

FT2 reprocessing P130 Vs archived FT2

Reprocessing P130 used the new ft2Util code to produce FT2 files.

Here I present the comparison between the files produced during P130 and the old files, archived in the astro server and produced with the old code.

All plots refers to 1s FT2 files, unless otherwise stated.

For each quantity q in the FT2 file I report a histogram of the fractional difference between the "new" FT2 file (from P130) and the "old" one (from the archive): $(q_{\text{NEW}} - q_{\text{OLD}})/q_{\text{OLD}}$.

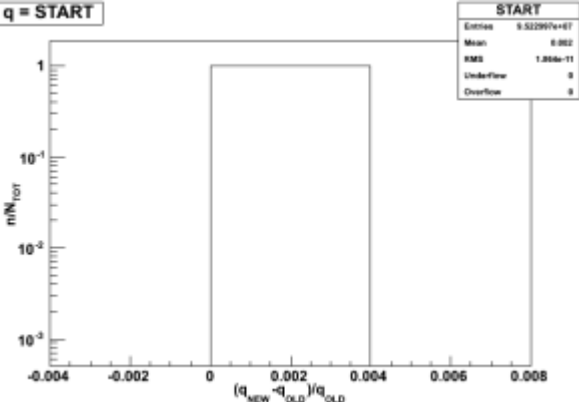
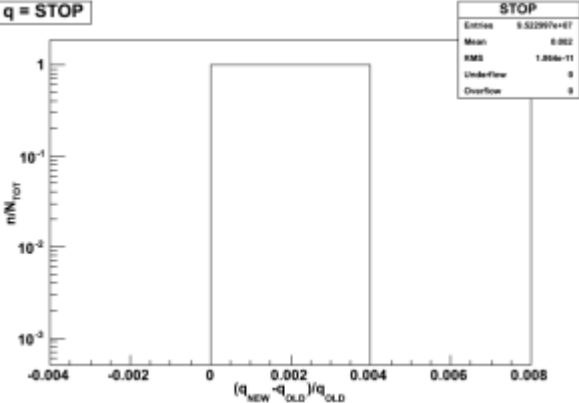
There are two main reasons for the small differences observed in the comparison:

1. The new code does a better job at the margin of the runs: the spacecraft position and attitude (and related quantities) are computed starting from messages in the Magic 7 file. The new code uses the whole M7 file, which usually cover a broader time interval than the run, to compute position and attitude at the start time and end time of the run. The old code, instead, discarded all the messages before the start and after the stop of the run, computing position and attitude at those instants by extrapolation.
2. A bug in the old code (see below for explanations) for the LIVETIME computation.

Note that in the following comparison I **used only time bins in common between the two sets of files** (i.e., with the same START and STOP time). Since we changed the way of handling gaps and other things (like change in mode or configuration), there are a small number of discrepancies in the binning scheme between the two files.

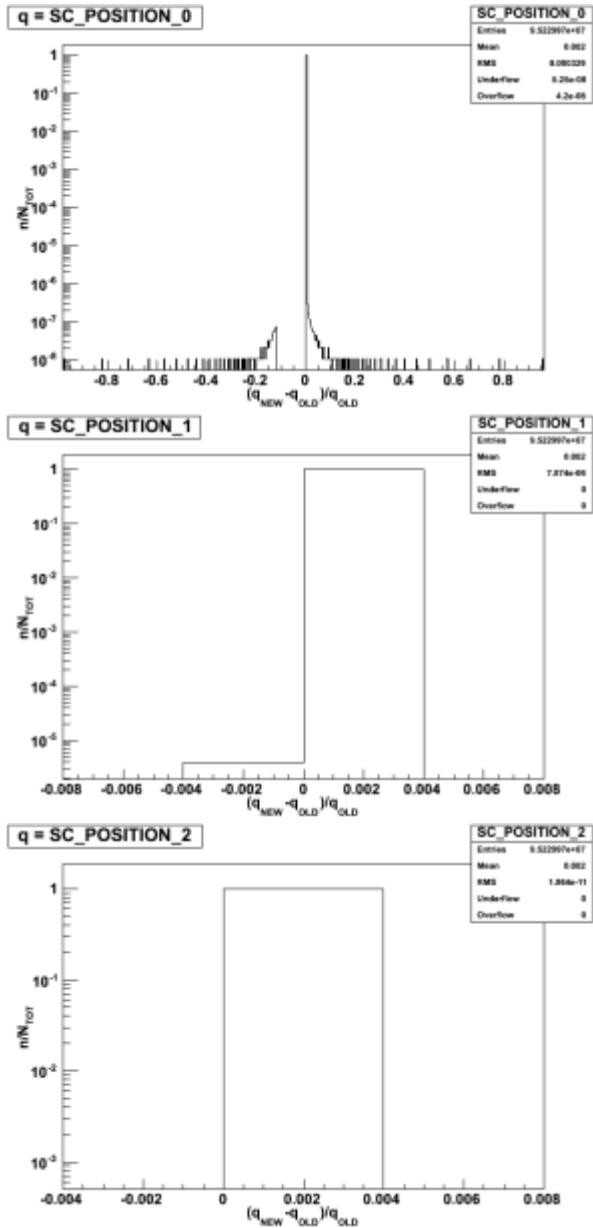
Whole FT2

For this first set of results, I compared each pair of FT2 files from the start of the first bin to the end of the last one. Thus, differences due to Reason 1 above are evident.

Quantity	Columns in the FT2 file	Histograms	Comment
Bin start and stop	START, STOP	q = START  q = STOP 	No differences, by definition.

Spacecraft
position in
ECI
cooridantes
(x,y,z)

SC_POSITIO
N array



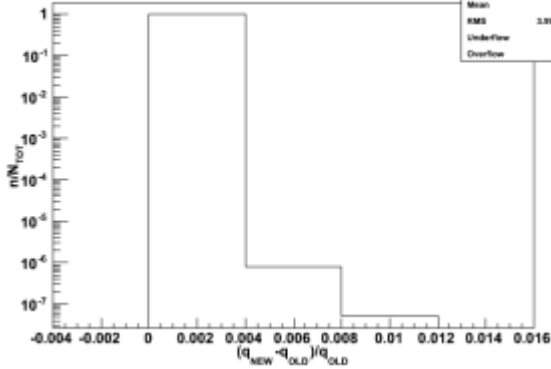
The differences in a very small number of bins are most probably due to Reason 1 above.

Ground point latitude and longitude	LAT_GEO, LON_GEO	q = LAT_GEO q = LON_GEO	Reason 1.
Spacecraft altitude	RAD_GEO	q = RAD_GEO	Reason 1.

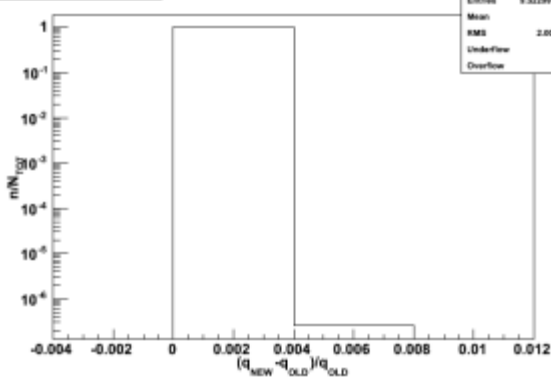
Zenith
direction

RA_ZENITH,
DEC_ZENITH
H

q = RA_ZENITH



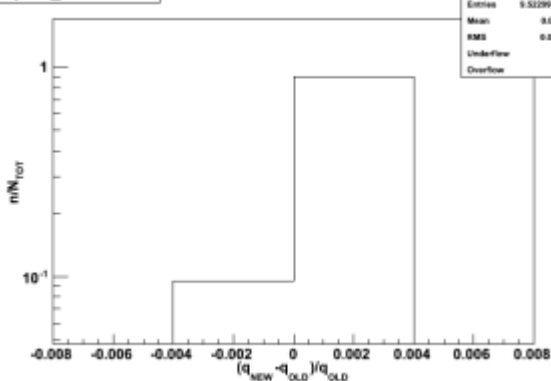
q = DEC_ZENITH



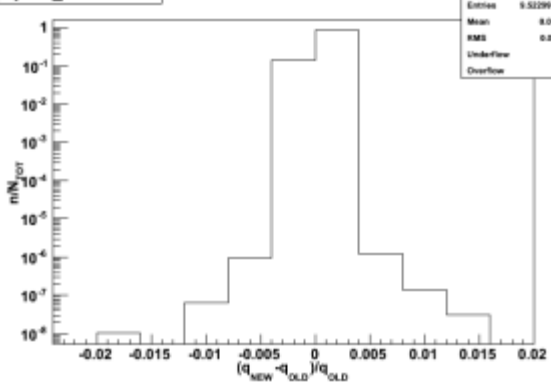
Mcilwain
coordinates

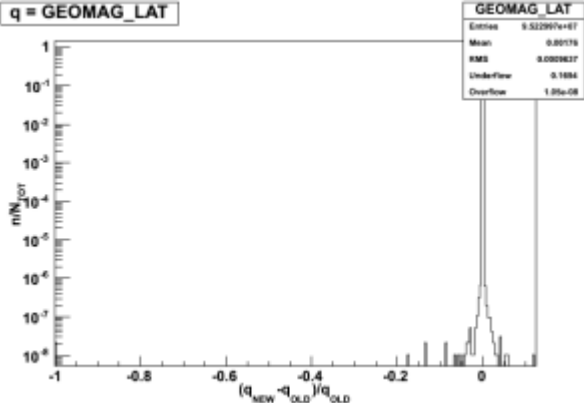
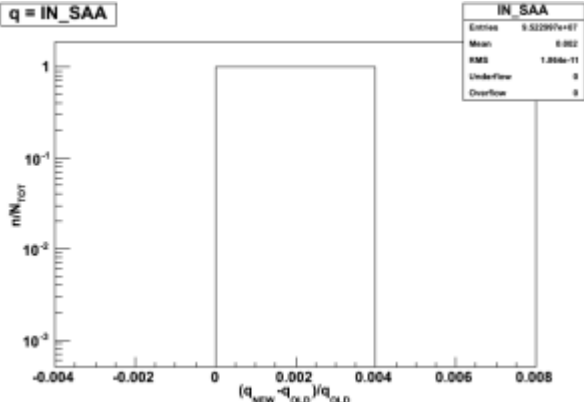
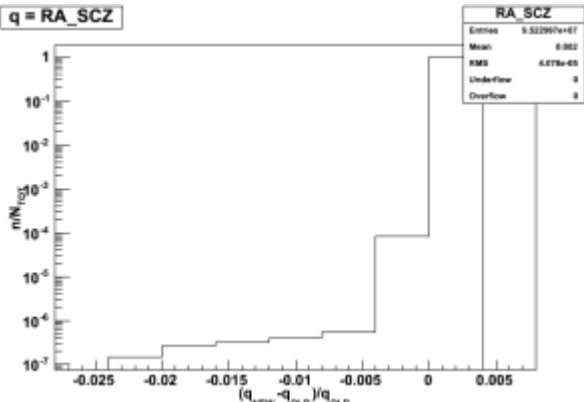
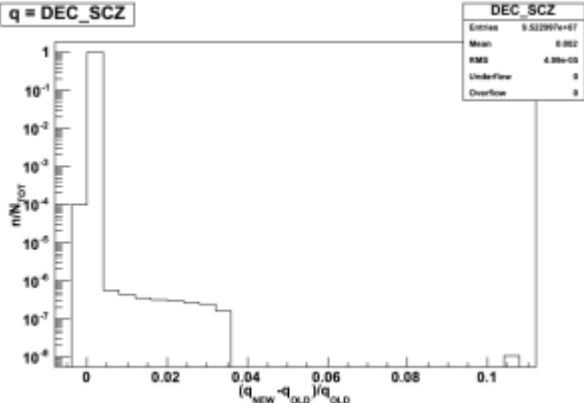
L_MCILWAIN
,
B_MCILWAIN

q = L_MCILWAIN



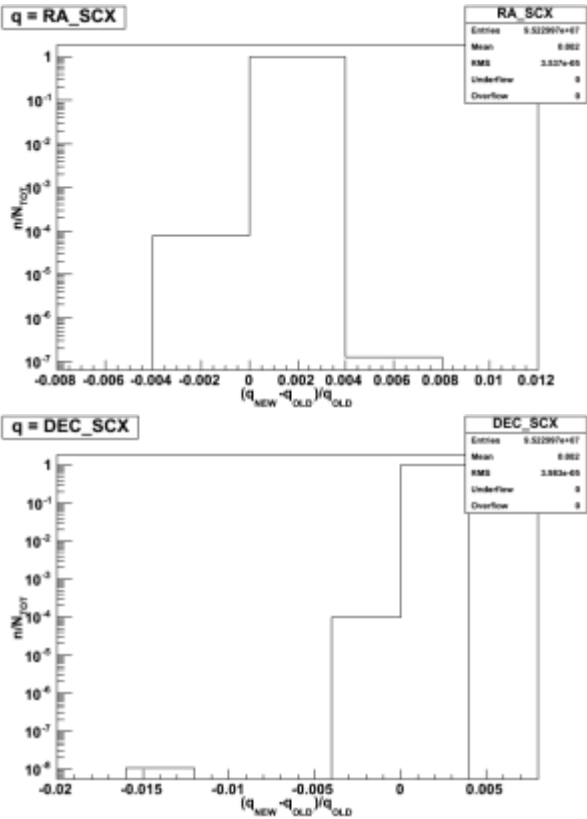
q = B_MCILWAIN



Geomagnetic latitude	GEOMAG_LAT AT	q = GEOMAG_LAT 	16 percent of the time bins are in the underflow bin of the histogram. This is due to a known issue: on Feb. 2, 2010 we changed the quantity from unsigned to signed. Thus, all old FT2 files produced before that date have a always-positive GEOMAG_LAT, while the reprocessed ones have it signed from the beginning of the mission.
South Atlantic Anomaly flag	IN_SAA	q = IN_SAA 	
Z-axis direction	RA_SCZ, DEC_SCZ	q = RA_SCZ  q = DEC_SCZ 	Reason 1.

X-axis
direction

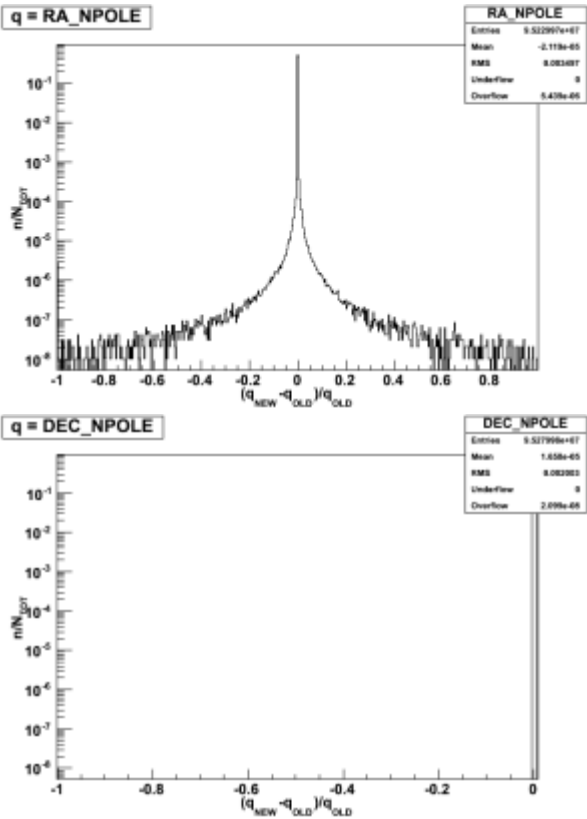
RA_SCX,
DEC_SCX



Reason 1.

Orbital Pole
direction

RA_NPOLE,
DEC_NPOLE

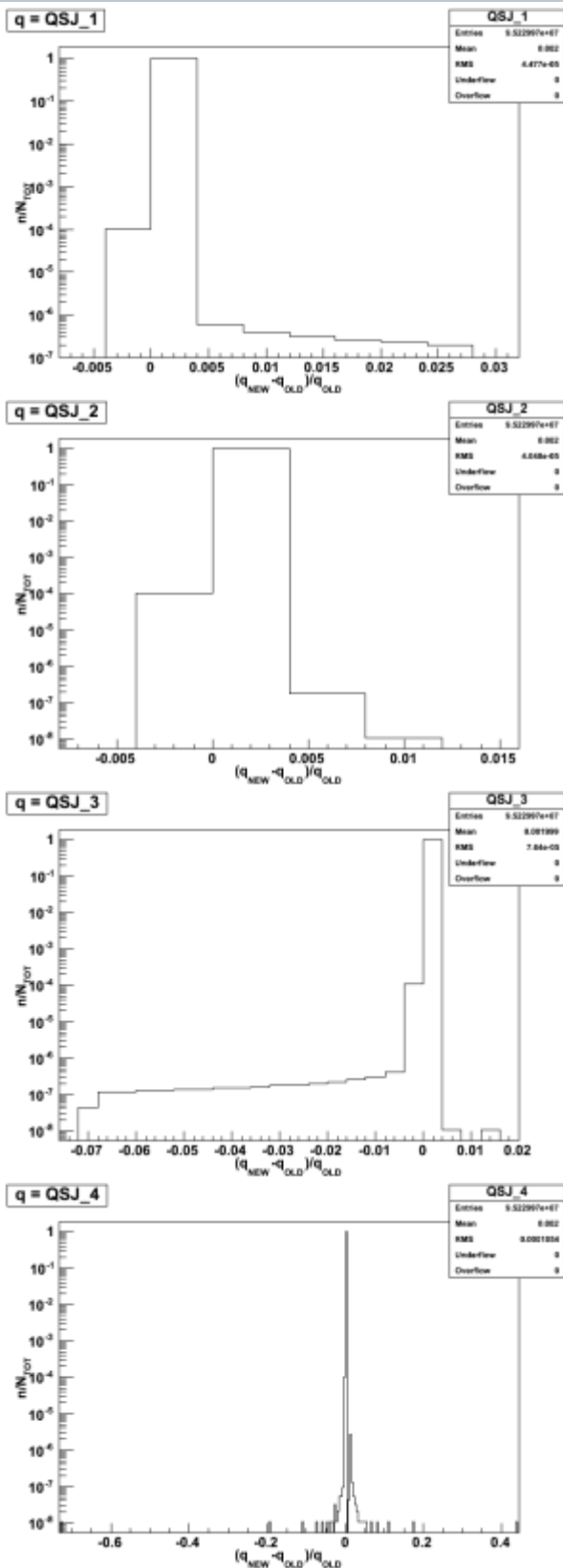


Here there are considerable differences, because the new code compute the coordinates of the Orbital Pole in a different way respect to the old one. If v_i is the position vector for the i -th bin in the FT2 file, the old code computed the orbital pole vector for the i -th bin as the cross product between v_{i-1} and v_i , while the new code uses the product v_i cross v_{i+1} . Thus basically I should compare each i -th entry in the new set with the $(i+1)$ th entry in the old one, to obtain the same results. I did this test, and everything is ok, see below.

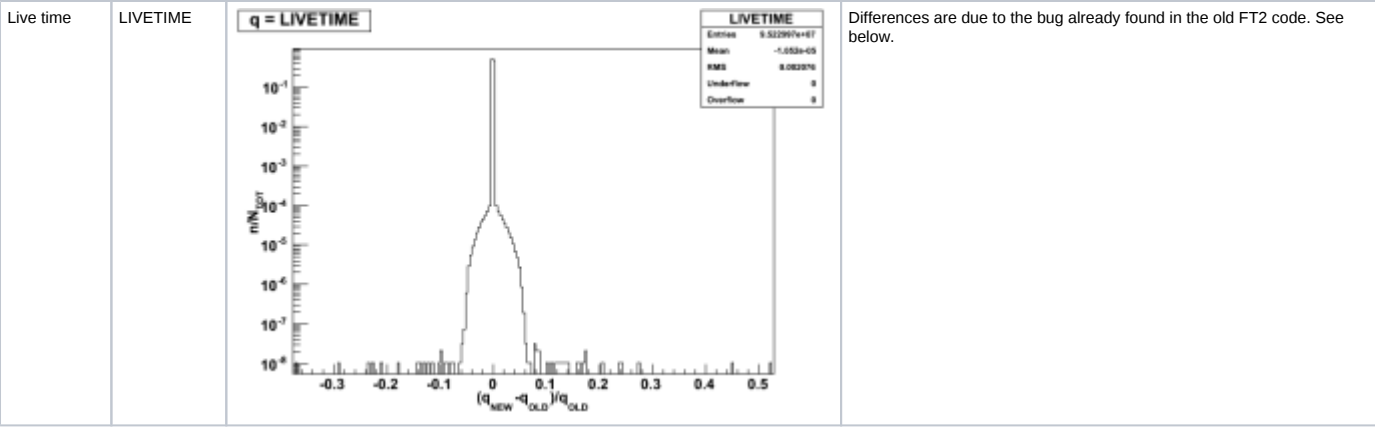
Rocking angle	ROCK_ANGLE E	q = ROCK_ANGLE ROCK_ANGLE Entries: 9.522997e+07 Mean: 0.002 RMS: 6.627e-05 Underflow: 0 Overflow: 0	Reason 1.
LAT GNC mode	LAT_MODE	q = LAT_MODE LAT_MODE Entries: 9.522997e+07 Mean: 0.002 RMS: 1.894e-11 Underflow: 0 Overflow: 0	
LAT Configuration	LAT_CONFIG	q = LAT_CONFIG LAT_CONFIG Entries: 9.522997e+07 Mean: 0.002 RMS: 1.894e-11 Underflow: 0 Overflow: 0	
Data quality flag	DATA_QUAL	q = DATA_QUAL DATA_QUAL Entries: 9.527999e+07 Mean: 0.00192 RMS: 0.000228 Underflow: 0.0004256 Overflow: 7.347e-08	Need to check why there are these differences in a couple of runs. I suspect those are BAD runs, or runs with the new values for the DQM flag.

Attitude
quaternion

QSJ_1,
QSJ_2,
QSJ_3,
QSJ_4

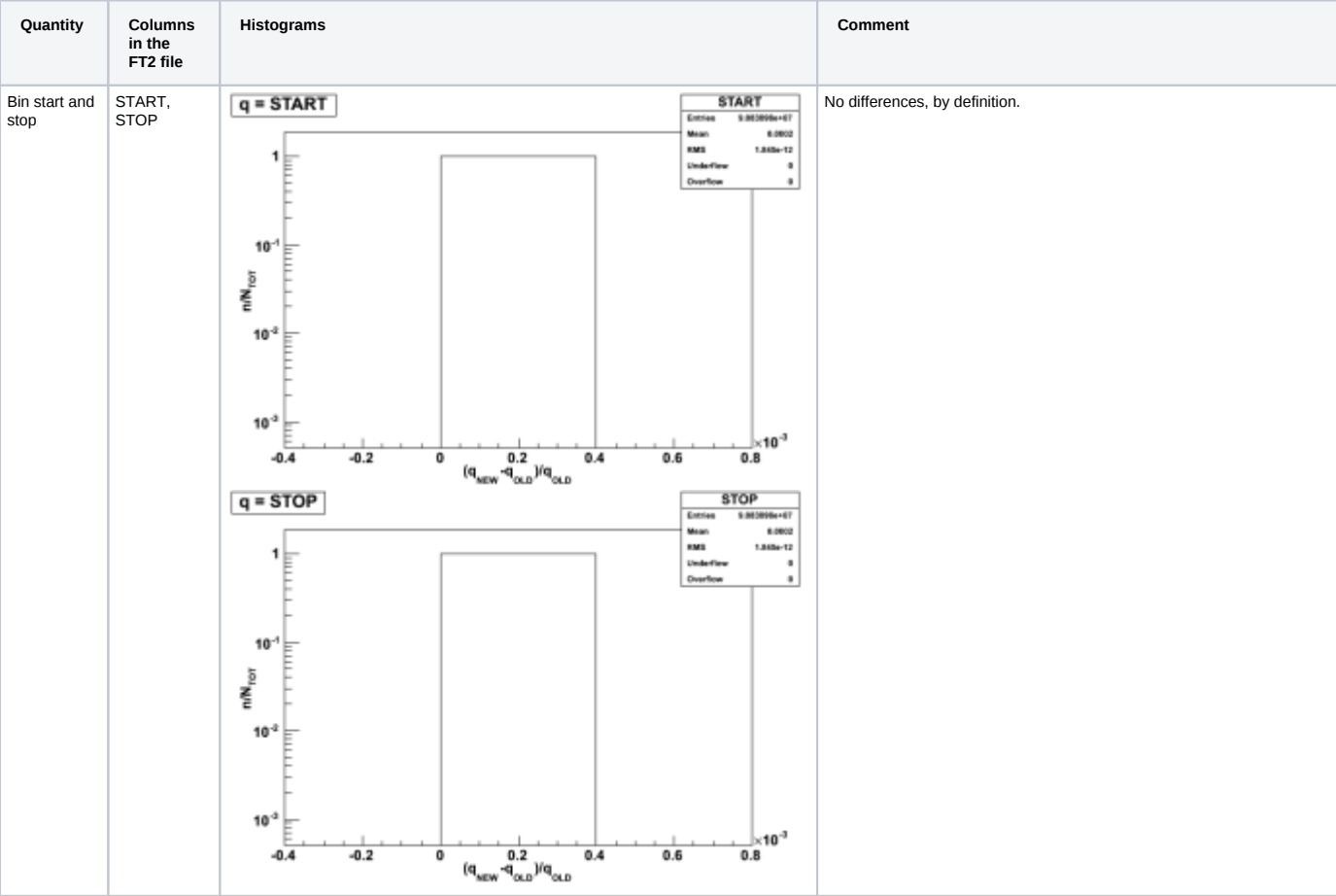


Reason 1.



Sub-interval

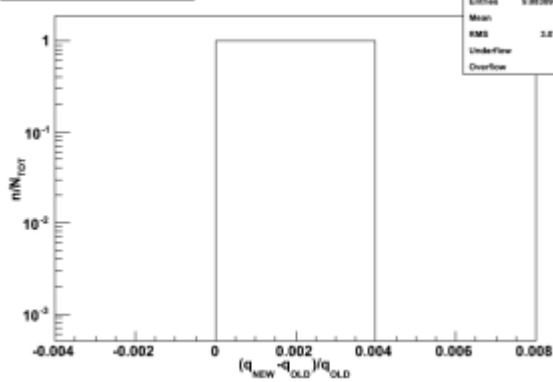
Here I repeated the same exercise, but excluding the first and the last 50 seconds from each file, to avoid the "border issue". Indeed, most of the differences are now gone:



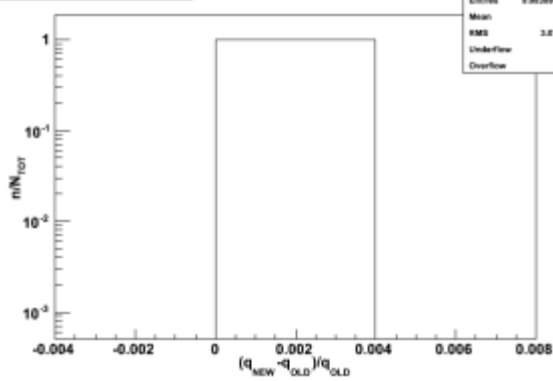
Spacecraft
position in
ECI
cooridantes
(x,y,z)

SC_POSITIO
N array

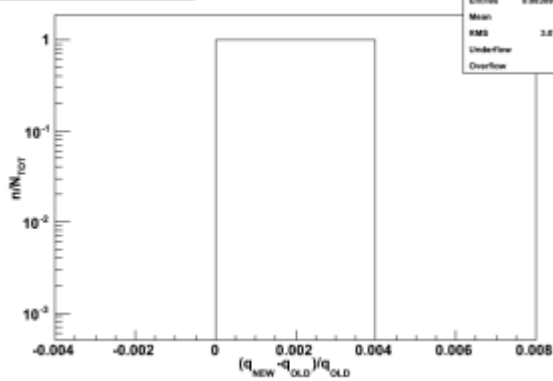
q = SC_POSITION_0



q = SC_POSITION_1

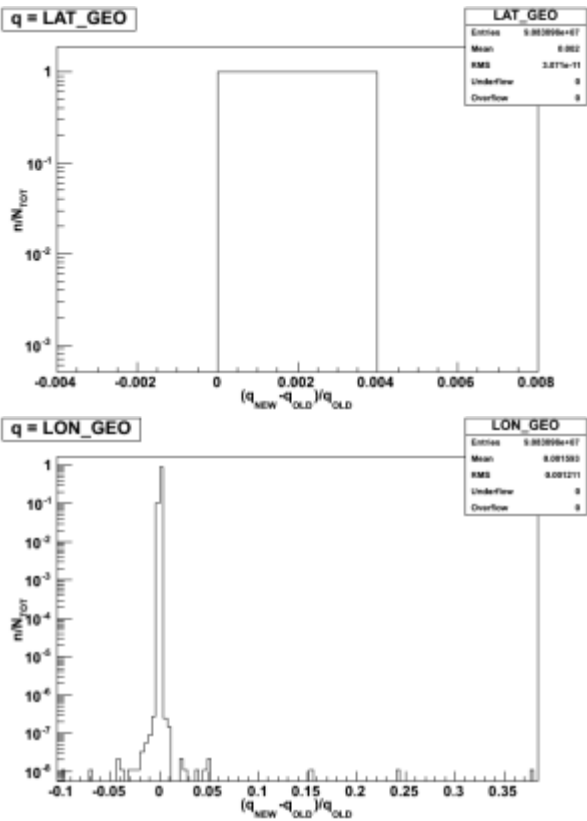


q = SC_POSITION_2



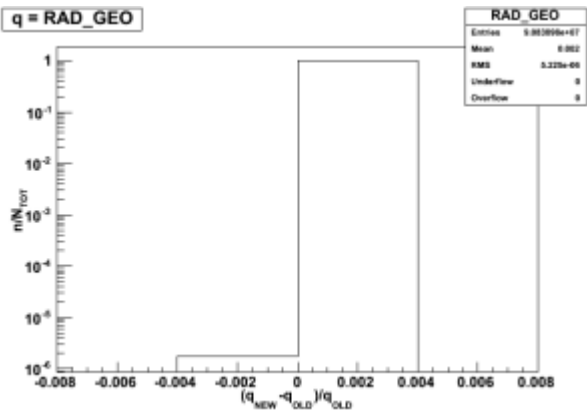
Ground
point latitude
and
longitude

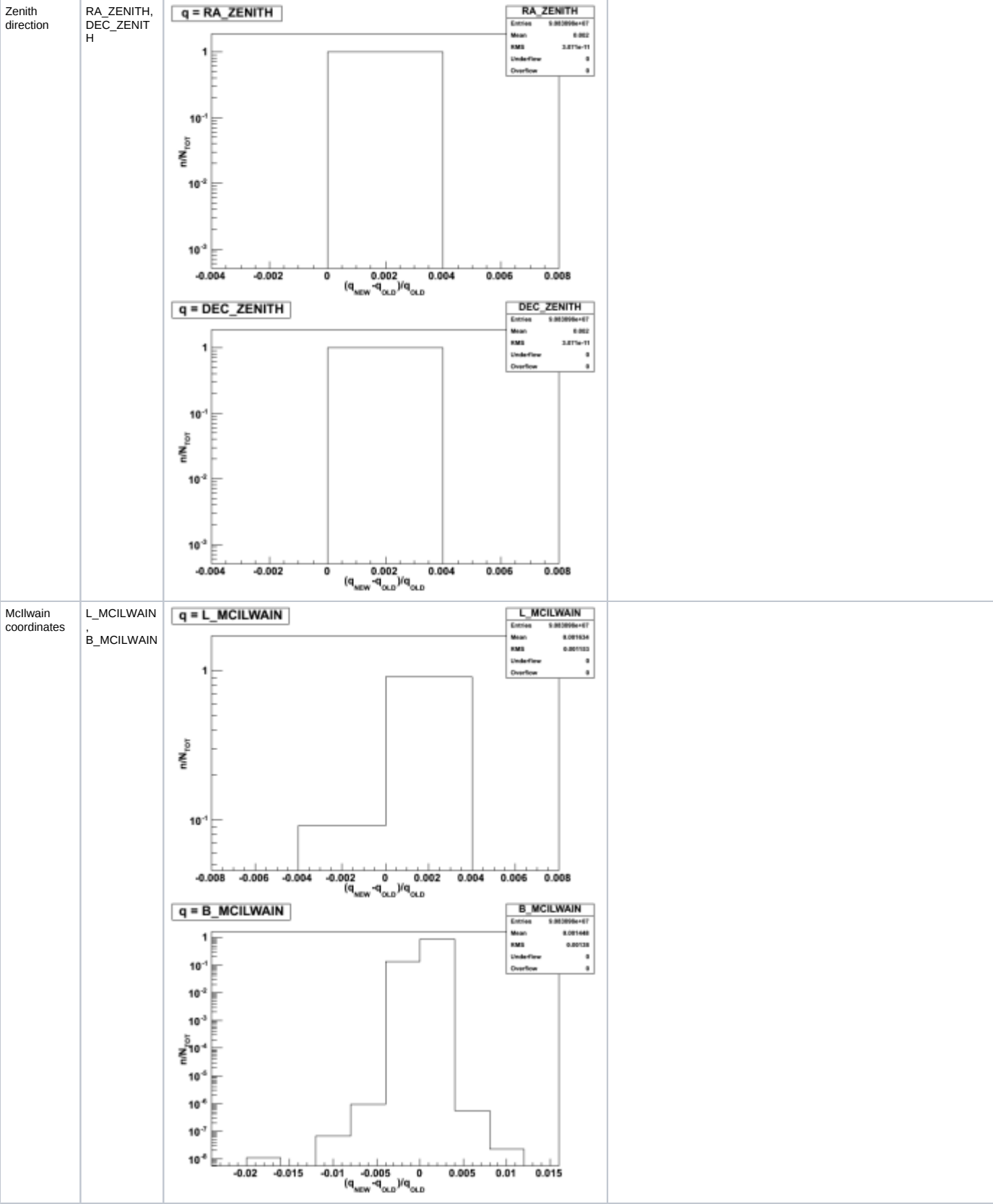
LAT_GEO,
LON_GEO



Spacecraft
altitude

RAD_GEO

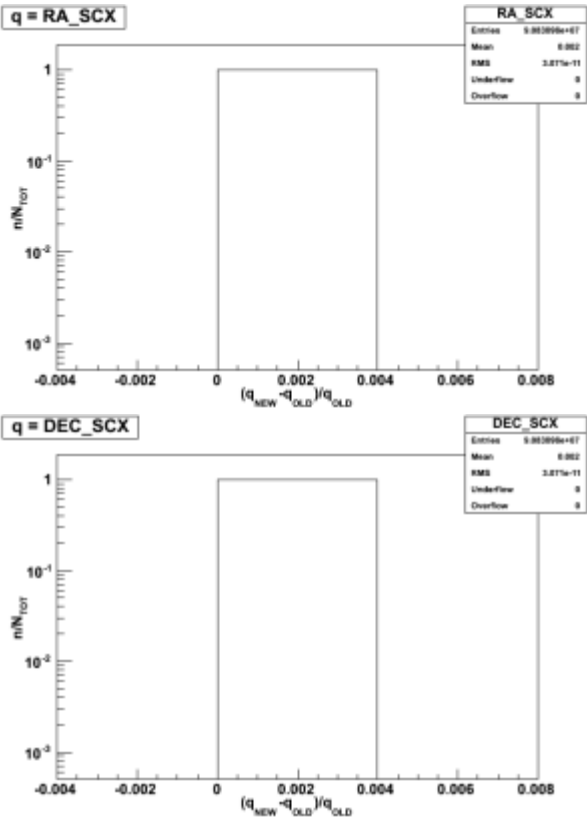




Geomagnetic latitude	GEOMAG_LAT AT	q = GEOMAG_LAT Statistics for GEOMAG_LAT: - Entries: 9.983899e+07 - Mean: 0.001789 - RMS: 0.0009472 - Underflow: 0.1823 - Overflow: 1.191e-08	16 percent of the time bins are in the underflow bin of the histogram. This is due to a known issue: on Feb. 2, 2010 we changed the quantity from unsigned to signed. Thus, all old FT2 files produced before that date have a always-positive GEOMAG_LAT, while the reprocessed ones have it signed from the beginning of the mission.
South Atlantic Anomaly flag	IN_SAA	q = IN_SAA Statistics for IN_SAA: - Entries: 9.983899e+07 - Mean: 0.002 - RMS: 3.871e-11 - Underflow: 0 - Overflow: 0	
Z-axis direction	RA_SCZ, DEC_SCZ	q = RA_SCZ Statistics for RA_SCZ: - Entries: 9.983899e+07 - Mean: 0.002 - RMS: 3.871e-11 - Underflow: 0 - Overflow: 0 q = DEC_SCZ Statistics for DEC_SCZ: - Entries: 9.983899e+07 - Mean: 0.002 - RMS: 3.871e-11 - Underflow: 0 - Overflow: 0	

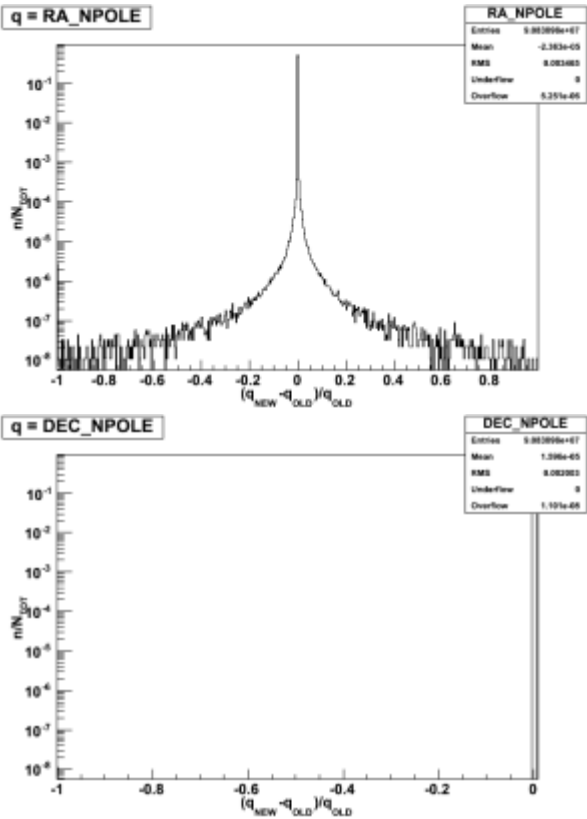
X-axis
direction

RA_SCX,
DEC_SCX



Orbital Pole
direction

RA_NPOLE,
DEC_NPOLE

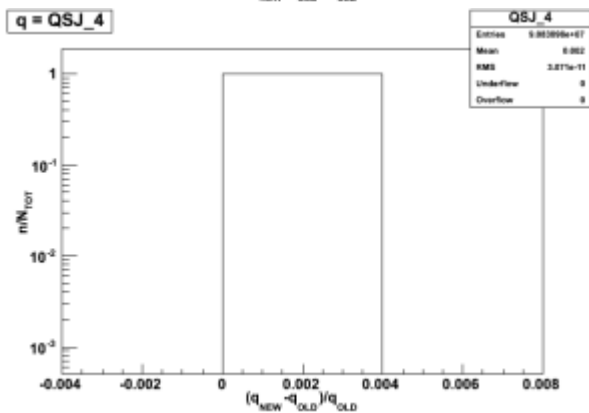
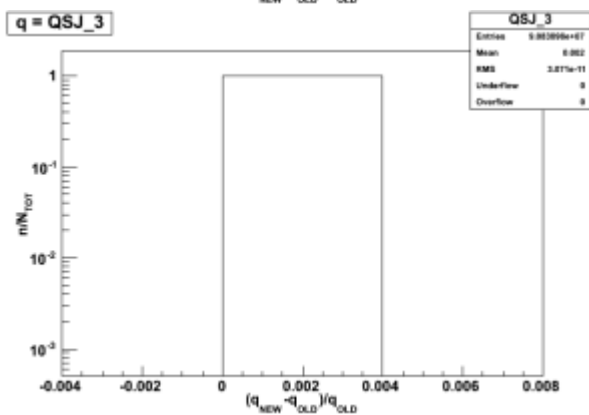
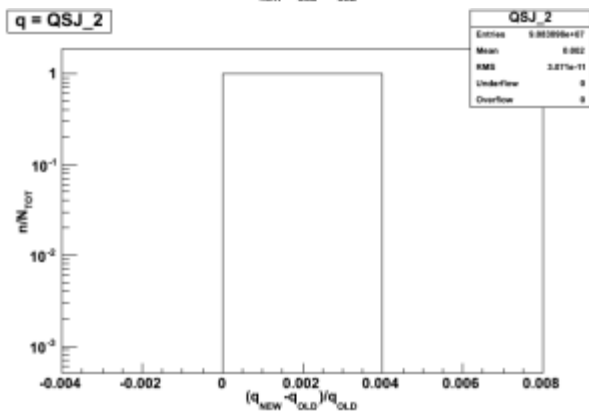
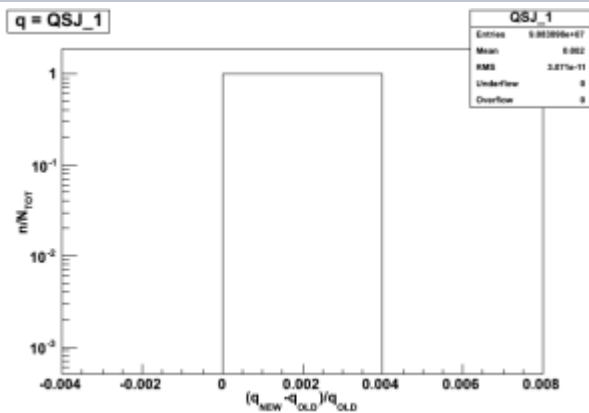


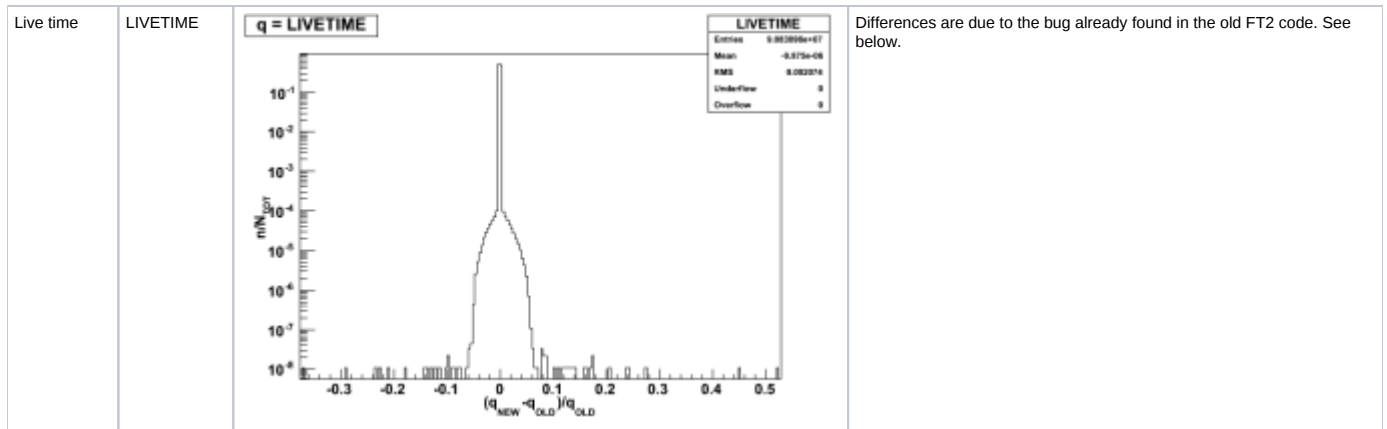
Here there are considerable differences, because the new code compute the coordinates of the Orbital Pole in a different way respect to the old one. If v_i is the position vector for the i -th bin in the FT2 file, the old code computed the orbital pole vector for the i -th bin as the cross product between v_{i-1} and v_i , while the new code uses the product v_i cross v_{i+1} . Thus basically I should compare each i -th entry in the new set with the $(i+1)$ th entry in the old one, to obtain the same results. I did this test, and everything is ok, see below.

Rocking angle	ROCK_ANGLE E	q = ROCK_ANGLE ROCK_ANGLE Entries: 9.983898e+07 Mean: 0.002 RMS: 4.197e-07 Underflow: 0 Overflow: 0	
LAT GNC mode	LAT_MODE	q = LAT_MODE LAT_MODE Entries: 9.983898e+07 Mean: 0.002 RMS: 3.871e-11 Underflow: 0 Overflow: 0	
LAT Configuration	LAT_CONFIG	q = LAT_CONFIG LAT_CONFIG Entries: 9.983898e+07 Mean: 0.002 RMS: 3.871e-11 Underflow: 0 Overflow: 0	
Data quality flag	DATA_QUAL	q = DATA_QUAL DATA_QUAL Entries: 9.983898e+07 Mean: 0.001991 RMS: 0.000073 Underflow: 0.0004421 Overflow: 7.708e-48	Need to check why there are these differences in a couple of runs. I suspect those are BAD runs, or runs with the new values for the DQM flag.

Attitude
quaternion

Q SJ_1,
Q SJ_2,
Q SJ_3,
Q SJ_4

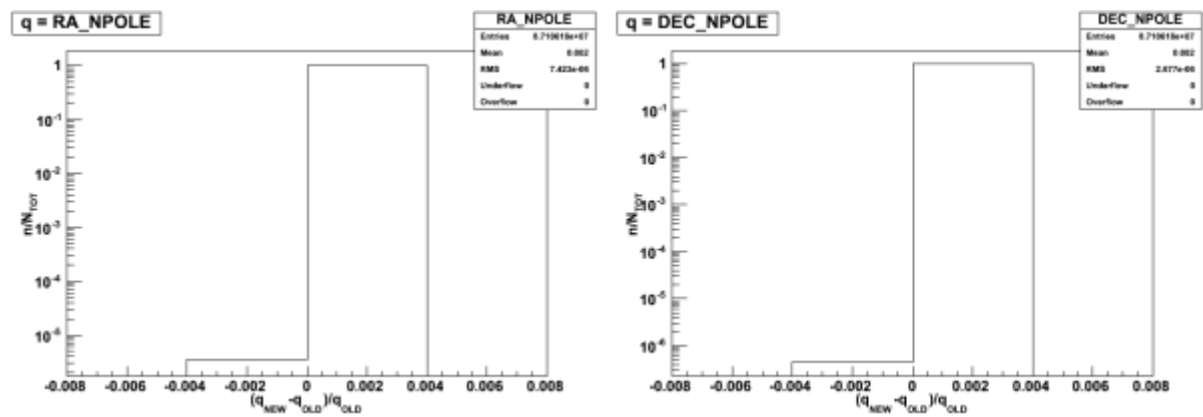




There are still some differences in some quantities related to the EarthCoordinate class of the astro package. By looking at the time histories, all the differences are before a certain date, thus they are almost certainly related to a bug-fix in the astro package. (to be confirmed)

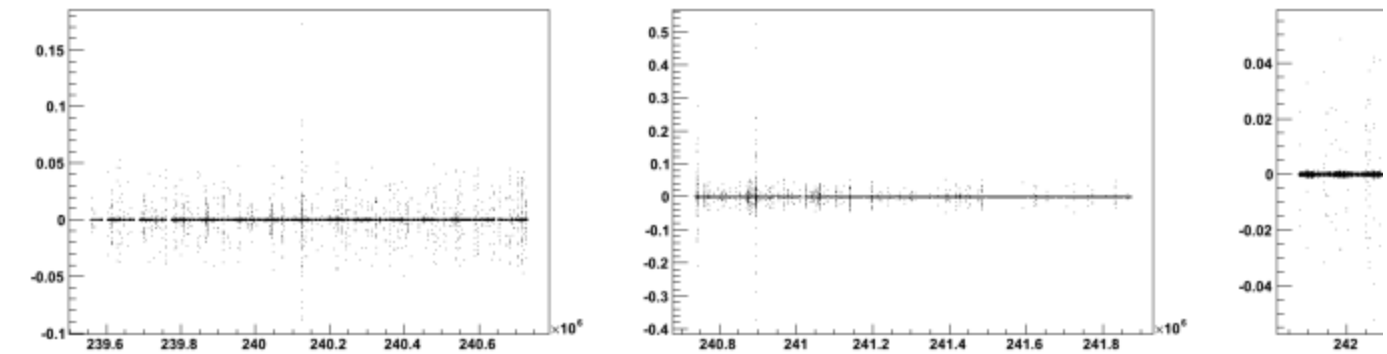
Orbital Pole

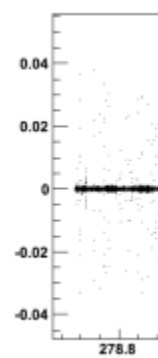
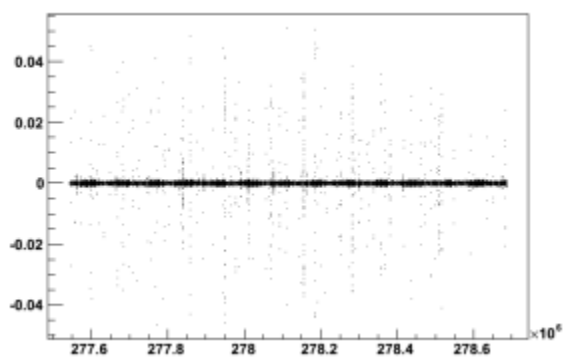
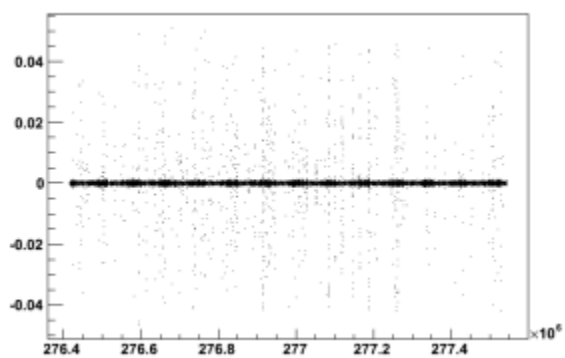
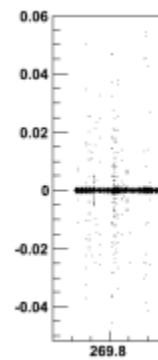
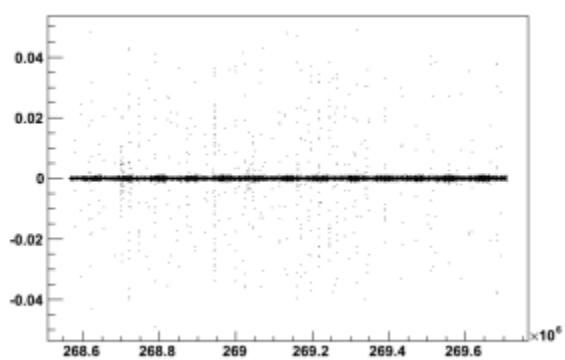
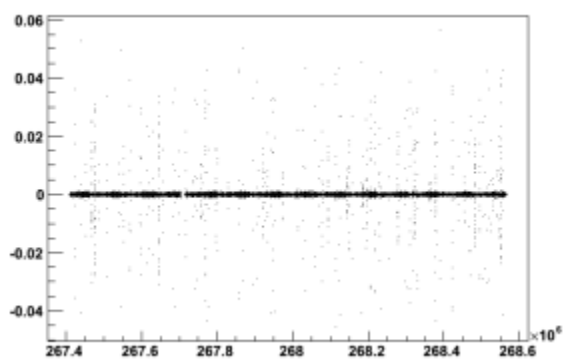
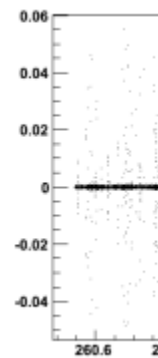
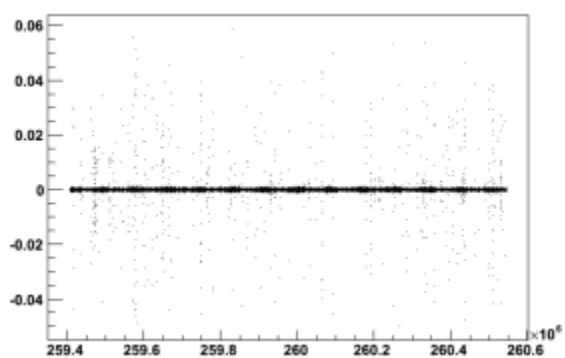
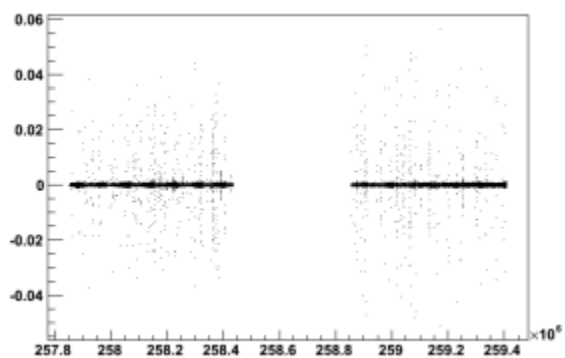
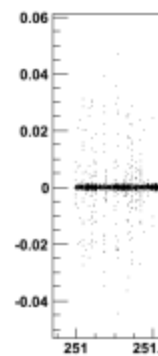
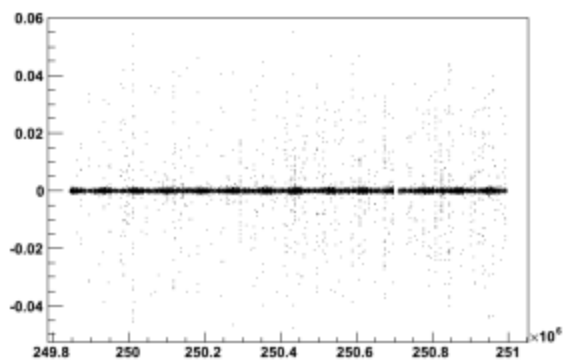
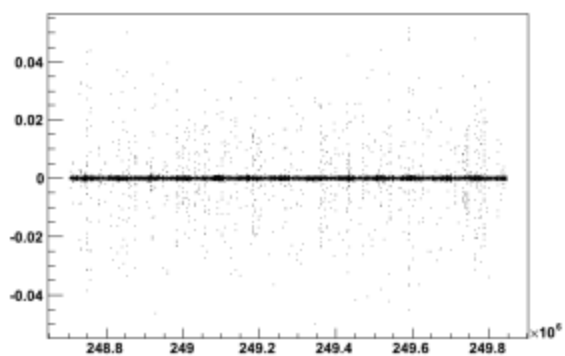
I repeated the comparison between the RA_NPOLE and DEC_NPOLE quantities, but comparing a given i-th bin of the old version of the FT2 files with the i-th-1 bin in the new FT2 file, to solve the discrepancy in the computation method. As expected, the differences vanish:

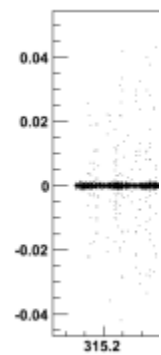
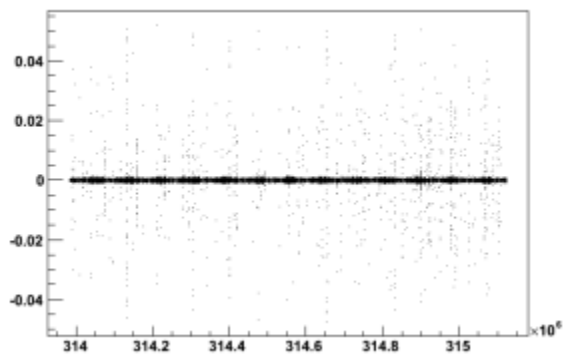
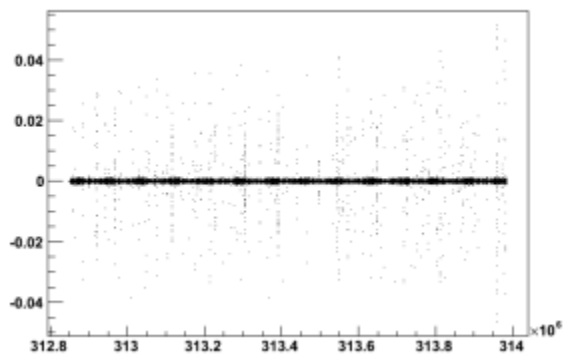
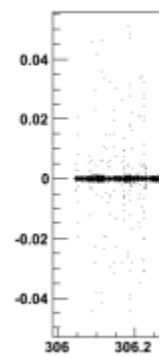
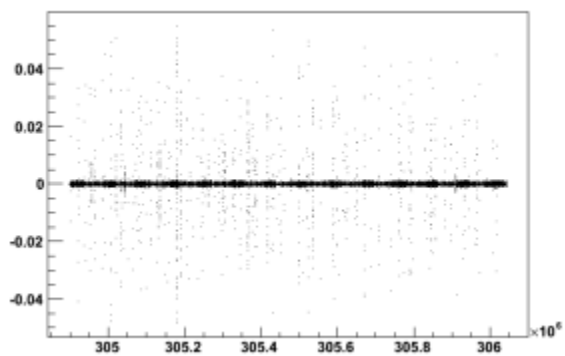
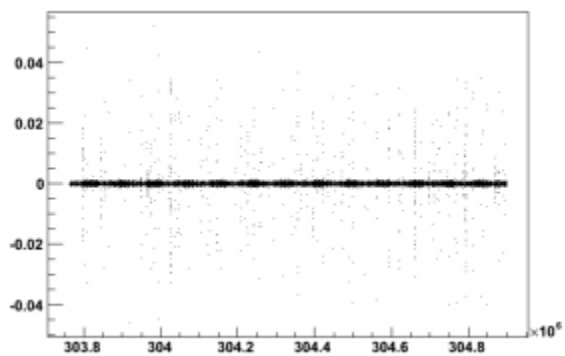
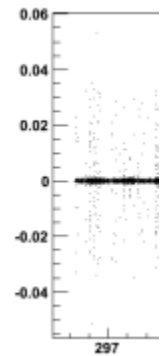
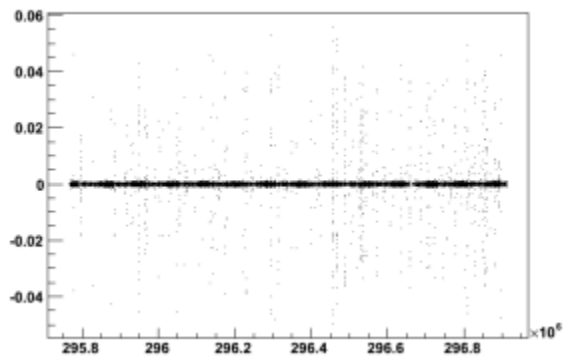
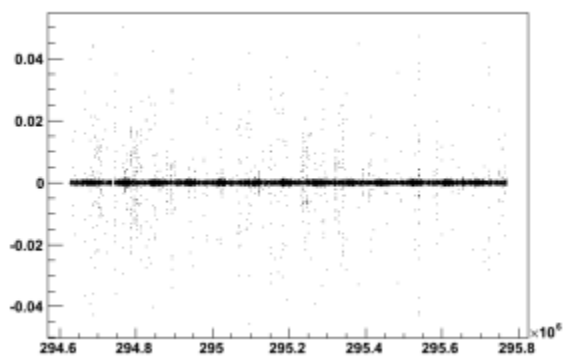
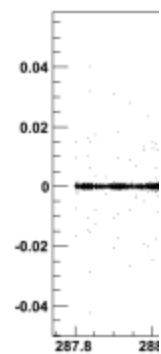
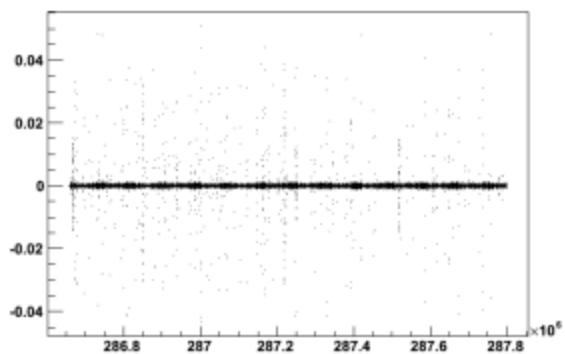
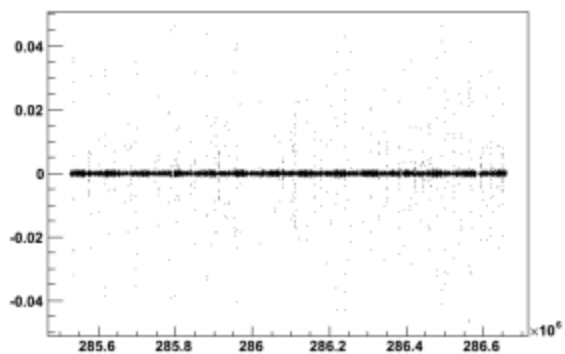


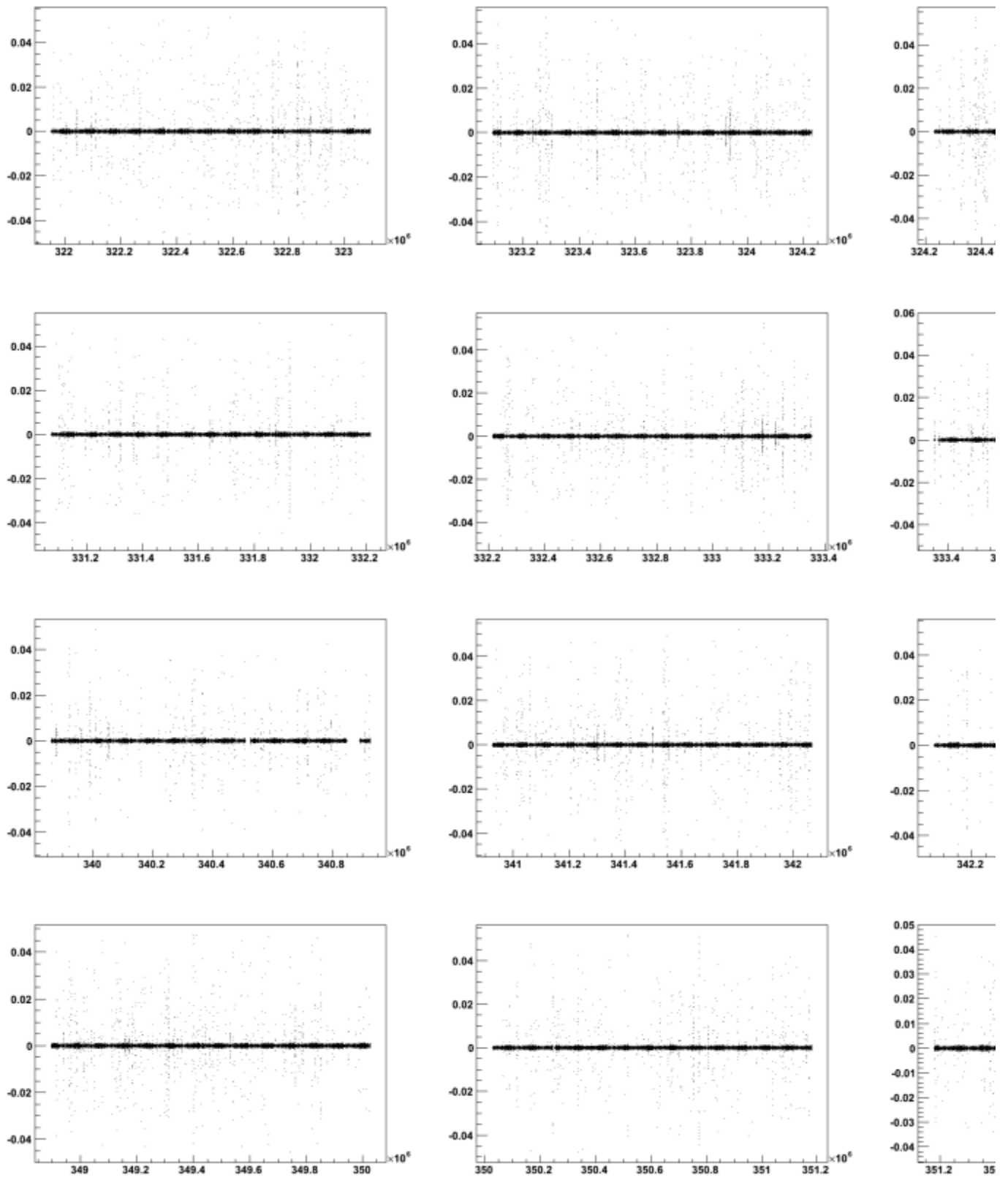
Livetime

As already found in my former test (see the mother page), the old code had a bug in the livetime computation. Here I show the difference in livetime between the two codes, as function of time, thus on the y-axis there is the usual fractional difference, while on the x-axis there is the MET time (in seconds):
(text continue below the gallery, click on an image to enlarge it)



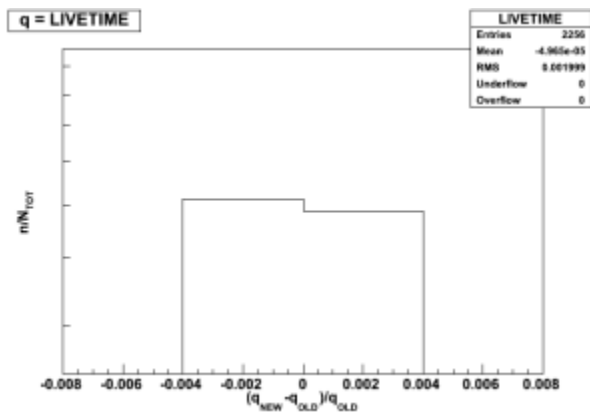






As you can see, differences are distributed around zero, with maximum differences of the order of few percent most of the time, except for some exceptional cases. Thus, they should cancel out when merging the 1s bins in the 30s ones.

Indeed, this is the histogram of the fractional difference in LIVETIME for 30s FT2 files:



Note that there is a rather small number of entries in this histogram. This is because the new code changed the binning scheme for the 30s FT2 file, thus the old and the new sets of 30s file are often not comparable on a bin-by-bin basis. Indeed, start and stop times of the bins are most of the time different between the two sets. I do not know how to solve this issue... Interpolating is not an option, obviously.