HGCAL ToA studies

Towards ToA linearization and performance measurements

Beatriz Ribeiro Lopes

27 March 2024 - HGCAL DPG

ToA in HGCAL

Why?

- HGCAL is designed to have precision timing capabilities
- Information of Time-of-Arrival (ToA) of the pulses expected to improve CMS event reconstruction

  ToA compatibility between energy deposits is a powerful handle to suppress PU, and can also be used to identify main collision vertex

- **Expected performance:** ~20 ps timing resolution for deposits > 100 fC

  Precise calibration needed to achieve such performance

ToA in HGCAL

Calibration

- Non-linear dependence between ”true” ToA and the measured one

  We need to understand the non-linearity and correct for it

- Two different effects:

  - **non-uniformity of ToA codes:** ToA signals go through TDC with 1024 intervals. These intervals are not all the same duration, e.g. the code between 510 and 511 can correspond to a longer time than the code between 2 and 3

  - ***timewalk:*** Hits corresponding to lower charges are detected later than hits with higher charge deposit

ToA in HGCAL

Calibration

- Non-linear dependence between ”true” ToA and the measured one

  We need to understand the non-linearity and correct for it

- Two different effects:

our focus today

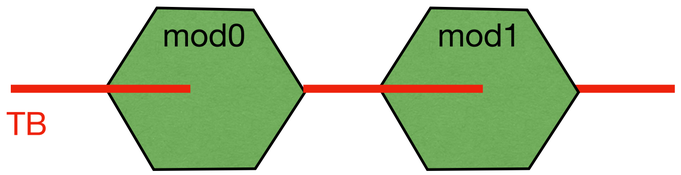
- **non-uniformity of ToA codes:** ToA signals go through TDC with 1024 intervals. These intervals are not all the same duration, e.g. the code between 510 and 511 can correspond to a longer time than the code between 2 and 3

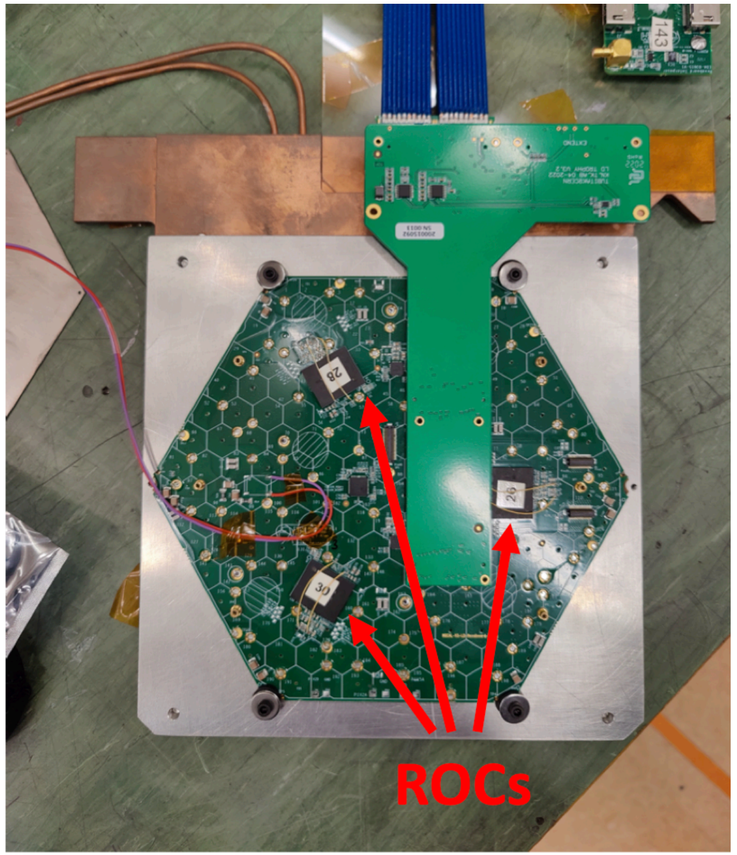
- ***timewalk:*** Hits corresponding to lower charges are detected later than hits with higher charge deposit

to be done, for previous results see e.g. Loukas’ talk

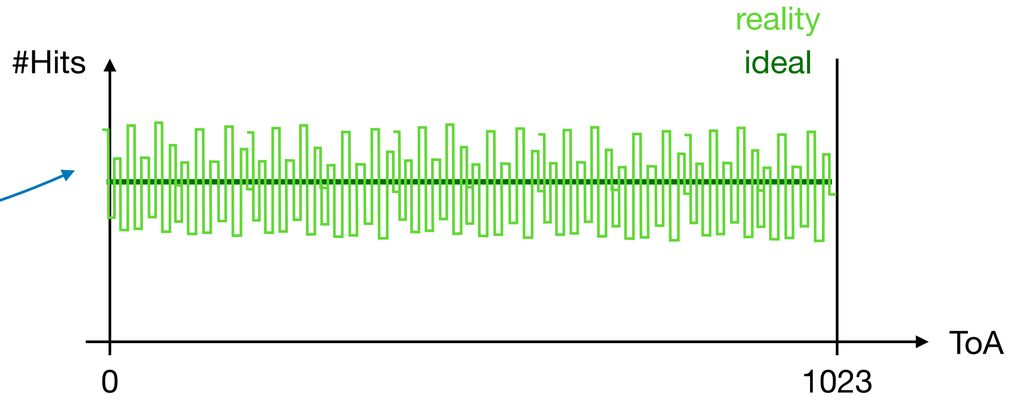
ToA in HGCAL

Method for calibration





- Use data collected during the Aug/Sep 2023 Test beam at CERN-SPS
  - 2 modules with 3 ROCs each
- To correct the non-uniform length of ToA codes, new data was taken with the same modules (no beam), with the asyncronous clock generator (ACG)
- The ACG will result in a uniform time signal
  - If all ToA codes had the same duration, we would get a uniform distribution, but they do not



The difference between real and uniform can be used as correction

ToA in HGCAL

ACG runs of March 15

Four runs taken with ACG signal every four channels

Data are saved in: `/vault/HgcalLabtest2023/`
Processed NANOAOD in: `/eos/cms/store/group/dpg_hgcal/tb_hgcal/2023/labtest/pro/`

Using only one of the two modules for now (216* channels in total)

Relay1710508811 : channels with ToA / ACG data : 0, 4, 8, 12 ...

Relay1710509013 : channels with ToA / ACG data : 1, 5, 9, 13 ...

Relay1710509063 : channels with ToA / ACG data : 2, 6, 10, 14 ...

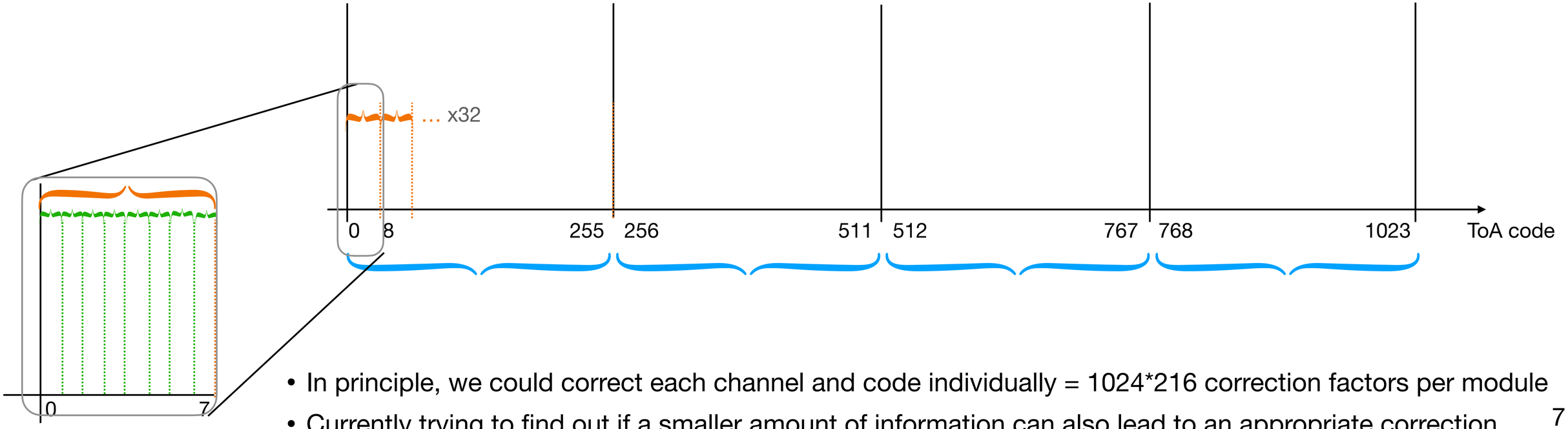
Relay1710509126 : channels with ToA / ACG data : 3, 7, 11, 15 …

*(222-6). Channel 18 in each ROC half is always skipped because it is a special channel for calibration

Analysis of ACG data

Strategy for calibration

- TDC has an internal structure
- Structure stems from each TDC being composed of 3 stages:
  - 2-bit grey counter LSB ≈ 6.25 ns
  - Coarse 5-bit TDC LSB ≈ 200 ps
  - Fine 3-bit TDC LSB ≈ 25 ps



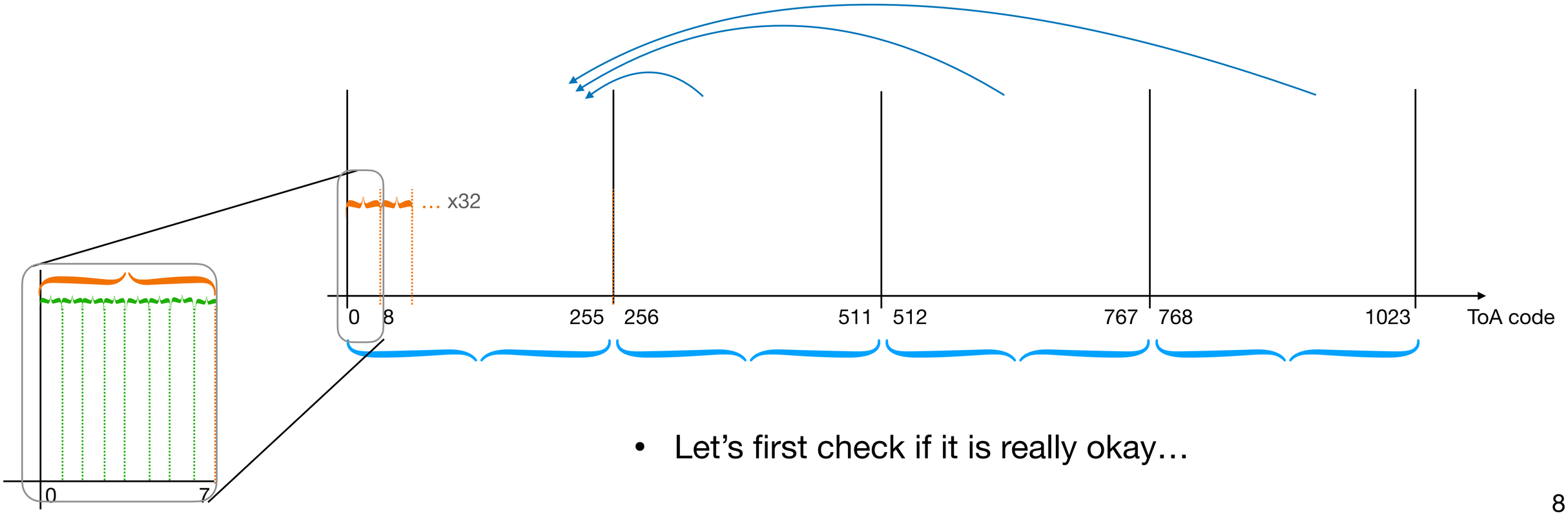
- In principle, we could correct each channel and code individually = 1024*216 correction factors per module
- Currently trying to find out if a smaller amount of information can also lead to an appropriate correction

Analysis of ACG data

Strategy for calibration

- In principle we can fold the first two bits for the calibration

ToA code

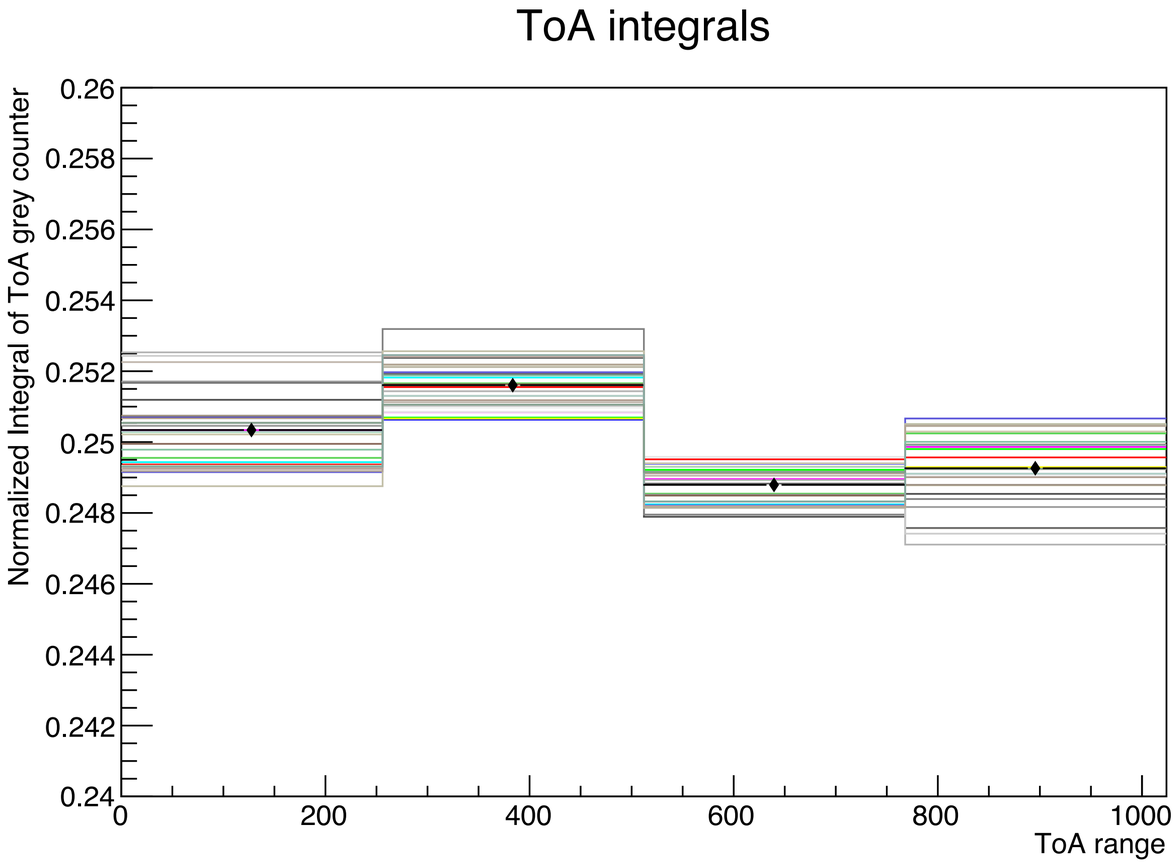


- Let’s first check if it is really okay…

Analysis of ACG data

Integral of each grey counter

- Compare the normalized integrals of the 4 grey counter bins, for all runs



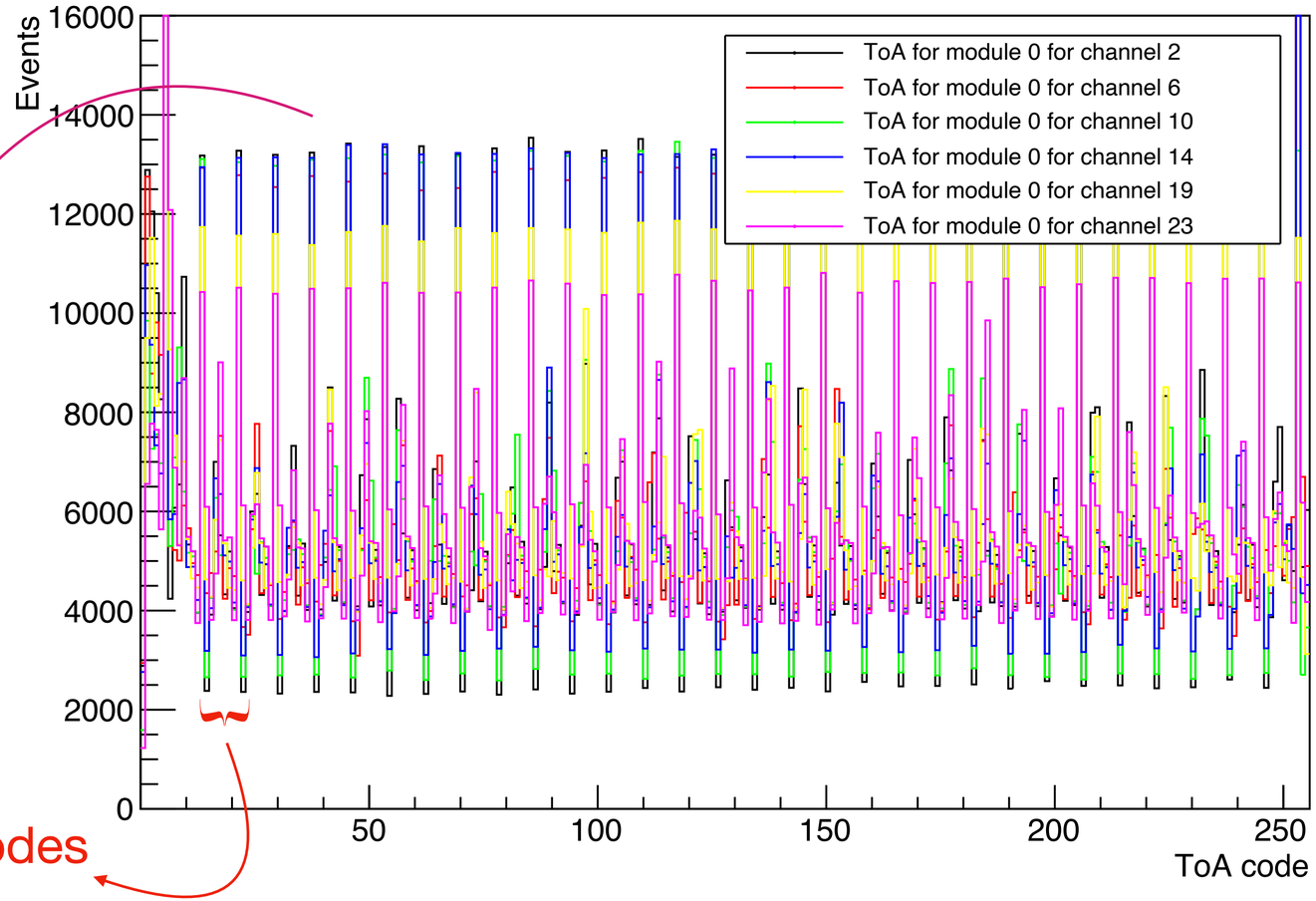
- They are the same within half %, so we do not consider any correction for now and simply fold these 2 bits (1024 codes → 256 codes)

Analysis of ACG data

Calibration per code for the coarse+fine TDC

- Fold first 2 bits and look at range 1-255 (excluding ToA 0)
- Results shown for one ACG run (but are similar for all runs)
- Histogram of ToA codes clearly reflects internal structure of TDCs

ToA per-channel for module 0 for Run1710509063



smaller variations between "blocks" of 8 codes

Clear pattern every 8 codes

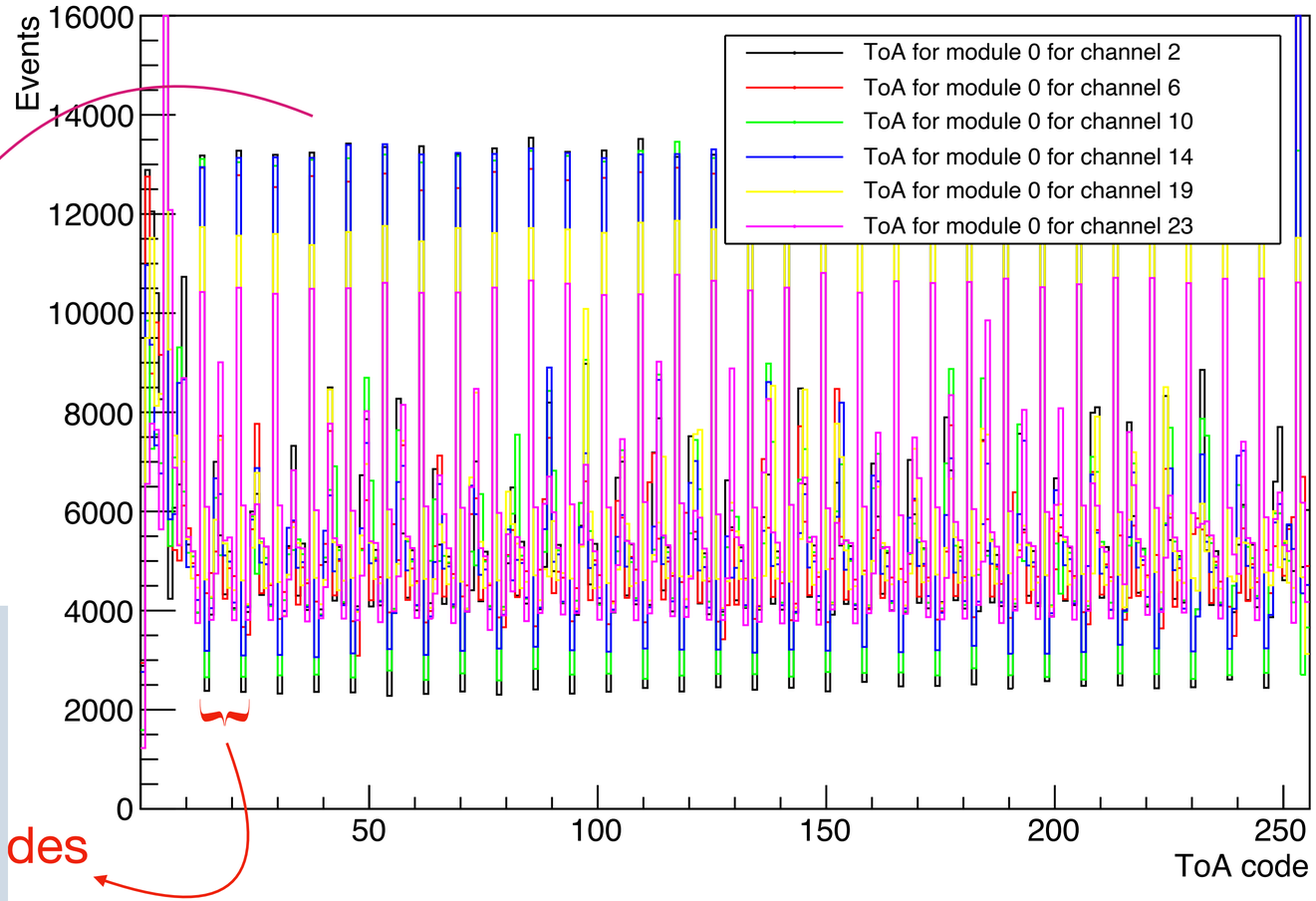
In the ACG runs there was trimming of the ToA analogue parameters in the ROC

Analysis of ACG data

Calibration per code for the coarse+fine TDC

- Fold first 2 bits and look at range 1-255 (excluding ToA 0)
- Results shown for one ACG run (but are similar for all runs)
- Histogram of ToA codes clearly reflects internal structure of TDCs

ToA per-channel for module 0 for Run1710509063



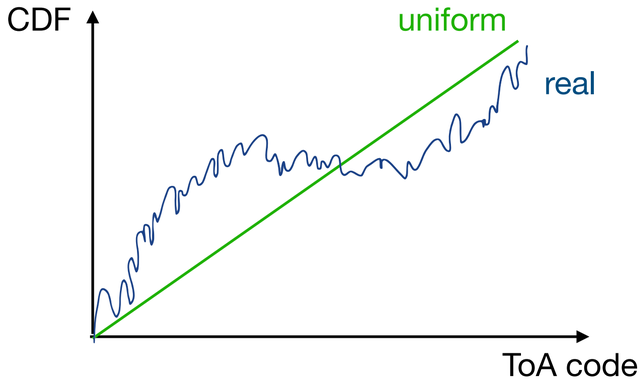
smaller variations between

des

Strategy: compare cumulative distributions with a straight diagonal line (uniform dist.) to derive corrections

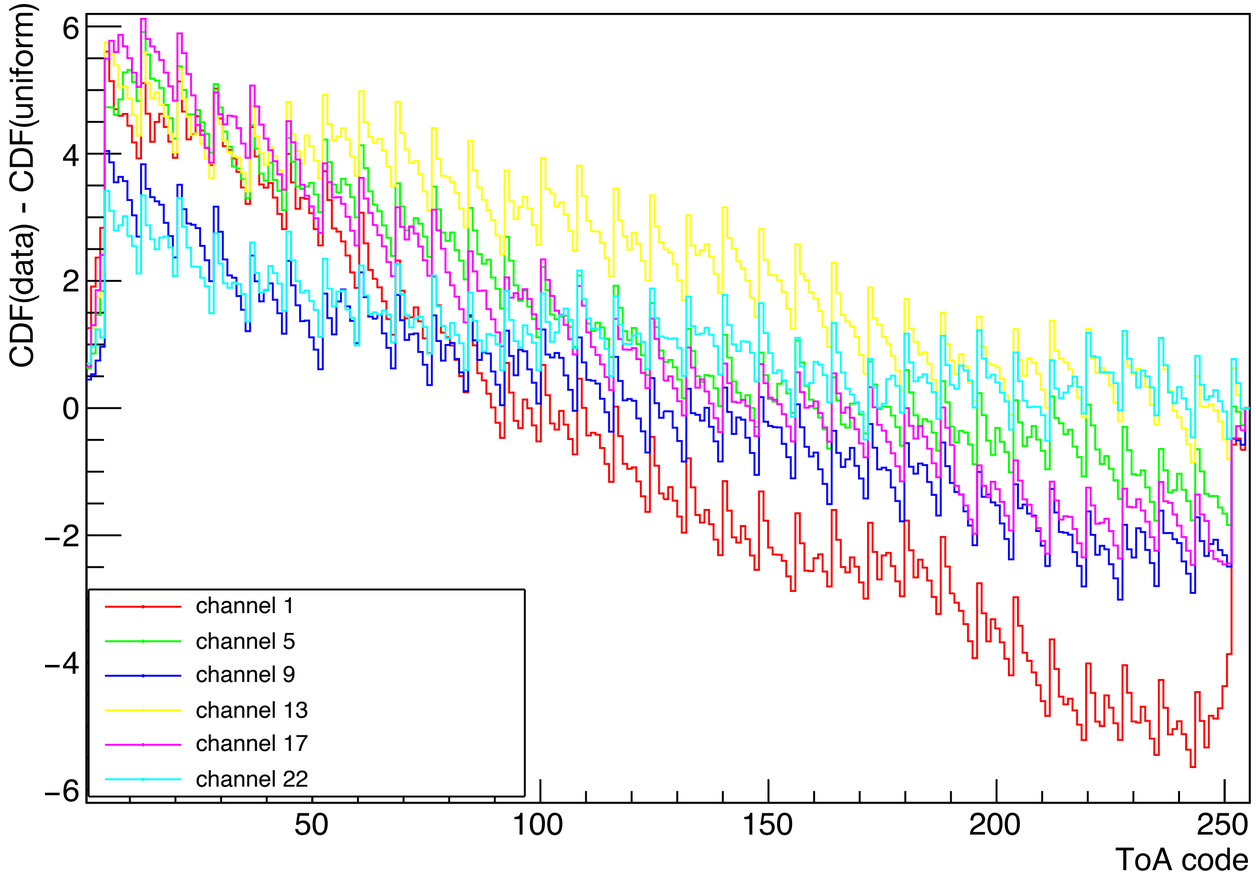
Analysis of ACG data

Calibration per code for the coarse+fine TDC



- Take difference between real and uniform distributions after normalising
- Start by correcting fine TDC

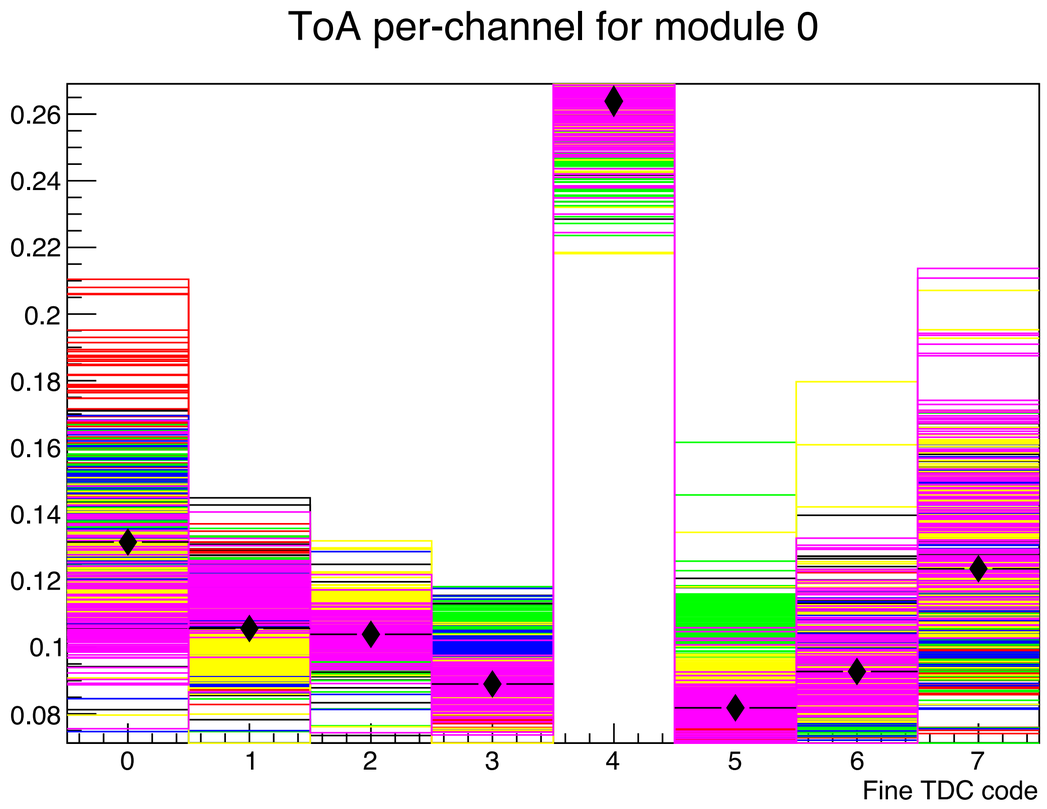
Run1710509013 CDF residuals



Analysis of ACG data

Calibration per code for the fine TDC

Derive per-code scale factors to correct for non-uniform codes in fine TDC:



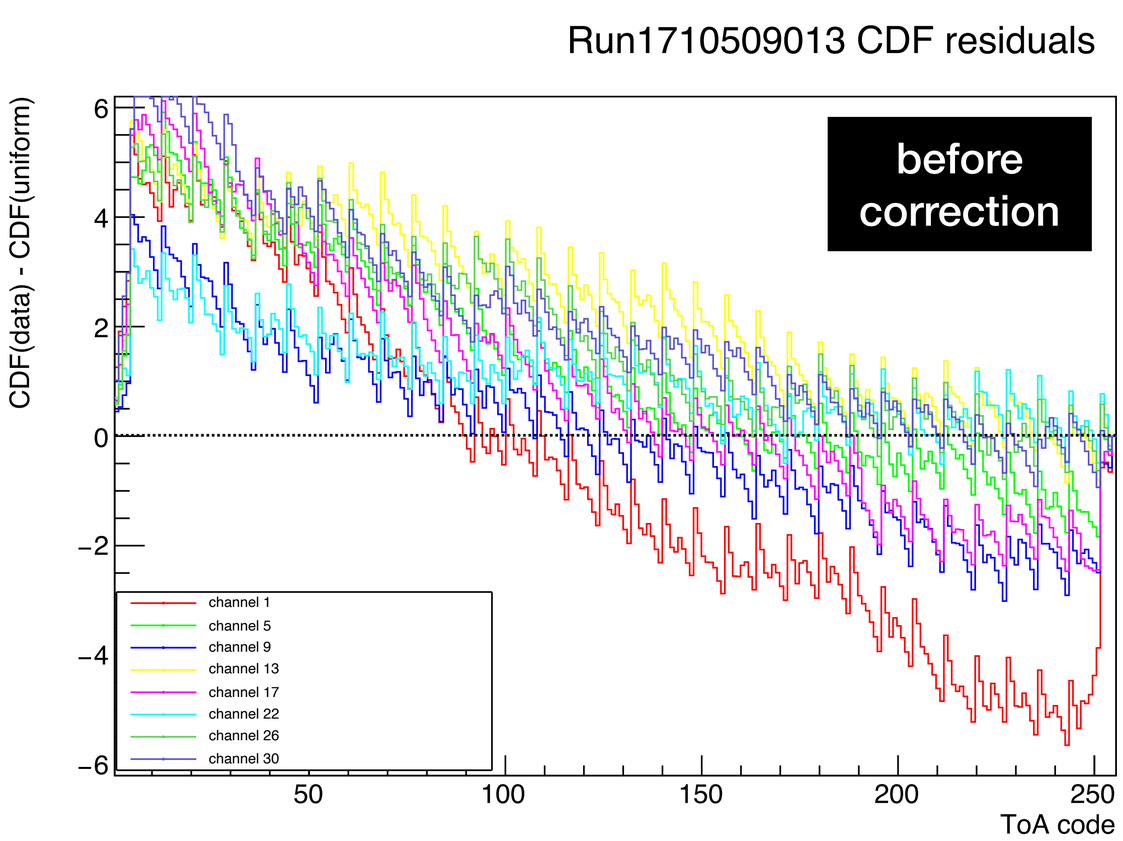

```
{
  "0": {
    "value": 0.9506682280430561,
    "err": 0.0012785005667492546
  },
  "1": {
    "value": 1.1841595447915,
    "err": 0.006956921536609901
  },
  "2": {
    "value": 1.204002979984159,
    "err": 0.003688156348012916
  },
  "3": {
    "value": 1.4058760263472387,
    "err": 0.0036777311282724574
  },
```

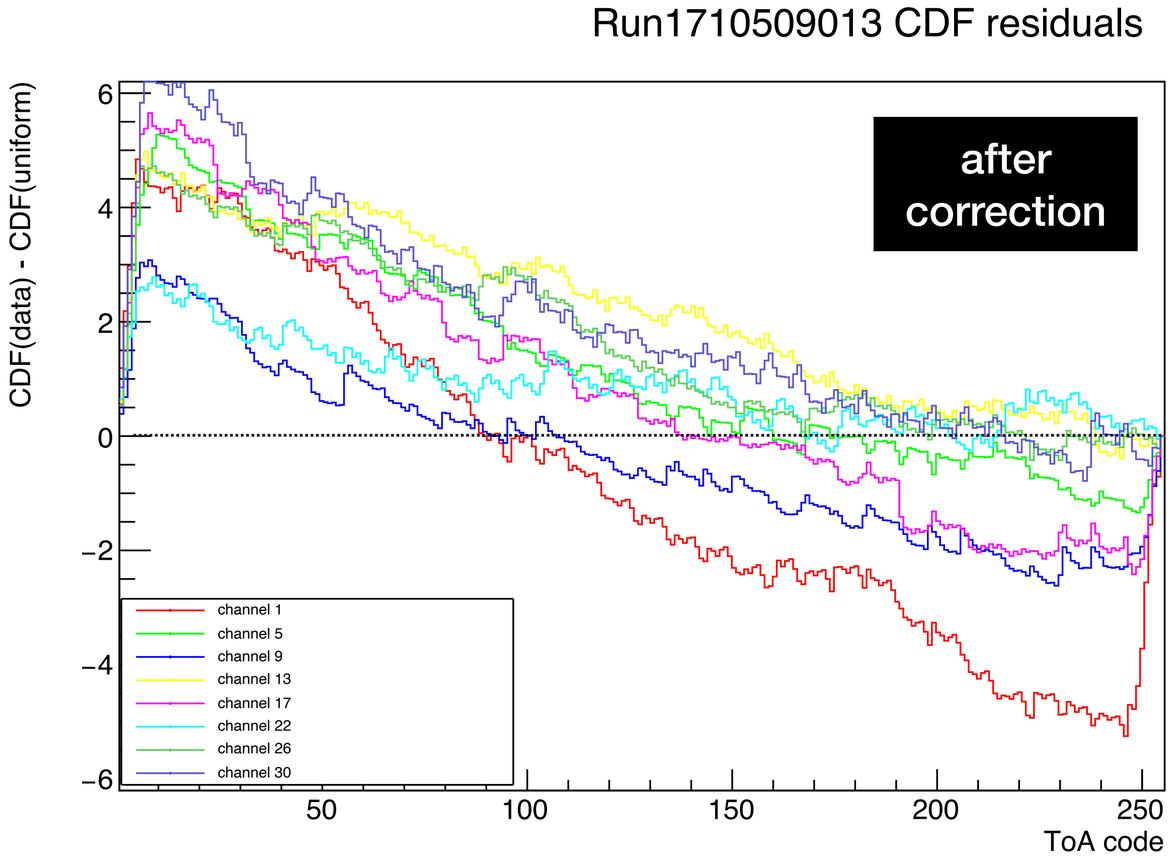
```
  "4": {
    "value": 0.4590597145208566,
    "err": 0.0010323804479778602
  },
  "5": {
    "value": 1.5286052535981178,
    "err": 0.0035685452867524254
  },
  "6": {
    "value": 1.3489258565328772,
    "err": 0.024038188764744405
  },
  "7": {
    "value": 1.0122718118197434,
    "err": 0.0013555041094243806
  }
}
```

Analysis of ACG data

Calibration per code for the fine TDC

- Check that it works by applying back to ACG data ✓

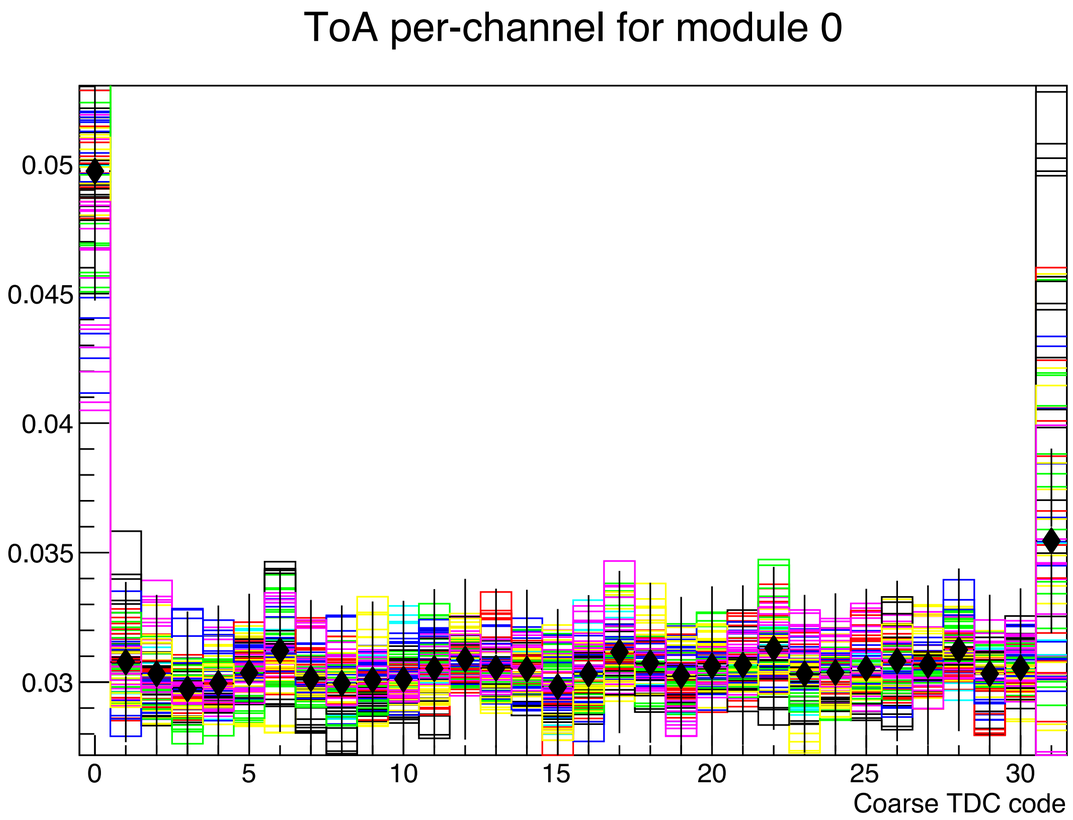




Analysis of ACG data

Calibration per code for the coarse TDC

- Now do the same for CTDC:



Not so clear in this case that there is a pattern, except for first and last codes


```
"0": {
  "value": 0.6284059987316678,
  "err": 0.06315718142977927
},
"1": {
  "value": 1.0154431702173028,
  "err": 0.10205587735788903
},
"2": {
  "value": 1.0306312306943424,
  "err": 0.10358233693433021
},
```

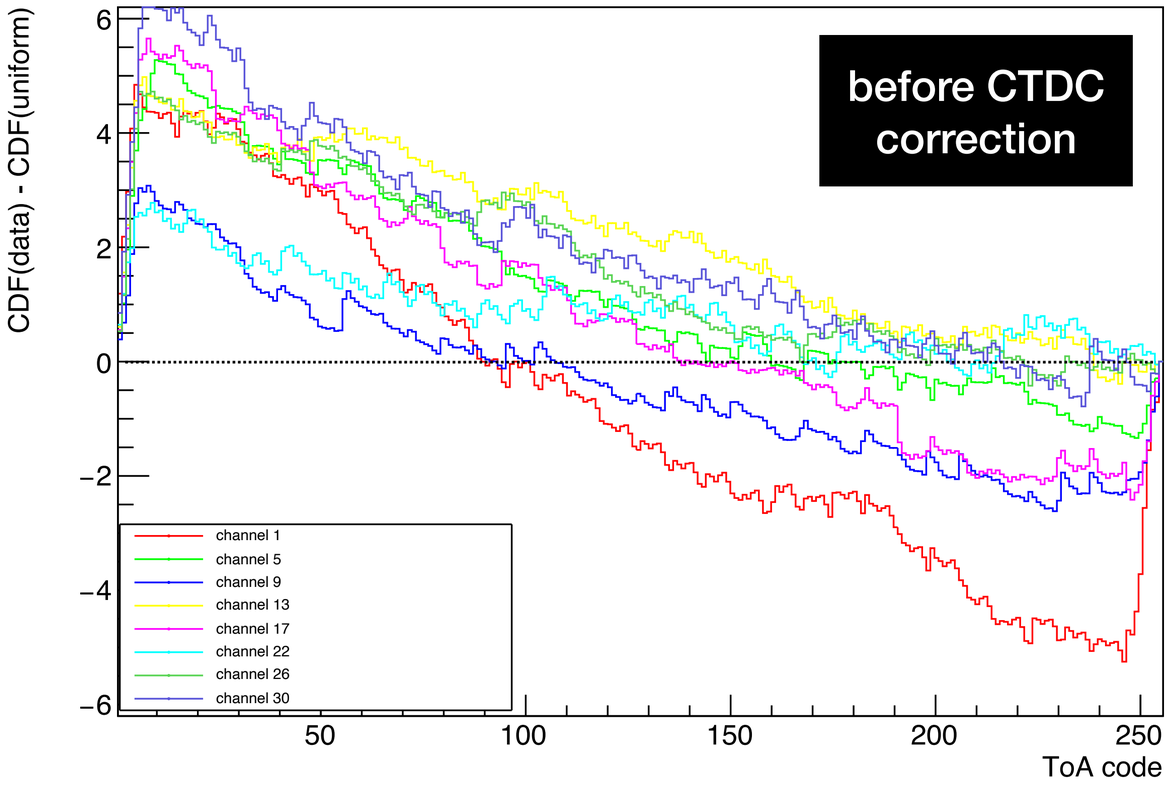

…


```
"29": {
  "value": 1.0305036093113036,
  "err": 0.10356950859405181
},
"30": {
  "value": 1.0228725571131039,
  "err": 0.10280255812865718
},
"31": {
  "value": 0.8814218417507501,
  "err": 0.08858622584193203
}
}
```

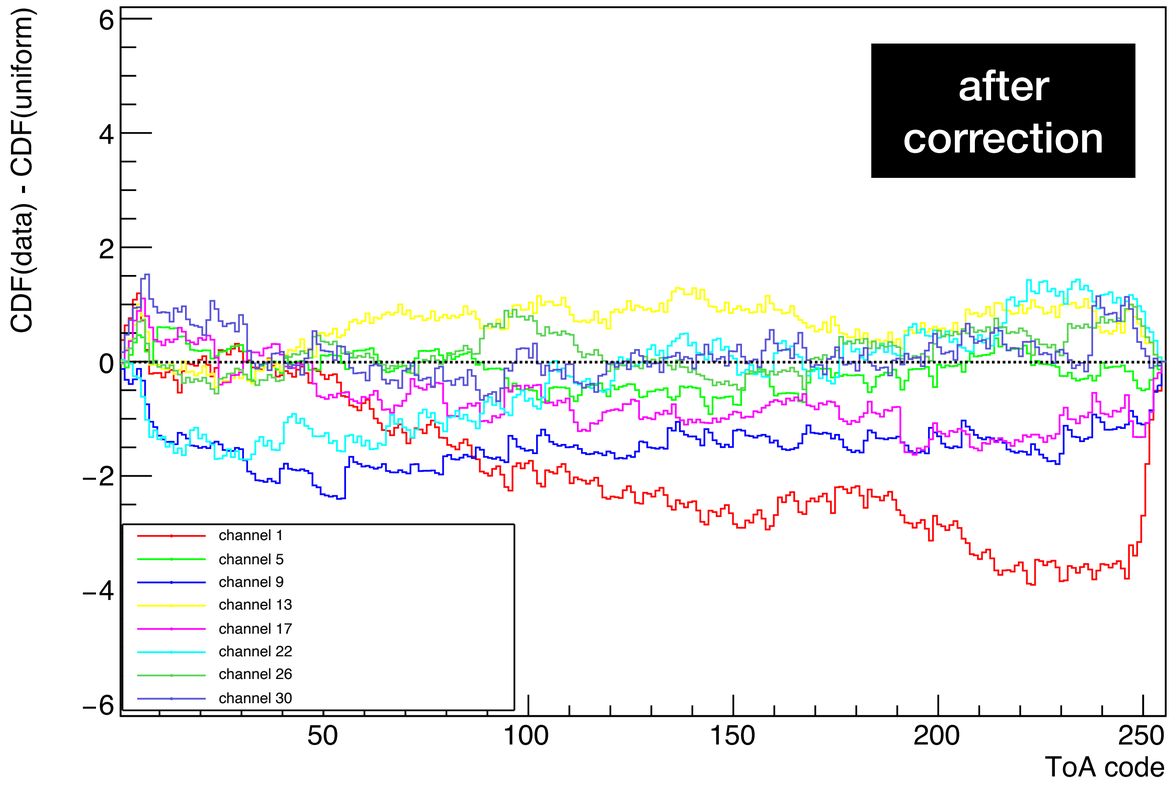
Analysis of ACG data

Calibration per code for the coarse TDC

Run1710509013 CDF residuals



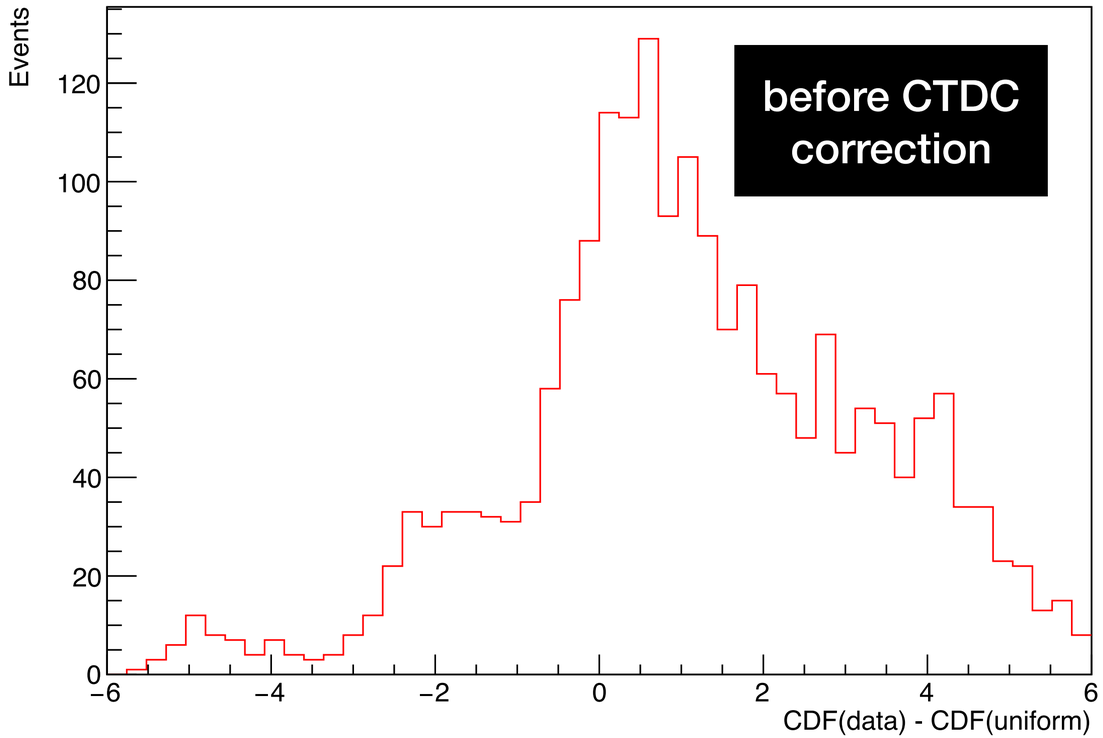
Run1710509013 CDF residuals

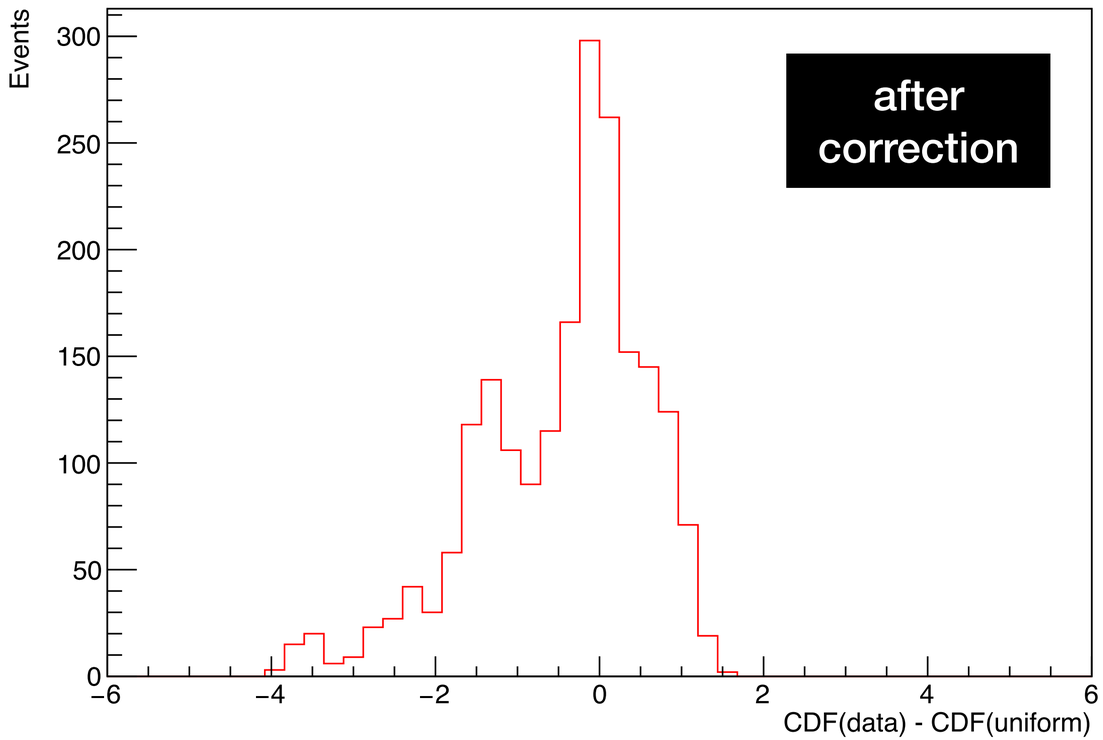


Check that it works by applying back to ACG data ✓ not perfect, but slope is clearly reduced

Analysis of ACG data

Calibration per code for the coarse TDC





- Looking at the histogram of the residuals, the improvement is clear

Summary and next steps

- Need to calibrate ToA response of HGCAL modules
- Important step is to correct for non-uniform length of ToA codes in TDC
  - Calibrating using data from ACG runs
  - Derived first correction factors to account for internal structure of FTDC and CTDC

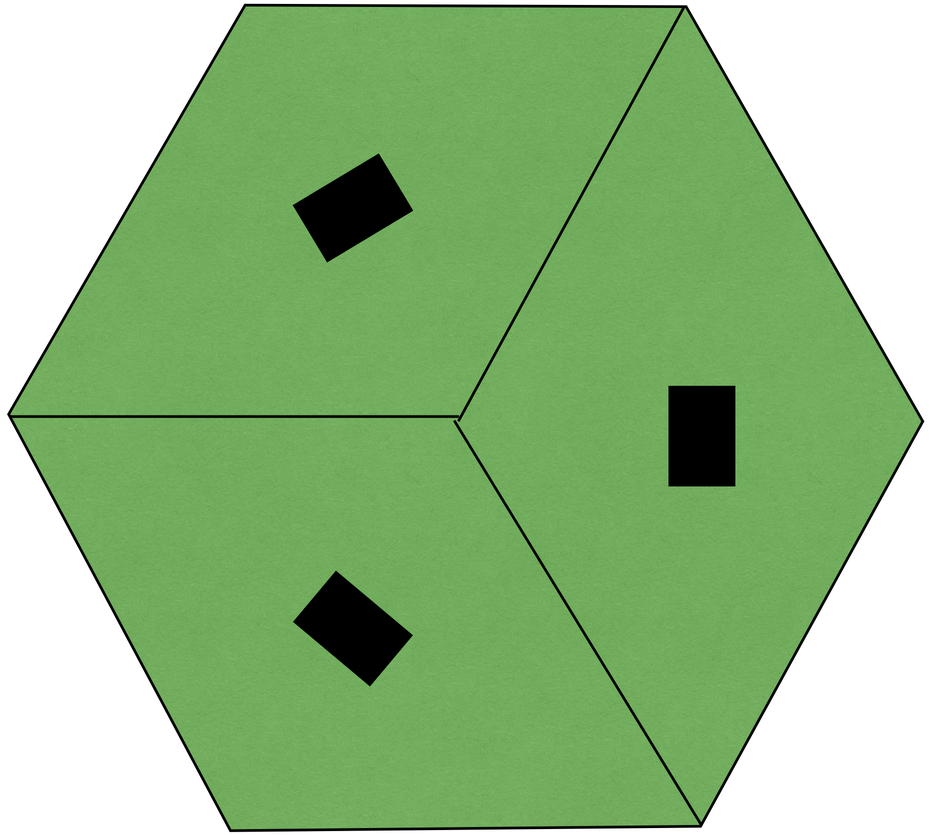
- **Next**:
  - Apply corrections to test beam data and compare calibrated and uncalibrated results
  - Continue with *timewalk* corrections

Thank you for your attention!

Backup

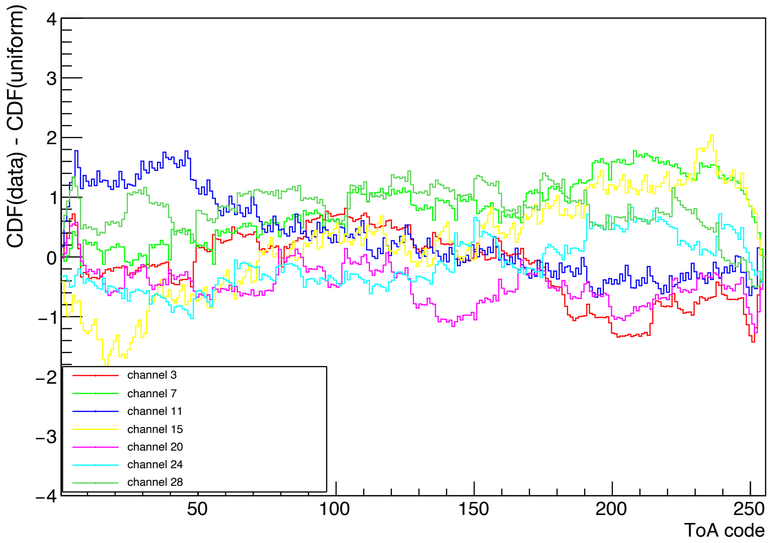
Structure of the modules

- Per module, 222 channels
  - 3 ROCs
  - 2 halves
  - 37 channels per half

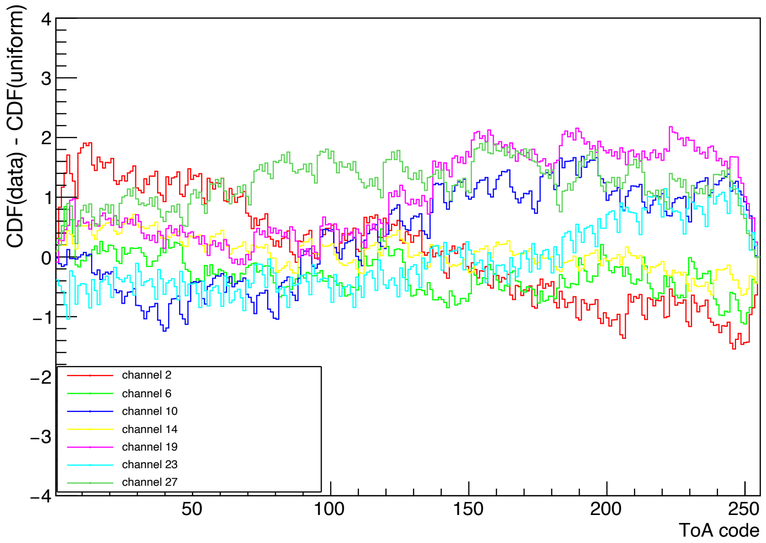


Corrected results for all ACG runs

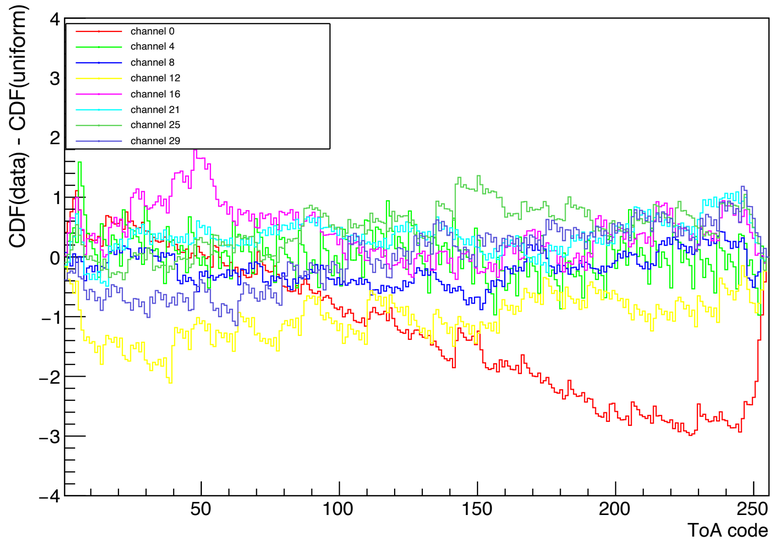
Run1710509126 CDF residuals



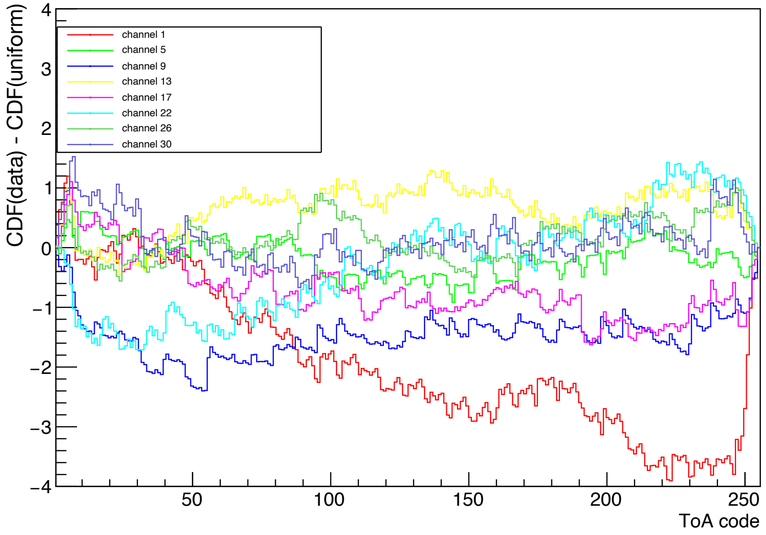
Run1710509063 CDF residuals



Run1710508811 CDF residuals

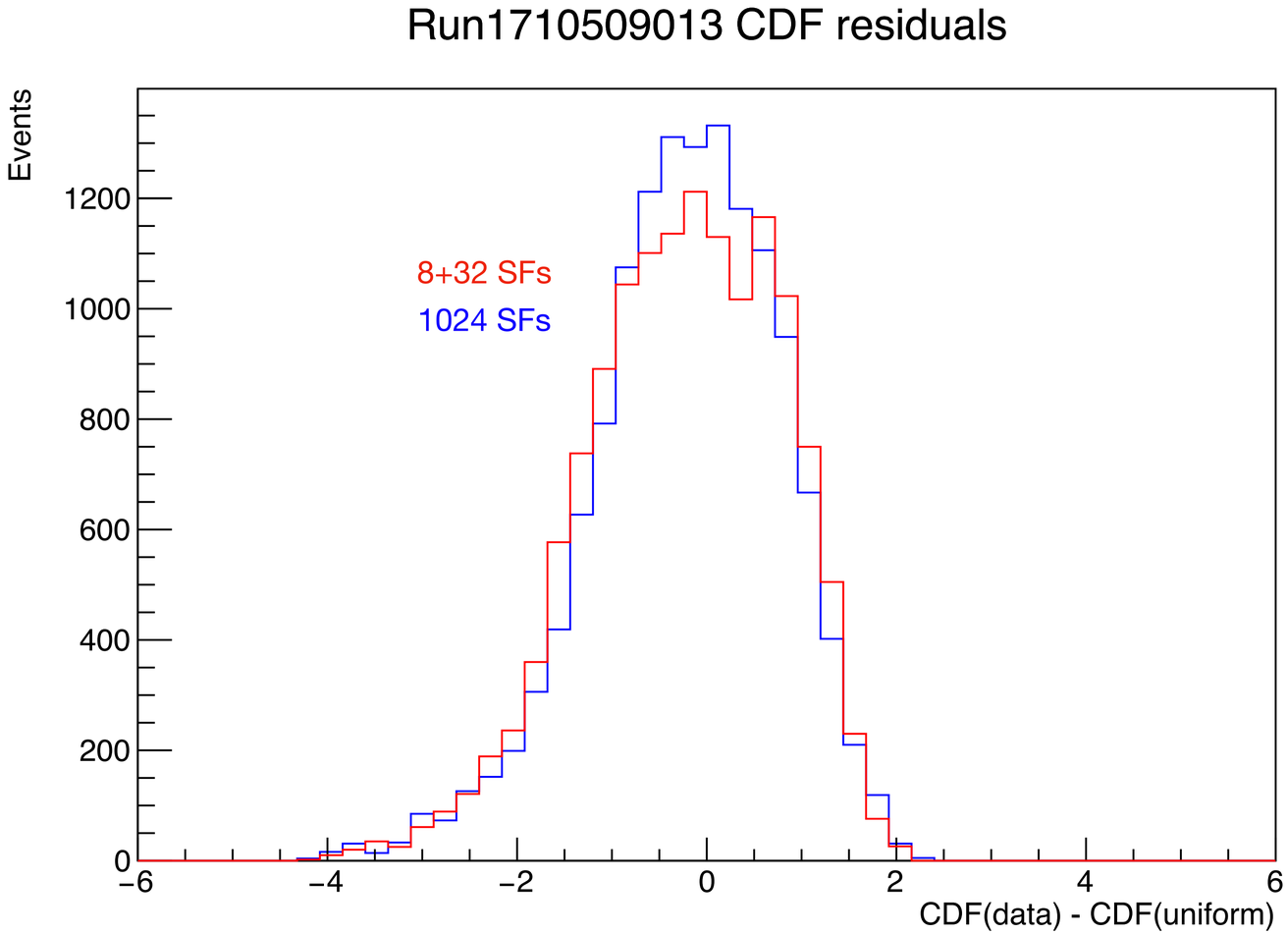


Run1710509013 CDF residuals



What happens if we simply apply 1024 corrections (one per ToA code)?

- Still averaging per channel



- Slightly better, but very small difference, the bigger effect that remains uncorrected is the difference between channels