

Checking FT1 files in the context of the FT2 reprocessing

Maria-Elena reprocessed a complete set of files for a run for 634008661:

root://glast-test-rdr.slac.stanford.edu/glast/Data/Flight/Level1/LPA/dev/5.9/ft2/gll_pt_p310_r0634008661_v000.fit
root://glast-test-rdr.slac.stanford.edu/glast/Data/Flight/Level1/LPA/dev/5.9/ft2Seconds/gll_pt1s_p310_r0634008661_v000.fit
root://glast-test-rdr.slac.stanford.edu/glast/Data/Flight/Level1/LPA/dev/5.9/ft1/gll_ph_p305_r0634008661_v000.fit
root://glast-test-rdr.slac.stanford.edu/glast/Data/Flight/Level1/LPA/dev/5.9/ls1/gll_ev_p305_r0634008661_v000.fit

Checks on run 634008661

- Philippe: I've compared the new FT1 file with the original file root://glast-rdr.slac.stanford.edu/glast/Data/Flight/Level1/LPA/prod/5.7/ft1/gll_ph_p305_r0634008661_v000.fit and the key variables (energy, direction, event class and types) are identical. The nb of events in the files are also identical.
- Don:
I compared the original and new photon and event files that Maria Elena created. The data in the photon (FT1) files match exactly. The event files (LS1) have small differences in the PtAlt, PtLat, and PtMagLat columns. Looking at a few examples, the differences are in the fourth significant figure or lower (e.g., 537.501 vs 537.494 in PtAlt). I assume this is because of the geodetic position calculation implemented for the FT2 files. I compared PtAlt values in the LS1 file against RAD_GEO values in the FT2 file for the reprocessed run and they are consistent, so that checks out. I supposed if we were to do everything properly we would need to reprocess all the event files, too, but I think we can just document the issue and move forward since the differences are small.

Later she created a few more FT1 files for several runs:

root://glast-test-rdr.slac.stanford.edu/glast/Data/Flight/Level1/LPA/dev/5.9/ft1/gll_ph_p305_r0600163450_v000.fit
root://glast-test-rdr.slac.stanford.edu/glast/Data/Flight/Level1/LPA/dev/5.9/ft1/gll_ph_p305_r0601783627_v002.fit
root://glast-test-rdr.slac.stanford.edu/glast/Data/Flight/Level1/LPA/dev/5.9/ft1/gll_ph_p305_r0634002957_v000.fit
root://glast-test-rdr.slac.stanford.edu/glast/Data/Flight/Level1/LPA/dev/5.9/ft1/gll_ph_p305_r0633997253_v000.fit

Don: I found some differences this time. While the 600163450 and 601783627 files are identical to the latest versions received at the FSSC, the last two files both have differences in the column 18 (LIVETIME) values starting at rows 11859 and 20150, respectively. For instance for 634002957

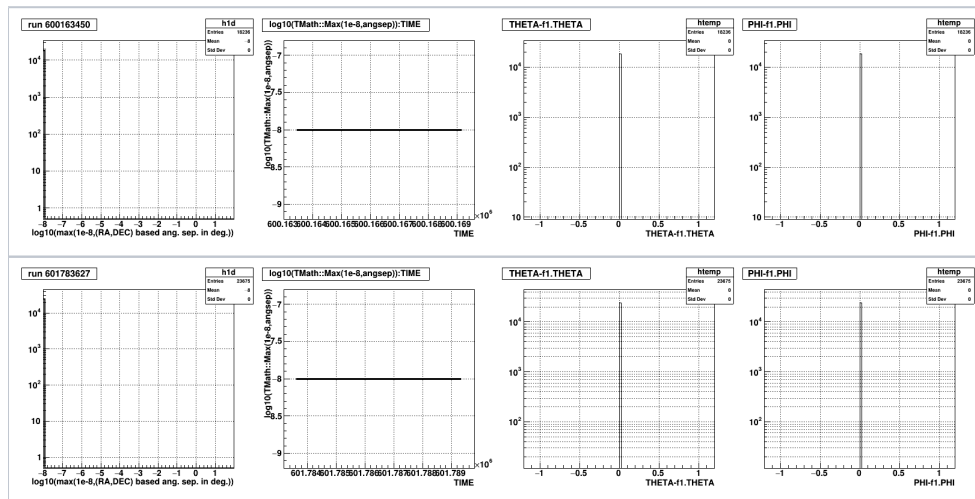
```
< row 11859, col 18: 0.0464113  
> row 11859, col 18: 122.326
```

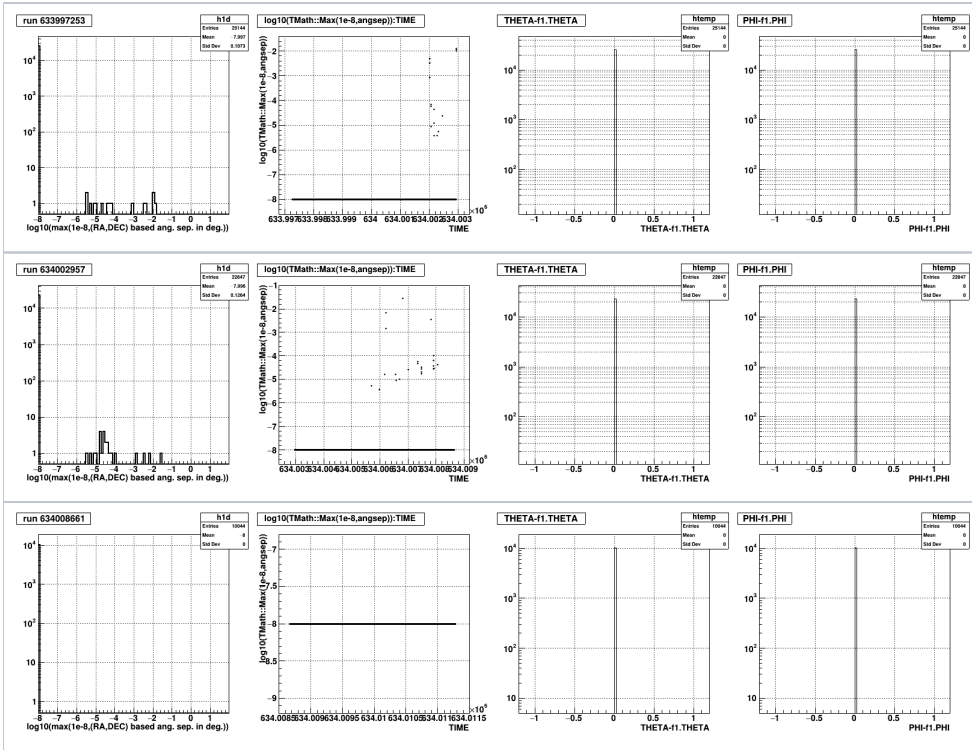
The first number is the for "old" file while second is the "new" file. Every row after this seems to have a different value for LIVETIME. The other run 633997253 shows the same behavior. The fdiff utility doesn't report any other differences in the data (e.g, the number of photons in the files are the same).

According to the confluence page [On FT1 Livetime](#): "So for now, the LIVETIME values in the FT1 files **should be ignored**. Again, they are not currently used by the Science Tools, and in particular do not affect calculations of exposure, which are made with the livetime accumulations in FT2 files."

Checking (RA,DEC) directions: (Philippe)

Following Seth's advice, the following plots were updated using the [Haversine](#) formula, which removed a peak at 1e-6.





The 2 runs with significant differences are the ones reported by Don with the weird LIVETIME differences.

Since the (theta,phi) values don't change, the differences occur when going from instrument to sky coordinates. But the differences are on the event basis so it doesn't look like a calibration difference between the 2 processings.

There is no correlation between the angular differences and the LIVETIME differences.

List of differences

```
633997253 root [8] f0->Scan("TIME-633997256.91035:RA:DEC:RA-f1.RA:DEC-f1.DEC","angsep>1e-6","colsize=15")
*****
* Row * TIME-633997256. * RA * DEC * RA-f1.RA * DEC-f1.DEC *
*****
* 20844 * 4759.7181751728 * 183.81973266601 * 67.601860046386 * 0.0005340576171 * 0.0008468627929 *
* 20845 * 4759.7695435285 * 287.62429809570 * 54.915344238281 * 0.0057373046875 * -0.000228881835 *
* 20846 * 4759.8253495693 * 199.27514648437 * 54.118850708007 * 0.0046844482421 * 0.0040397644042 *
* 21019 * 4802.3366923332 * 183.78639221191 * 69.426277160644 * 0 * -6.86645507e-05 *
* 21020 * 4802.3548659086 * 294.52005004882 * 17.876571655273 * -6.10351562e-05 * 1.907348632e-06 *
* 21021 * 4802.5739853382 * 312.29360961914 * 6.4632229804992 * 0 * 8.583068847e-06 *
* 21490 * 4907.4797089099 * 105.74105072021 * 85.111404418945 * 0.0002593994140 * 3.814697265e-05 *
* 21491 * 4907.6401162147 * 187.39306640625 * 41.797592163085 * -1.52587890e-05 * -3.81469726e-06 *
* 21492 * 4907.6606949567 * 279.35891723632 * -19.56085777282 * 0 * -3.81469726e-06 *
* 22036 * 5018.6335679292 * 173.26010131835 * 62.035305023193 * 0 * 3.814697265e-06 *
* 22225 * 5059.6422821283 * 292.18984985351 * 16.839588165283 * 0 * -5.72204589e-06 *
* 22973 * 5212.6489586830 * 314.70776367187 * 36.770217895507 * -3.05175781e-05 * 0 *
* 25093 * 5679.9408695697 * 262.84677124023 * 21.012033462524 * 0.0036315917968 * 0.0106353759765 *
* 25094 * 5679.9463177919 * 281.66174316406 * 13.183306694030 * 0.0039978027343 * 0.0091753005981 *
* 25095 * 5679.9687132835 * 24.909524917602 * 73.471138000488 * 0.0296649932861 * -0.009437561035 *
*****
==> 15 selected entries
```

634002957	<pre> root [6] f0->Scan("TIME-634002960.80257:RA:DEC:RA-f1.RA:DEC-f1.DEC", "angsep>1e-6", "colsize=15") ***** * Row * TIME-634002960. * RA * DEC * RA-f1.RA * DEC-f1.DEC * ***** * 12595 * 2745.5844659805 * 122.91301727294 * 62.059600830078 * -7.62939453e-06 * 3.814697265e-06 * * 13259 * 3023.5338615179 * 278.90939331054 * 39.736934661865 * 0 * 3.814697265e-06 * * 13663 * 3205.6217298507 * 129.55728149414 * 58.954723358154 * -3.05175781e-05 * -3.81469726e-06 * * 13786 * 3257.8420428037 * 282.16900634765 * 57.907760620117 * 0.0002746582031 * 0.0014915466308 * * 13787 * 3258.0019516944 * 43.209045410156 * 72.377243041992 * 0.0084953308105 * -0.006492614746 * * 14646 * 3602.6789500713 * 306.63739013671 * 79.018928527832 * -3.05175781e-05 * -1.52587890e-05 * * 14670 * 3610.7295476198 * 279.90295410156 * 81.300567626953 * 3.051757812e-05 * -7.62939453e-06 * * 15040 * 3747.5052045583 * 349.90283203125 * 89.010185241699 * -0.000610351562 * 0 * * 15298 * 3848.9711569547 * 301.80361938476 * 83.319664001464 * 0.2264404296875 * 0.0086517333984 * * 15557 * 4055.3627098798 * 262.99191284179 * -30.51374626159 * -3.05175781e-05 * -3.81469726e-06 * * 16742 * 4396.6694991588 * 63.681659698486 * -41.06742477416 * 1.144409179e-05 * -5.72204589e-05 * * 16743 * 4396.6903454065 * 89.633560180664 * -43.20569992065 * 2.288818359e-05 * -4.57763671e-05 * * 17579 * 4512.4310297966 * 296.17922973632 * -31.47754669189 * 3.051757812e-05 * 1.144409179e-05 * * 17580 * 4512.4617695808 * 64.590911865234 * -50.29770660400 * 3.051757812e-05 * -2.67028808e-05 * * 17581 * 4512.5952510833 * 88.158439636230 * -40.75327301025 * 2.288818359e-05 * -1.14440917e-05 * * 17582 * 4512.5988693237 * 51.922626495361 * -46.82504653930 * 1.144409179e-05 * -1.52587890e-05 * * 17583 * 4512.6140358448 * 48.694889068603 * -40.54943084716 * 1.144409179e-05 * -1.52587890e-05 * * 19758 * 4858.8858329057 * 101.87527465820 * -46.33024597167 * -0.005050659179 * -0.000286102294 * * 20206 * 4946.3916902542 * 82.908439636230 * -44.45342254638 * 6.103515625e-05 * -9.53674316e-05 * * 20207 * 4946.5567092895 * 85.310897827148 * -45.91974639892 * 3.814697265e-05 * -5.72204589e-05 * * 20208 * 4946.6146457195 * 127.75340270996 * -48.30835342407 * 5.340576171e-05 * -1.14440917e-05 * * 20209 * 4946.6949472427 * 59.188949584960 * -29.97546958923 * 0 * -2.86102294e-05 * * 20379 * 4980.6338151693 * 90.215309143066 * -48.75658035278 * -3.05175781e-05 * 2.288818359e-05 * * 20918 * 5103.6386698484 * 82.866554260253 * -47.59976577758 * -4.57763671e-05 * 3.051757812e-05 * ***** ==> 24 selected entries </pre>
-----------	--

The fact that for some events there is a difference in one of the RA,DEC variables and not in the other one is very strange.

I note that some numbers appear several times. Very weird !

And:

- $3.814697265e-06 * 2 = 7.6293945e-06$
- $3.814697265e-06 * 3 = 1.15258789e-05$
- $3.814697265e-06 * 8 = 3.0517578e-05$

Merit files

I've compared:

- original=root://glast-rdr.slac.stanford.edu/glast/Data/Flight/Level1/LPA/prod/5.7/merit/r0634002957_v001_merit.root
- new=root://glast-test-rdr.slac.stanford.edu/glast/Data/Flight/Level1/LPA/dev/5.9/merit/r0634002957_v000_merit.root

And I see the same differences for the variable FT1Ra and FT1Dec.

Pieces of code (/nfs/farm/g/glast/u52/repomanBuild/redhat6-x86_64-bit-gcc44/GlastRelease/20-10-04-gr07/)

The variables FT1Ra and FT1Dec are filled in AnalysisNtuple/src/FT1Alg.cxx:

- `gps = astro::GPS::instance();`
- `double etime(headertime());`
- `gpstime(etime);`
- `glastDir= Hep3Vector(CTBBestXDir, CTBBestYDir, CTBBestZDir);`
 - **this information is the same in both merit files**
- `SkyDir sdir(gps->toSky(glastDir));`
- `m_ft1ra = sdir.ra();`
- `m_ft1dec = sdir.dec();`

The gps object lives in astro/src/GPS.cxx:

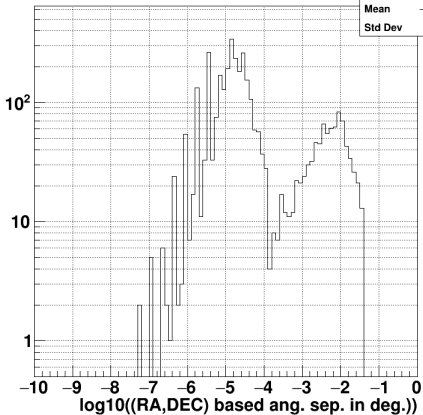
- `astro::SkyDir GPS::toSky(const CLHEP::Hep3Vector& latdir, double met)`
 - `CLHEP::Hep3Vector tdir(- (m_currentPoint.rotation() * m_alignment * latdir));`
- `void GPS::update(double inputTime)`
 - `//this function calculates all the relevant position and orientation info`
 - **the differences could come from this part of the code**

Plots with merit files from run 634002957:

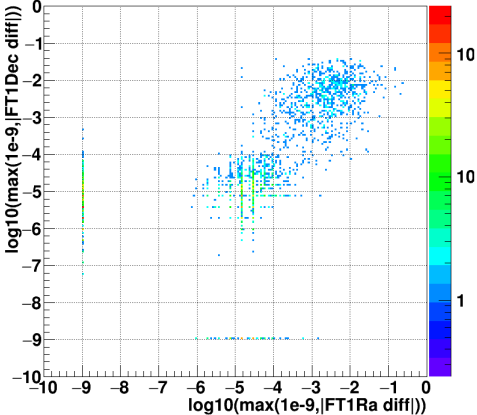
For each column (either for or cal directions)

FT1Ra,FT1Dec direction

run 634002957

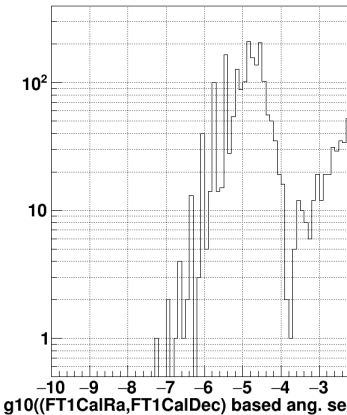


run 634002957



FT1CalRa,FT1CalDec direction

run 634002957

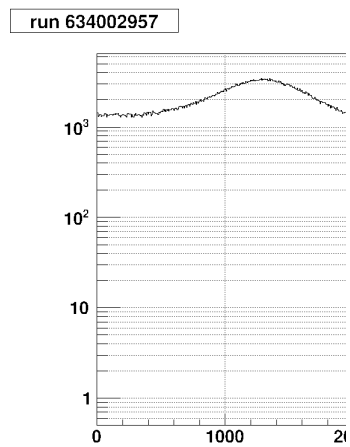
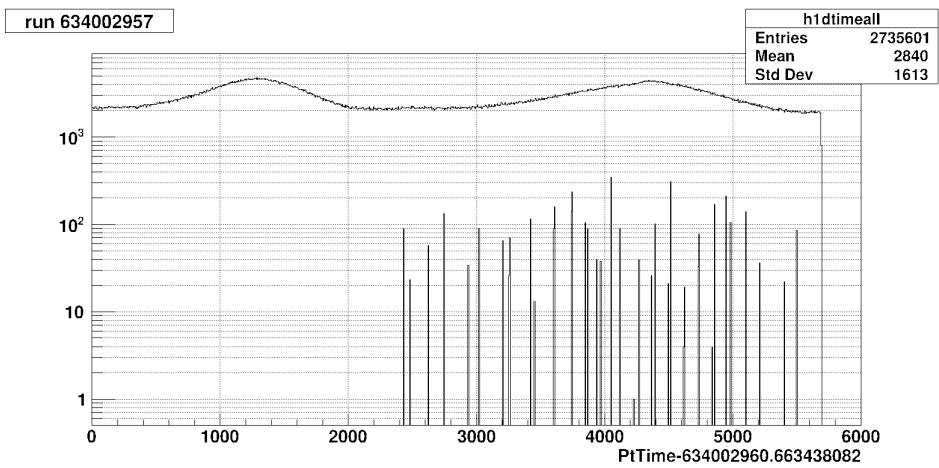


left : logarithm of the (RA, Dec) angular separation in degrees for events within a given cone $> 1^\circ$ - 10

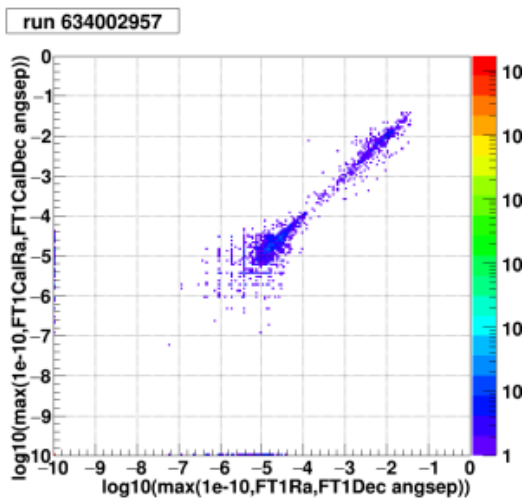
F
o
r
tk
r:
3
4
4
2
e
vt
s
o
u
t
o
f
2
7
3
5
6
01

F
o
r
c
al
:
2
3
0
7
e
vt
s
o
u
t
o
f
1
8
0
6
3
07

P
t
T
i
m
e
d
i
s
t
r
i
b
u
t
i
o
n
f
o
r
a
l
l
e
v
t
s
(
b
l
a
c
k
)
a
n
d
t
h
e
o
n
e
s
w
i
t
h
a
n
g
s
e
p
>
1
e
-
1
0
(
r
e
d
)



Very weird: some events with Calangsep > 1e-10 have Tkrangsep = 0 and vice versa!



Other important differences with the MeritTuple Pt variables

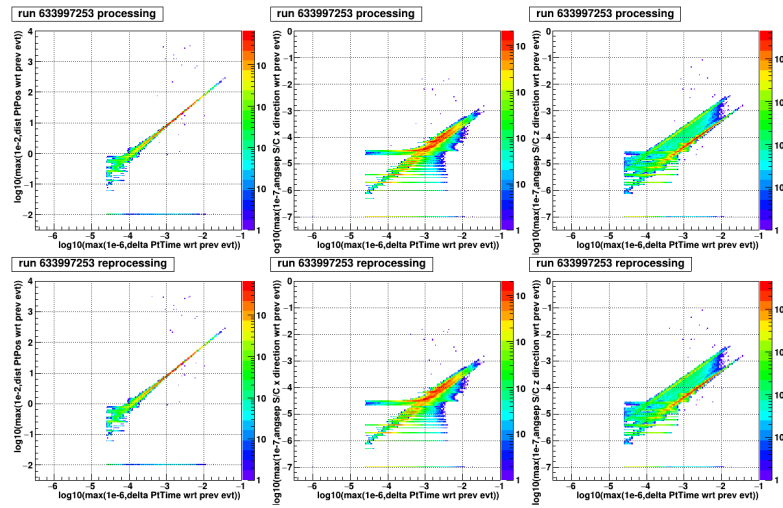
While answering Seth's request about providing the times of the problematic events, I've started comparing more variables and I've discovered other differences concerns some Pt variables in the MeritTuple:

- PtPos[3] = current orbit position
- PtRax, PtDecx = equatorial coordinates for orientation of S/C x-axis
- PtRaz, PtDecz = equatorial coordinates for orientation of S/C z-axis

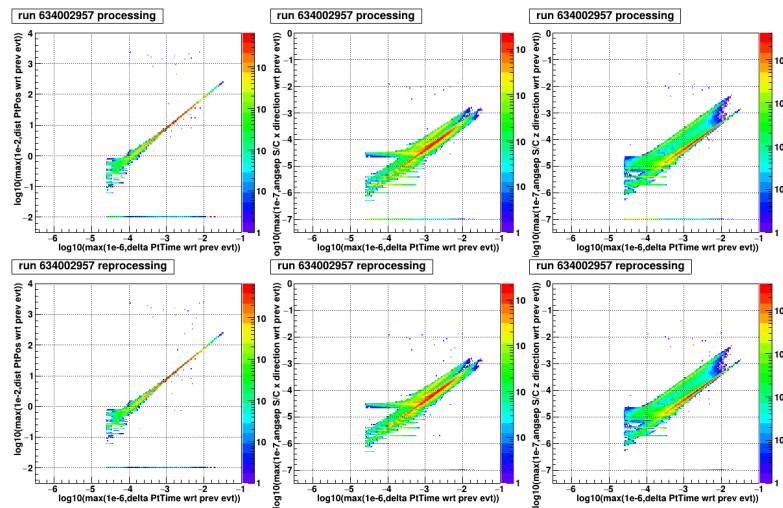
As can be seen in the pieces of code below, these variables are a copy of some quantities of the gps object. Unfortunately they do not include the quaternion information.

These quantities are the result of some interpolation between values computed from the FT2 file. So we expect them to change continuously with time. One way to check it is to look at the difference between 2 consecutive events w.r.t. to the delta time between these events:

633997253



634002957



One can see:

- a "main" correlation branch, with clear sub-branches for the PtRaDec x and z directions
- some events with a difference (graph y-axis) = 0
 - it occurs often that the quantity does not change during several consecutive events
- some events scattered away from the main branch ()
 - they correspond to when the information goes back to normal

The non-differences between consecutive events are not exactly the same between the processing and reprocessing merit files. For the position information for instance:

- processing && reprocessing : 3612 evts
- processing && !reprocessing : 466 evts
- !processing && reprocessing: 464 evts

In one processing flavor, there is not a 100% correlation between the deltas of these variables (delta pos=0 does not imply delta S/C z-axis direction = 0, and vice versa) but if one delta is 0, then the fraction of delta=0 in the other variables is larger.

I've dumped the Merit information in a [text file](#) (for the run 634002957, starting at 1st event with (Ra,Dec) diff between processings) with the following information:

1345961 id 7012702 t 634005394.600154 as 0.000000 0.000000 d 1.224745 ft2 d 0.000000 1.224745 x 0.000008 0.000008 z 0.000018 0.000018

- first row = row number in MeritTuple
- id = EvtEvent_Id
- t = PTime
- as = (Ra,Dec) angsep and cal (Ra,Dec) angsep between processings
- d = PtPos distance between processings
- ft2 d = PtPos distance w.r.t. to FT2 file information for processing and reprocessing
- x = S/C x-axis ang sep w.r.t. to FT2 file information for processing and reprocessing
- z = S/C z-axis ang sep w.r.t. to FT2 file information for processing and reprocessing

If the event is bad (the maximum of as is >0 or if d>0), the line starts with:

- !D! if as=0 and d>0
- !A! if as>0 and d=0
- !B! if both as=0 and d=0

To write the text file, I loop over the merit events, loading the FT2 second information when needed (so that START<PtTime<STOP; and a line is written starting with "Loading new ft2 second") and I write the information as soon as there is a bad events (and also the previous events when we start having bad events) and I stop writing it after 100 good evts in a row (so that we can see these problems mix with the good ones).

The bad events come in bunches but in 2 flavors:

- consecutive bad events (e.g. the first bunch in the text file: 100 bad events in a row)
 - occurring right after a new FT2 second has been loaded
 - for one of the processing, the Pt variables stick to the values in the FT2 file
- bad and good events mixed together: e.g. second bunch in the text file: ~25 bad events among good ones
 - not occurring right after a new FT2 second has been loaded
 - the Pt variables don't stick to the values in the FT2 file (but it doesn't mean that they agree between processings)

The main conclusion is that there is something wrong regarding the PtVariables , possibly related to the FT1 Ra,Dec direction of the events, in both processings !

Related pieces of code:

AnalysisNtuple/src/PointingInfo::execute(const astro::GPS& gps)

- start = gps.time();
- CLHEP::Hep3Vector pos_km = gps.position();
- CLHEP::Hep3Vector location = 1.e3* pos_km;
- sc_position[0] = location.x(); sc_position[1] = location.y(); sc_position[2] = location.z();
- ra_zenith = gps.zenithDir().ra(); dec_zenith= gps.zenithDir().dec();
- ra_scx = gps.xAxisDir().ra(); dec_scx = gps.xAxisDir().dec();
- ra_scz = gps.zAxisDir().ra(); dec_scz = gps.zAxisDir().dec();

The interpolation is performed during in void GPS::update(double inputTime)

- // if history, just ask the history to interpolate the table, or whatever
- if(m_rockType == HISTORY || m_rockType == HISTORY_X_EAST){
- m_currentPoint = (*m_history)(inputTime);

The last command is defined in const astro::PointingInfo& PointingHistory::operator()(double time)const throw(TimeRangeError)

- m_currentPoint = h1.interpolate(h2, prop, time);

where m_currentPoint contains the following quantities:

- CLHEP::Hep3Vector m_position; ///< position (km)
- Quaternion m_q; ///< orientaion of the SC: xaxis direction
- EarthCoordinate m_earth; ///< Earth position: (lat, lon)

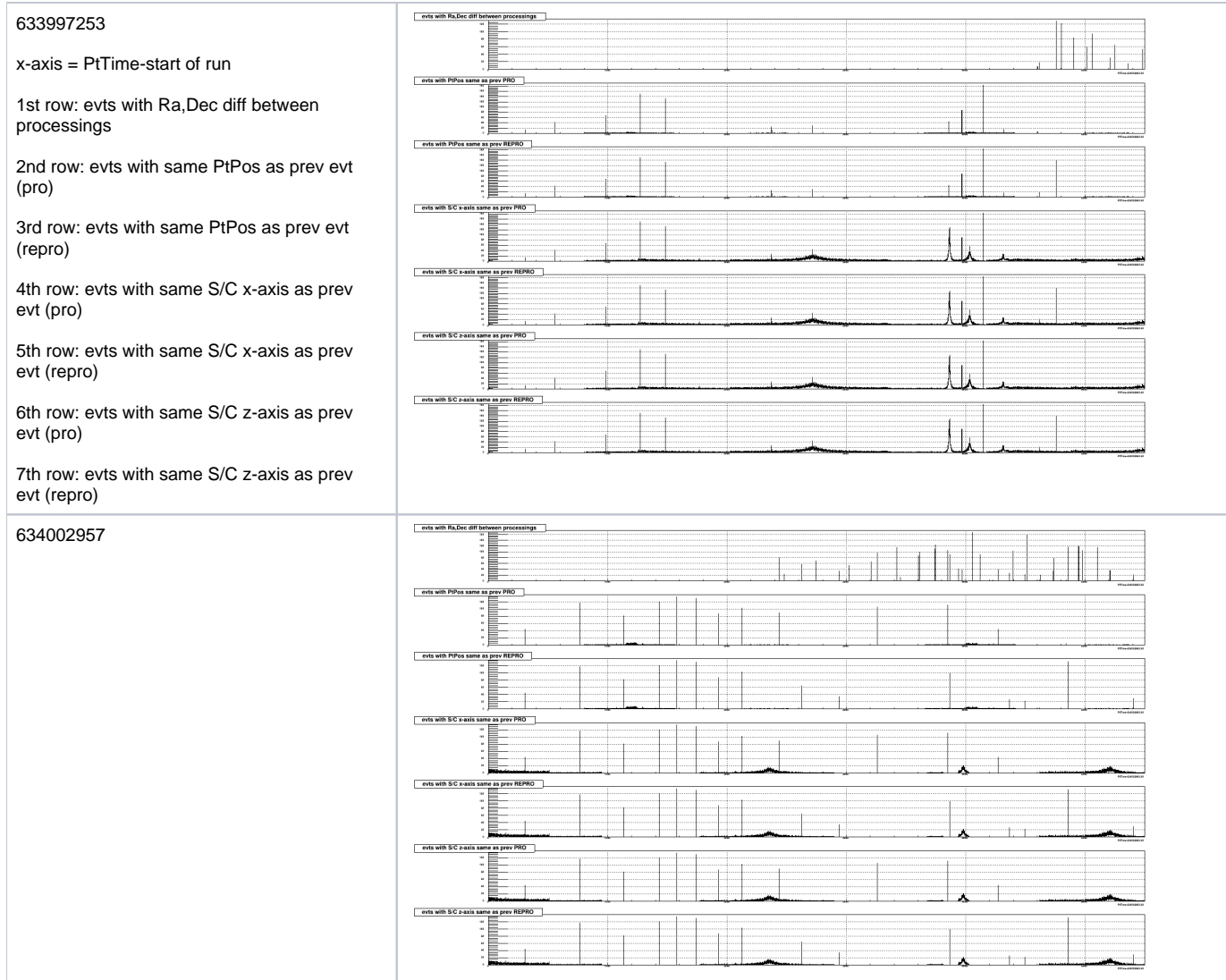
The interpolation is done independently for these quantities (see Seth's comment on the quaternion interpolation)

More on Pt variables and the question on correlation with Ra,Dec difference primary problem

The following plots show the time of problematic events (either Ra,Dec diff between processings or evts with PtPos or S/C x or z axis identical to prev evt). One can clearly see that there is no 100% correlation between these problems.

I was misled by run 634002957 and the [corresponding txt file](#) in which I wrote information starting from the first evt with Ra,Dec diff between processings. So I missed all the evts with PtPos or S/C x or z axis identical to prev evt at the beginning of the run for which there is no Ra,Dec diff between processings.

But nevertheless some spikes do correspond. And the spikes in the pro and repro plots are not identical, showing a different behaviour in the original and new processings.



Comparison of Pt variables with home-made FT2 interpolation

Using the FT2 information, one can predict/interpolate the distance w.r.t. to the position at START and the angular separation w.r.t to the direction at START:

- PtPos
 - compute the distance between SC_POSITION at START and at STOP (=START of next event)
 - multiply this distance by PtTime-START
- S/C x-axis and z-axis:
 - compute the angular separation between S/C direction at START and at STOP (=START of next event)
 - multiply this angular separation by PtTime-START

The following plots shows the absolute difference between the measured and predicted quantities:

633997253

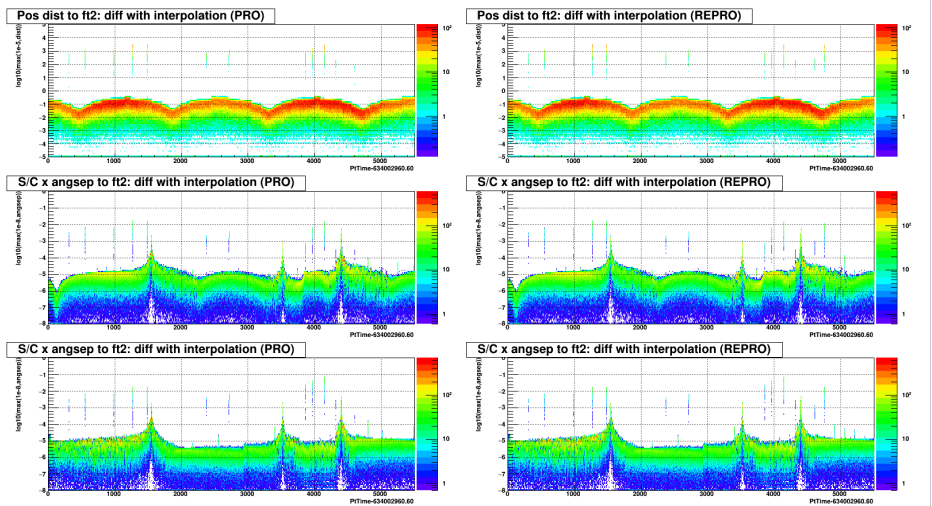
left: pro

right: repro

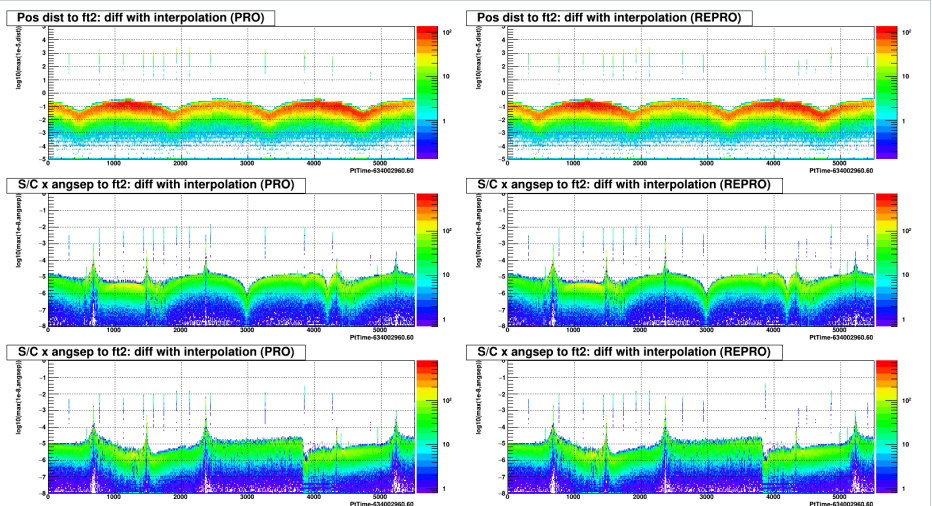
each variable is computed w.r.t. to ft2 value at START

(e.g. the measured pos dist is the distance between

the event PtPos and SC_POSITION at START)



634002957



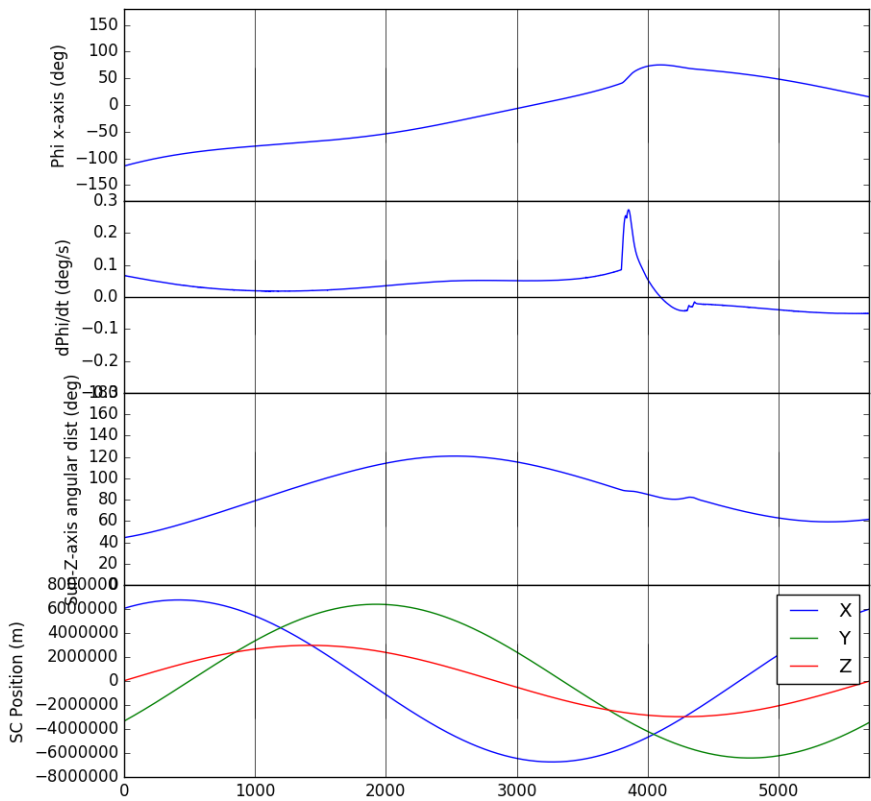
One can see the outliers come in bunches. A lot of bunches are common between pro and repro but some are different.

The pos dist and S/C ang sep outliers are 100% correlated. The most convenient way to detect them is to select $\text{abs}(\text{pos dist to ft2: diff with interpolation}) > 1\text{m}$.

Since we don't know yet whether the FT1 difference is 100% related to the interpolation problem, a separate confluence page to discuss the latter has been created : [Bad interpolation of FT2 information during processing](#)

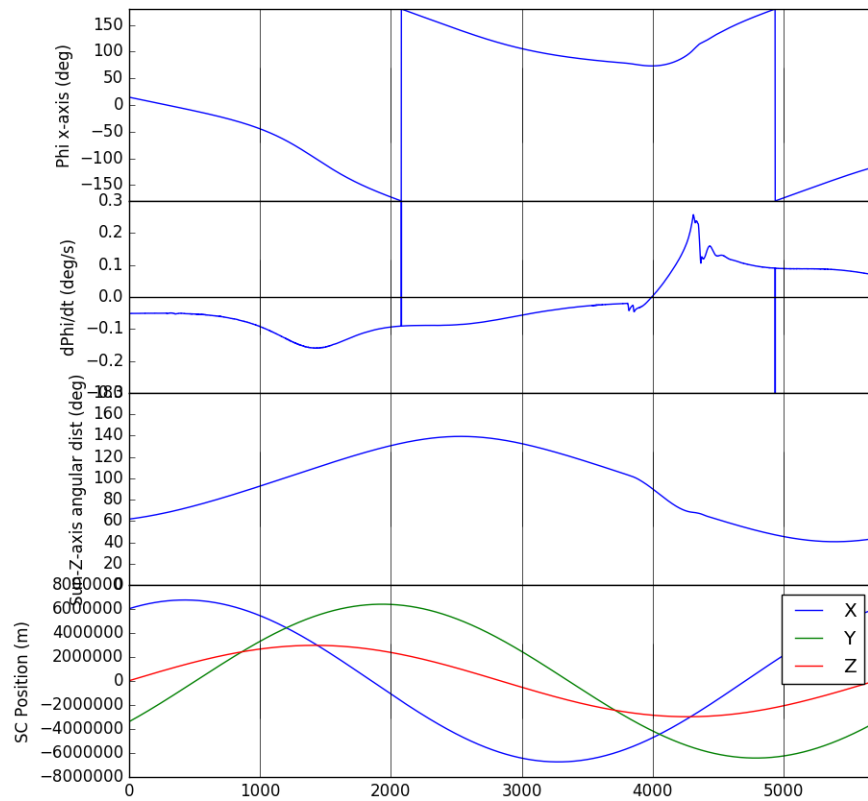
(SD, 16 Feb 2021) Here is a quick summary of some information in the ft2seconds files for the two runs Philippe reports on above. I was curious to see if the broad periods of increased fraction of events with 'stuck' spacecraft X and Z axes corresponded with any special part of the orbit (since they look periodic in Philippe's plots). I was especially interested to see if they corresponded to periods when the spacecraft was rotating most quickly around the Z axis. The angle Phi in the plots below is derived by rotating the SC_Rax and SC_Decx to a coordinate system that has SC_Raz, SC_Decx as the pole, with Phi = 0 corresponding to RA = 0. (Big jumps in Phi correspond to wrapping around +/-180 deg.) The first panel below is Phi. The second is the rate of change of Phi. The third is the angular distance between the Sun and the LAT Z axis. The fourth panel is just the SC_POSITION components. The times are seconds since TSTART, as in Philippe's plots. The vertical lines are at 1000 s intervals, also as in his plots.

I can't say I see any compelling correspondence with Philippe's plots. Note that these runs are on successive orbits. Each started and stopped with the ascending node crossing the equator.



634002957

FT2 file: gll_pt1s_p310_r0634002957_v000.fit



(SD, 22 Feb 2021)

The broad features indicating 'stuck' spacecraft pointing directions in Philippe's plots above seem to indicate just conditions when the coordinates are in fact not changing very quickly and successive events can in fact have the same reported spacecraft orientation. Here is a comparison of the rate of events with the same PtRax and PtDecx as the previous event in the Merit file with the rates of change of RA_SCX and DEC_SCX from the pointing history file for r0634002957_v001_merit.root and gll_pt1s_p310_r0634002957_v000.fit. The histogram in the upper panel is consistent with Philippe's (with different binning in time). The broad peaks correspond to times when DEC_SCX in particular is changing only slowly. This indicates that the broad features in the 'stuck' pointing plots are not an issue with the interpolation. (I get different profiles from Philippe's plot for PtRaz and PtDecz but I think the same explanation applies for 'stuck' spacecraft orientation for them as well.)

/glast/Data/Flight/Level1/LPA/prod/5.7/merit/r0634002957_v001_merit.root

