Documenting HMS calorimeter coding in ENGINE (an outline)

Vardan Tadevosyan

General organization

The description is broken into few parts:

1. Construction of the calorimeter (in short)
2. Input parameter files
3. Common blocks
4. The code flow
    a) main subroutines
    b) debugging subroutines
    c) calibration subroutine

A short description of the counter, related to data analysis :

- location (at the back of hut);
- sizes (10cmx10cmx70cm);
- arrangement of blocks (4 layers, 13 rows);
- two types of modules (1 PMT, 2PMT);
- number of channels (52 positive, 26 negative).

Input Parameter Files

Two input parameter files:

hcal.pos: contains primary geometric parameters (positions and thicknesses of the layers, number of blocks in the layers, etc);

hcal.param: contains the "slop" parameter (for track-cluster association), calorimeter fiducial volume test flag (for tracking), initial and current gains, calibration constants, etc.

For each file, parameters are listed along with their meanings.

Parameter files for HMS calorimeter.
In ENGINE, the primary geometry of the calorimeter is described in replay/PARAM/hcal.pos file. The CTP parameteres there are:
hcal_num_neg_columns - number of front layers with negative PMTs, nominally 2;
hcal_1pr_zpos - Z position of the front layer (Preshower);
hcal_<i>ta_zpos - Z postions of the tail layers, i=2,3,4;
hcal_1pr_thick - thickness of the first layer (10 cm);
hcal_<i>ta_thick - thicknesses of the tail layers, i=2,3,4 (10cm); ... … …

Common Blocks

In /INCLUDE/hms_calorimeter.cmn:

hms_cal_parms: primary geometric parameters;
hms_geometry_cal: secondary geometric parameters;
hms_sparsified_cal: sparsified hit parameters;
hms_clusters_cal: hit clusters;
hms_tracks_cal: track tired clusters;
hms_cal_pedestal: pedestal parameters;
hms_cal_const: calibration constants;
hms_cal_monitor: relative gains;
hms_cal_flags: debugging flags;
hms_cal_zero: ?
hms_cal_normalized: Normalized to momentum energy depositions.

In /INCLUDE/hms_data_structures.cmn:

hms_raw_cal: raw hit data;
hms_decoded_cal: decoded data;
hms_track_test: per track PID information;
hms_physics_r4: main PID information.

Common Blocks continued

For each common block, its location is given, what kind of variables it holds (cluster quantities, track quantities …), in which subroutine is filled.

The hms_sparsified_cal block keeps data on the sparsified hit modules. It is filled \
in h_sparsify_cal subroutine. The variables are:

hcal_rows - row number of a sparsified hit module;

hcal_cols - column number of a sparsified hit module;

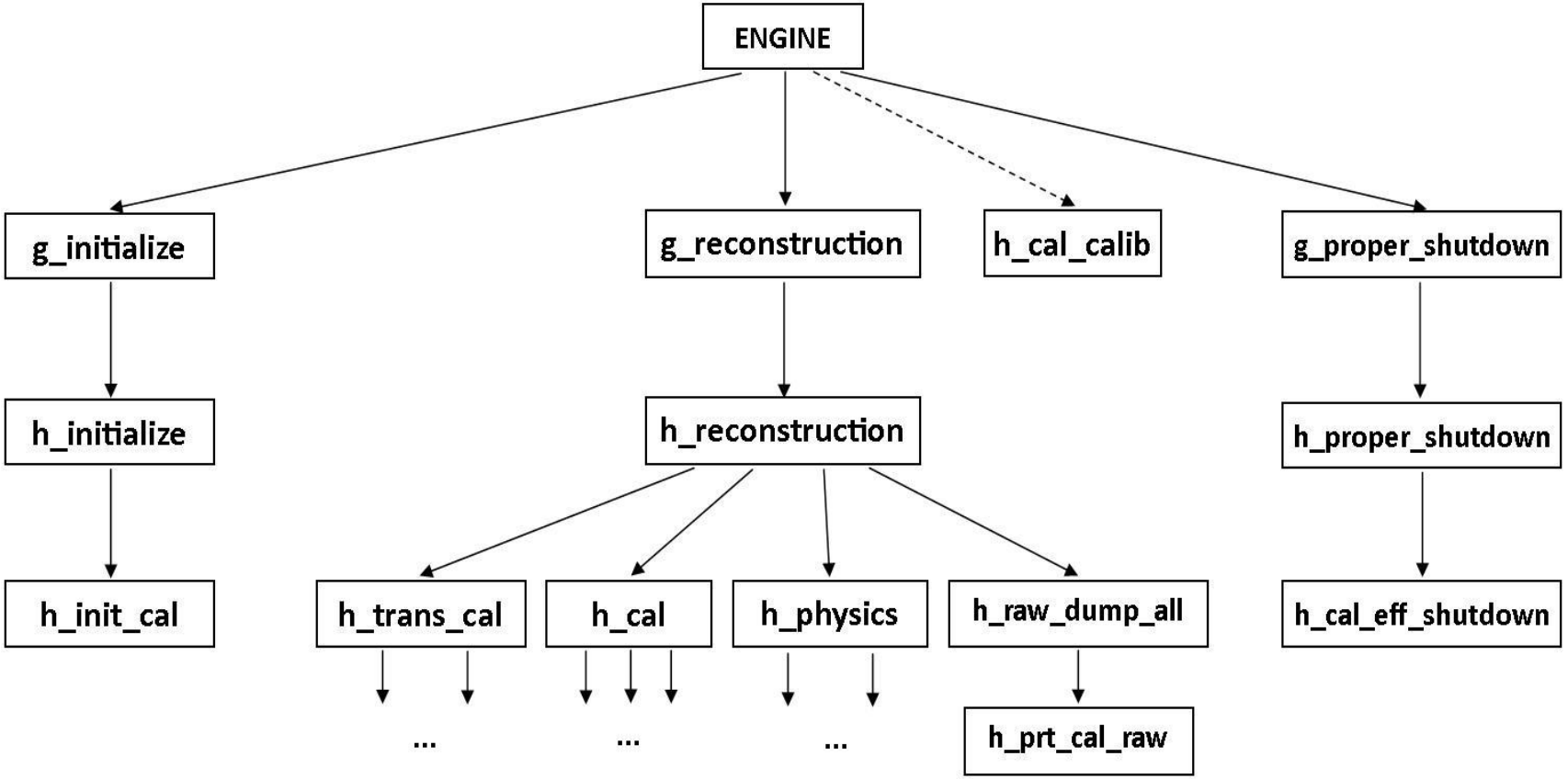
hcal_adcs_pos - ADC pulse height in the positive channel (pedestal subtracted);

hcal_adcs_neg - same for the negative channel;

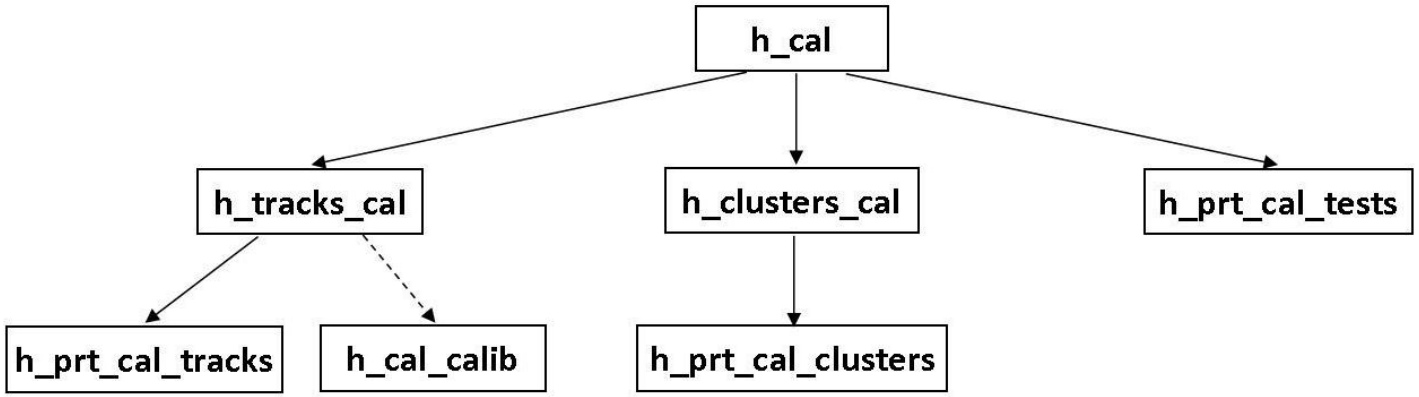
hcal_num_hits - total number of the sparsified hit modules.

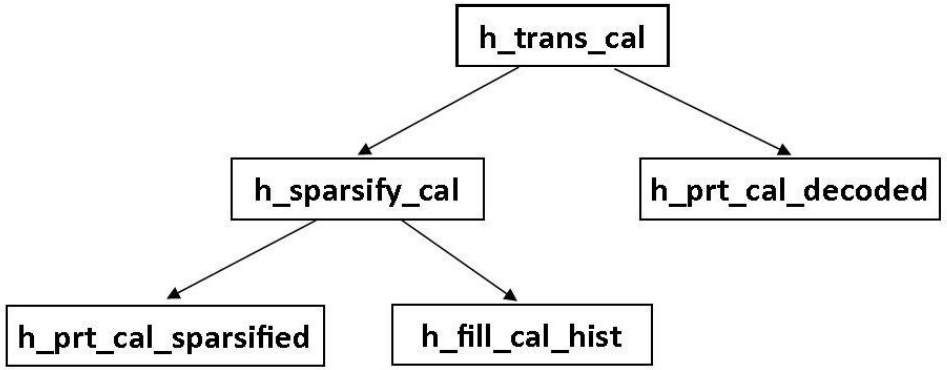
All the variables in this block are arrays of HMAX_CAL_BLOCKS dimension, except hcal_num_hits.

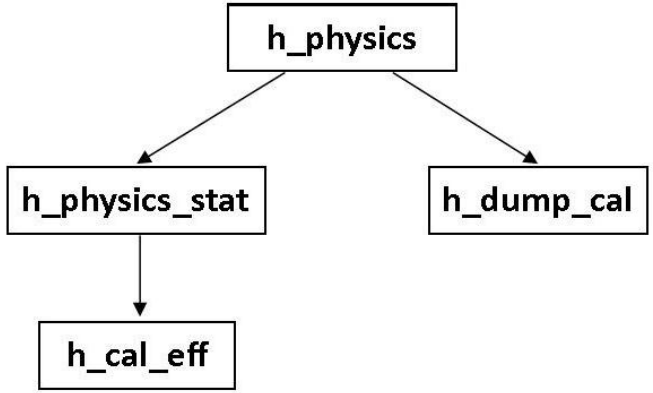
The Code Flow



The Code Flow continued







The Code Flow continued

Main subroutines' description:

- at which stage are called, call chain;
- what is calculated (algorithm, optional);
- filled blocks and calculated variables;
- which variables, and from which blocks are used.

Debugging subroutines:

- input flag for use;
- stage from which it is called;
- calling subroutine;
- printout information.

Calibration:

- the flag to calibrate;
- points of call, and input flag value;
- calculations at each call;
- calibration algorithm.

The Code Flow continued

As an example:

… … …

Next, h_cal subroutine is called for computation of track related calorimeter qunatities. This forks to h_clusters_cal subroutine at first for clustering of the hits.

The cluster is defined as a set of adjacent hit blocks which share common edge or corner. Starting from hit number 1 as a seed and cluster number 1, the code looks for adjacent hits around the hit seed and tags them with the cluster number. When the cluster is filled, the next untagged hit is chosen as a seed, the cluster number is incremented, and the next cluster is filled.
The process goes on until all the hits are tagged with a cluster number. Upon return from the subroutine hms_clusters_cal block is filled. For each cluster energy depositions per layer (from positive and negative channels separately, and the sum of them as well), total energy deposition and energy weighted average X coordinate are calculated. Note that the deposited energies are still not coordinate corrected yet.
The total number of found clusters is also determined.

… … …

To Do List

1, Streamline the list of common blocks with order of calculations (code flow).

2. Modify the code flow diagrams to distinguish the mandatory calls (main subroutines), the debugging calls and the calibration calls.

3. Add description of the debugging subroutines.

4. Add description of the calibration.

5. Produce latex and pdf formats.