass** with a flat order sorting (e.g, osipfile=NONE, osort_lo=0.3, osort_hi=0.3) to tolerate large errors in CTI/gain/tgain corrections. We also use an eventdef to write the columns required later by acis_process_events: "eventdef=)cclev1a" or "eventdef=)ccgrdlev1a".

acis_process_events **second pass** will detect that tg_resolve_events has provided CHIPY_TG and will update CHIPY-dependent dependencies.

tg_resolve_events **second pass** with response-based order sorting (osipfile=CALDB and appropriate eventdef.

tgextract spectrum extraction, as usual

make responses as usual, but being careful to specify osipfile=CALDB in mkgarf for proper application of the order-sorting region's enclosed energy fraction.

2.1 Caveats

Higher orders seem to be generally overcorrected, but still fall within the order-sorting region. Some cases, however, clip data on the first pass and may need a wider region (e.g., Sco X-1). While the procedure and parameters above work in general, care and inspection of the details are prudent.

[1] For the few cases in which zeroth order is off the array, acis_process_events will now put the 1-D trace through the source's celestial coordinates, not the projected array center. This means that standard processing should be sufficient, rather than manually determined source sky x, y positions, and masks wide enough to reach the events.

3 First pass of acis_process_events

The first pass of acis_process_events has no operational changes, but there have been changes in the output sky 1D coordinates: they will now pass through the target's celestial coordinates (even if the zeroth order is off the array). For CHIPY-dependent corrections, the CHIPY-value corresponding to the target position (RA_TARG, DEC_TARG) is used, if the target is on the array. Named eventdefs are used as appropriate for faint or graded modes (cclev1 or ccgrdlev1).

4 First pass of tg_resolve_events

On the first pass of tg_resolve_events, there are no changes to the algorithm. However, we need to write columns to the output event file which are required for the second pass of acis_process_events. This is specified by the "eventdef" parameter which gives the FITS event file columns to write along with their datatype. Several options are typically in the program's parameter file and eventdef redirects to one of them. Two sets are relevant for CC-modes, one for CC-faint, and the other for CC-graded. These will be included in both acis_process_events.par and tg_resolve_events.par as follows:

```
cclev1a,s,h,"{d:time,d:time_ro,l:expno,s:ccd_id,s:node_id,s:chip,\
          f:chipy_tg,f:chipy_zo,s:tdet,f:det,f:sky,f:sky_1d,s:phas,\
          l:pha,l:pha_ro,f:energy,l:pi,s:fltgrade,s:grade,f:rd,\
          s:tg_m,f:tg_lam,f:tg_mlam,s:tg_srcid,s:tg_part,s:tg_smap,\
          x:status}"

 ccgrdlev1a,s,h,"{d:time,d:time_ro,l:expno,s:ccd_id,s:node_id,s:chip,\
          f:chipy_tg,f:chipy_zo,s:tdet,f:det,f:sky,f:sky_1d,l:pha,\
          l:pha_ro,s:corn_pha,f:energy,l:pi,s:fltgrade,s:grade,f:rd,\
          s:tg_m,f:tg_lam,f:tg_mlam,s:tg_srcid,s:tg_part,s:tg_smap,\
          x:status}"
```

(which differ only in columns phas and corn_pha).

In the parameter file, a specific event definition is typically invoked by a redirection to an alias within the parameter file; for example:

```
eventdef,s,h,")cclev1a",,,"output format definition"
```

On the first pass of tg_resolve_events for CC-mode data, it is recommended to use the "flat" order-sorting option (instead of the default CALDB file) by use of these parameters and values:

```
osipfile,f,h,"none",,,"Lookup table for order resolving (for acis data only)"
osort_lo,r,h,0.3,0,0.5,"Order-sorting lower bound fraction; order>m-osort_lo"
osort_hi,r,h,0.3,0,0.5,"Order-sorting high bound fraction; order<=m+osort_hi"
```

5 Second Pass of acis_process_events

The second pass of acis_process_events will automatically detect by the CONTENT keyword value of TGEVT1 that grating processing has been applied. acis_process_events will use the CHIPY_TG column to update dependent quantities, such as ENERGY and times.

A header keyword will indicate that this processing has been done:

TIME_ADJ = GRATING / time adjustment algorithm[2]

One of two new eventdef parameters should be specified (one for faint and one for graded modes) to output grating data columns, as required downstream:

eventdef =)cclev1a or

eventdef =)ccgrdlev1a

6 Second Pass of tg_resolve_events

The second pass of tg_resolve_events will detect that the first passes have been applied (by the presence of grating-related columns, and the TIME_ADJ keyword value), and only order-sorting will be updated. Order-sorting will be repeated using the improved ENERGY values. Here we should generally use the osipfile=CALDB setting, since this has a calibrated enclosed energy fraction, and the order-sorting region excludes much of the background.

Columns which may change under this procedure are TG_M, TG_LAM, TG_PART, and CHIPY_TG.

During this update to orders, it is possible for the grating types (HEG vs MEG) to change, if an event is excluded from one grating's orders, but is accepted by the other. This will also change CHIPY_TG, which introduces an inconsistency. In principle, one could iterate (a 3rd pass), but tests have shown that the number of such swaps is at a level of fractions of a percent.

7 Testing & Examples

Tests were done on a set of HETGS/CC-mode observations at detector offsets which placed the zeroth order off the array, near the bottom (low CHIPY), centered, or near the top (high CHIPY), in both faint and graded modes. Data were processed both with and without the osipfile (acisD1999-09-16osipN0007.fits), and for osort_lo,hi values of 0.3 and 0.4. Data were processed with CIAO 4.7 and with CIAOX 4.8. For processing efficiency, files were truncated to an exposure giving $\leq 400,000$ events. One LETG/ACIS case was included (ObsID 12444) to verify that this mode did not change (since the CHIPY location is known from the target location, given the spectrum trace is nearly constant in CHIPY).

[2]Other allowed values are NONE (TE-mode), MIDCHIP (CC-mode secondary observation, which has no aspect file), or TARGET (CC-mode Level 1 with the source on the detector).

Table 1: Test data information

ObsID	SIM	⟨CHIPY⟩	Datamode	Object
680	10.0	917.70	CC33_GRADED	XTE J1550-564
3505	-15.0	-125.99	CC33_GRADED	SCO X-1
5888	-11.3	23.26	CC33_GRADED	GX 5-1
6297	-4.0	330.83	CC33_FAINT	SAX J1808.4-3658
6628	-7.5	183.98	CC33_GRADED	GX 349+2
7268	-6.8	208.73	CC33_GRADED	Cygnus X-3
10660	-6.8	230.59	CC33_FAINT	4U 1957+11
10691	-11.3	45.16	CC33_GRADED	GX 5-1
12314	-12.3	5.80	CC33_FAINT	Cyg X-1
12444	-8.0	177.77	CC33_FAINT	4U 1820-30
13222	-11.3	47.08	CC33_FAINT	GX 349+2
17697	-12.6	-8.92	CC33_GRADED	v404 Cyg

The SIM column gives a relative SIM_Z offset in mm, where -12 puts the zeroth order near the bottom, -7 is near the center, and $+10$ is near the top. The CHIPY column gives the nominal CHIPY position of the target, as a median over the dither.

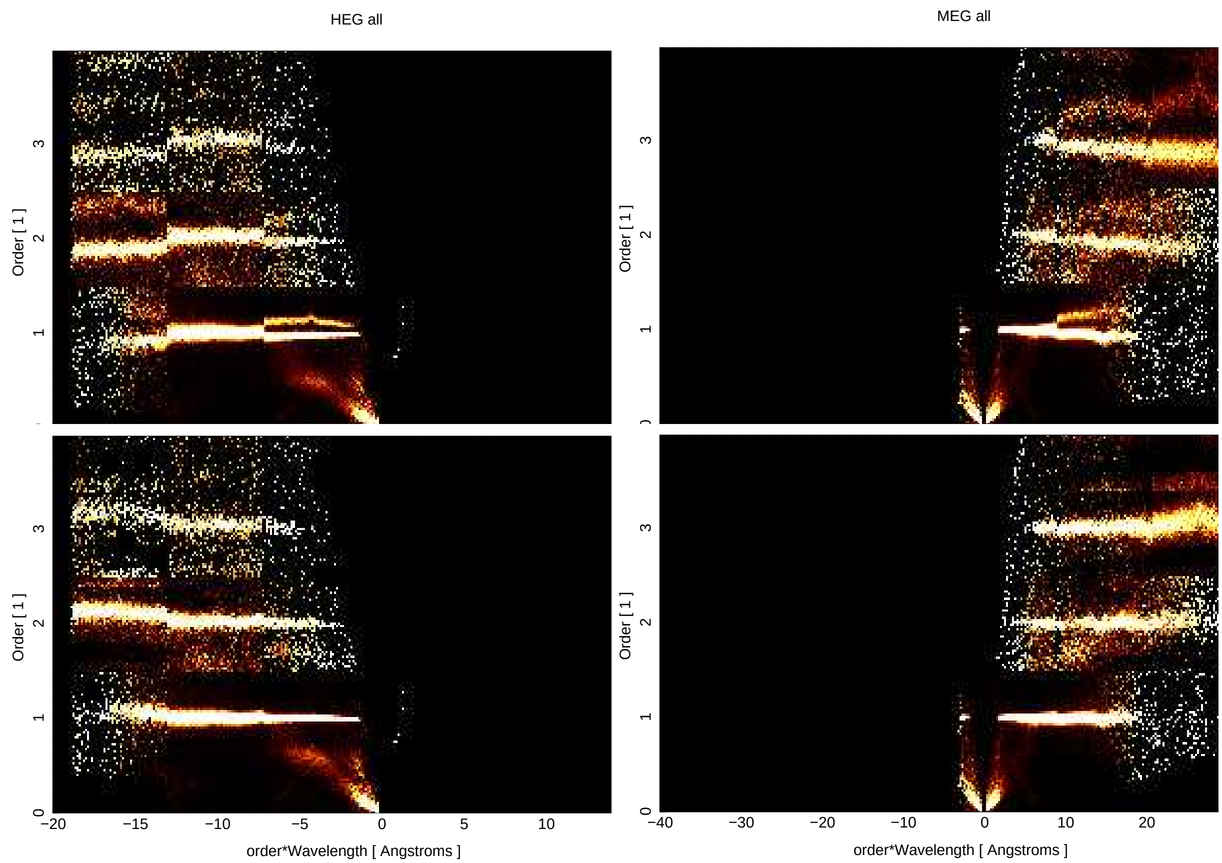

Figure 1: Example of large CTI/gain correction effects in ObsID 12314 (Cyg X-1), in which the zeroth order dithered on-and-off the detector. These are "order-sorting" images, in which real-valued orders have been computed for each event (y-axis); intensity has been normalized in each order at each wavelength. The top panels show Pass 1 output, and the bottom Pass 2. On the left is HEG, and right MEG. The orders have *not been filtered* by either flat or "OSIP" regions so that effects can be seen before filtering. The splitting seen in the top panels Pass 1 orders is due to some events having center-of-chip CTI/gain corrections applied while the zeroth order was off the array, and zeroth order CHIPY coordinates applied otherwise. In the bottom panels, after Pass 2, we see that the this splitting has been corrected, and that first orders are more horizontal. Higher orders appear a bit over-corrected.

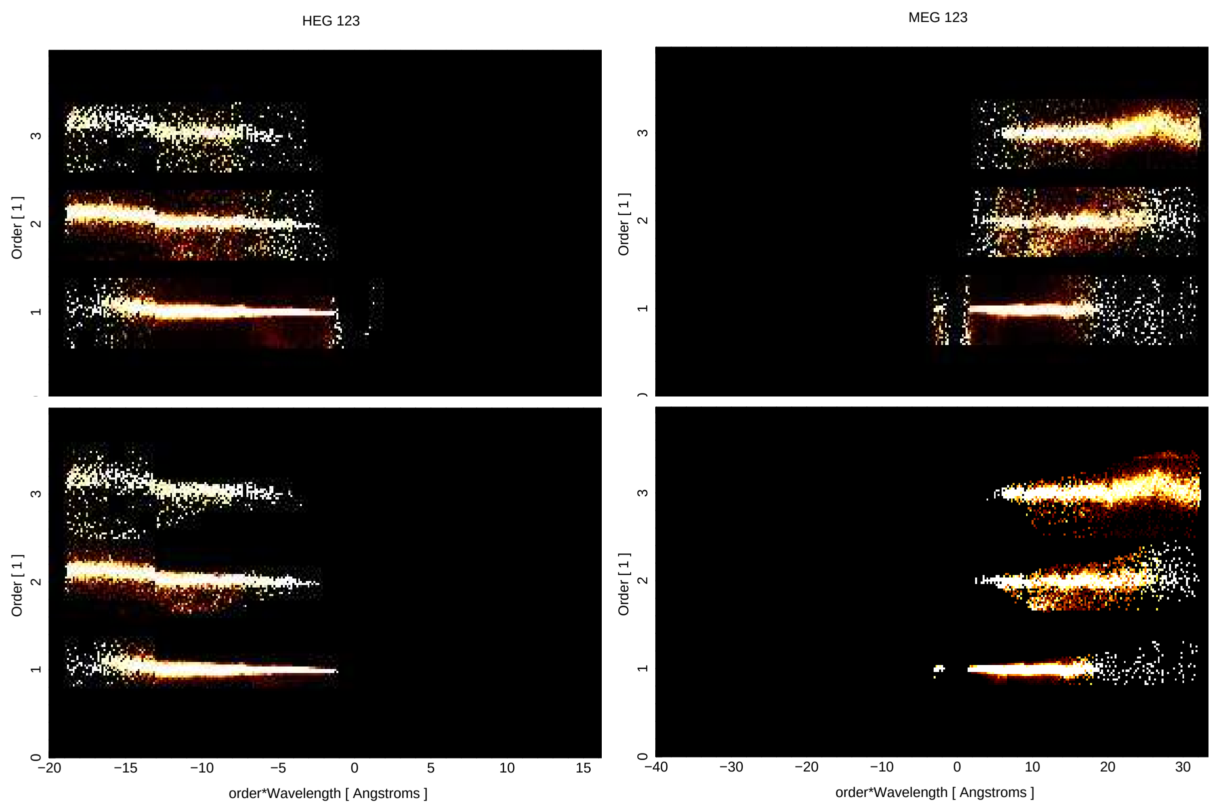

Figure 2: This is similar to Figure 1, but here clipping regions are applied, either a "flat" order-sorting (top), or the `CALDB` "`OSIP`" (bottom). In the bottom plot, you can see how the background is reduced near $\lambda = 0$, and at longer wavelengths, but that the source spectrum is unaffected.

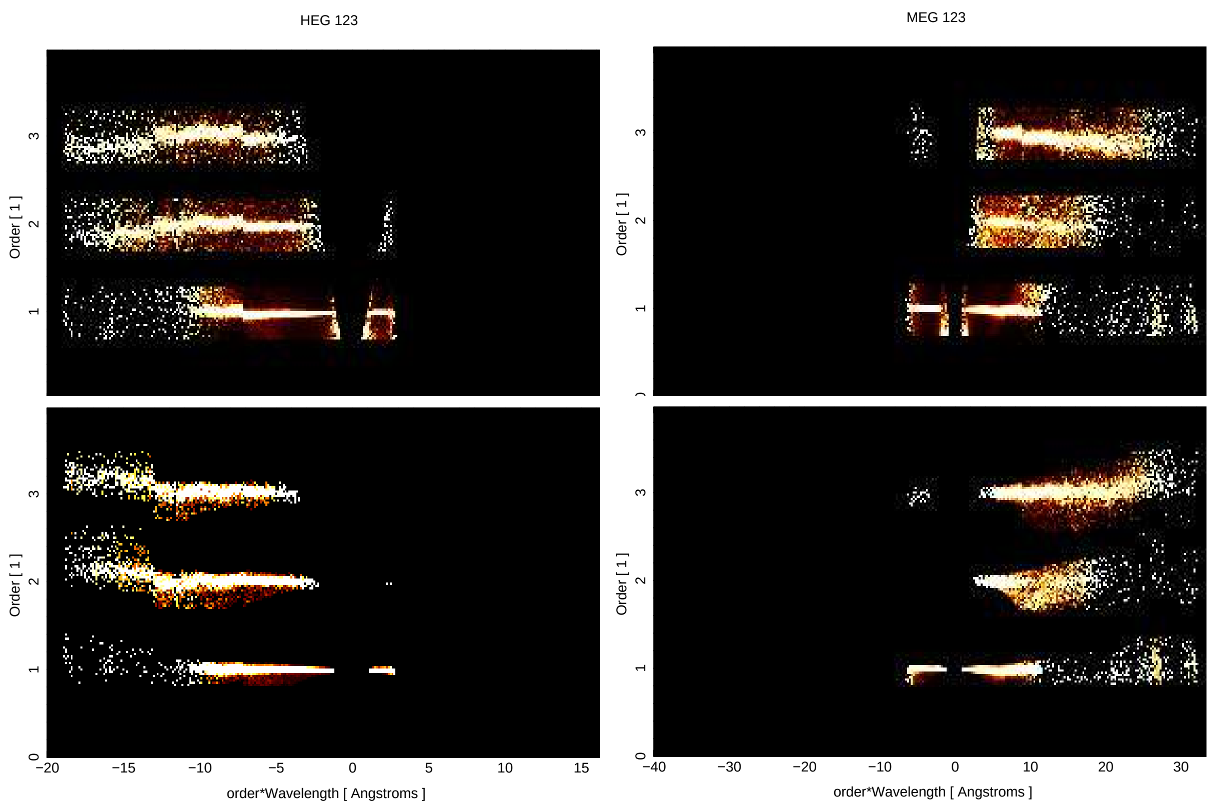

Figure 3: Another example, ObsID 5888 (GX 5-1, zeroth order near the bottom of the array) showing mild correction, and still also over-correction for higher orders. Top panels are Pass 1, bottom Pass 2. The CCD boundary seen in the HEG -1 order (upper left panel) is barely visible after the second pass (lower left panel).

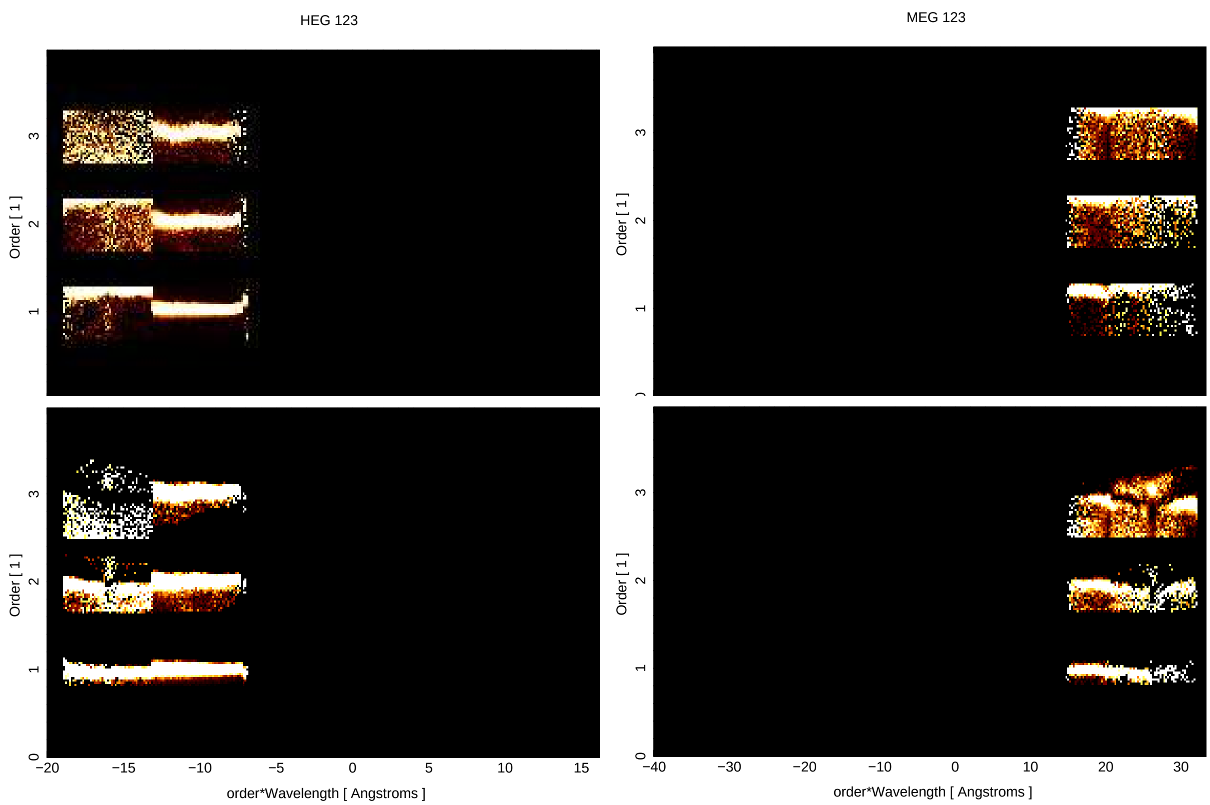

Figure 4: Another example, ObsID 3505 (Sco X-1, zeroth order well off the bottom of the array) showing large correction. Top panels are Pass 1, bottom Pass 2. Here, the first pass flat order sorting region is too narrow, clipping the HEG -1 and MEG +1 order significantly. In this extreme case, using `tg_resolve_events` parameters `osort_lo,hi` of 0.4 for the first pass is better.

Appendix A Specification Notes for tg_resolve_events

For the record, these are notes sent to CXC/DS as descriptive changes required for tg_resolve_events.

```
1 - input required:
   -- infile - evt
      input event file conditions which must be met for new
      processing:
        a) CHIPY_TG column exists
        b) TIME_ADJ == GRATING
        c) GRATING == HETG

      WARNING cases, in which data can be reprocessed in the
      normal mode:

      If (a and c are true) and (b is false),

       WARNING: HETG/CC-mode detected, but acis_process_events
            CC/Grating reprocessing not detected; proceeding
            with normal tg_resolve_events processing.

      OR if (a is true) and (b is true) and (c is false):

      WARNING: Special CC-mode reprocessing is not necessary for LETG;
                proceeding with normal tg_resolve_events processing.

   -- If we are doing a 2nd pass, we will only be doing the
      order-sorting.  The 2nd pass of acis_process_events re-computes
      the CTI correction and updates the ENERGY column, which is used
      used in order-sorting.

      The event TIMEs have also changed, which may affect any
      transformations involving aspect.

      Grating columns TG_PART, TG_MLAM, TG_M and TG_LAM may change
      (compared to the first pass, additional events may be included,
      some resolved before might now be rejected, some may change
      TG_PARTS from MEG to HEG or vice versa, primarily due to changes
      in ENERGY in acis_process_events).

      If TG_PART changes between MEG and HEG, then CHIPY_TG will change.

      If the order-sorting table (osipfile, or osort_lo, osort_hi)
      parameters are different on the second pass, then the TG_PART,
      TG_MLAM, TG_M, TG_LAM are likely to change.
```

```
      We are NOT re-computing other grating columns: TG_R, TG_D,
      TG_SRCID, CHIPY_ZO.

      Events which have TG_PART==0, TG_M==0 will not change.

2 - order-sorting, grating coordinates:

  For CC-mode, we first assume all events not in zeroth order
  (TG_PART == 0) are MEG (TG_PART == 2 ), and do order-sorting.

  MEG events assigned to even orders (other than 0) or background
  (TG_M=99) are then considered to be HEG (TG_PART==1) and
  order-sorting computation is re-done for that grating type.

  Assignment of grating parts, HEG or MEG, allow CHIPY_TG to be computed.

 For every event for which
        ( 0 <= CHIPY_TG <= 1023 )  AND  ( TG_PART != 0 )

  - Re-compute the part (TG_PART), TG_MLAM (which depends on the
    grating period), order (TG_M), and TG_LAM.
  - Re-compute CHIPY_TG for TG_PART == 1 or 2
    (necessary only for any events for which TG_PART changed between
    HEG and MEG parts; will be effectively constant for any which
    did not change, to within very small changes in event TIME, via
    the aspect solution).
---------------------------------------------------------------------
 The relevant source code functions are tg_process_event_data_CC(),
 (in tg_process_event_data.c), and tg_order_resolve() (from
 tg_rm_table_routines.c, and tg_determine_energy_hilo() in
 tg_rm_newtable_routines.c).

  -- Aspect only needs to be re-applied in order to compute CHIPY_TG.
     (or, pass 1's values could be cached and the swapped for those
     HEG-MEG event exchanges).
  -- Re-compute order via tg_order_resolve(), and update the
     TG_PART, TG_MLAM, TG_M, TG_LAM columns.

Tests on one dataset showed that less that 1% of events change
TG_PART. (This will be larger in the event of a change in osip
parameters, or with possible a_p_e updates to the PHA, ENERGY
computation, which are pending.)
```