The calorimeter coding in the Hall C data analysis package ENGINE

V. Tadevosyan
(tadevosn@jlab.org)

May 14, 2012

1 Construction of HMS/SOS calorimeters in short

The HMS calorimeter is situated at the very end of the detector stack, on a platform attached to the rear wall of the detector hut. Looking along the spectrometer's central ray, it consists of 4 layers of 13 transversely oriented modules. Each module consists of an optically isolated $10 \times 10 \times 70\ cm^3$ lead glass radiator, and optically coupled to it one (in the last two layers) or two (in the first two layers) PMTs. In the single PMT modules the PMT is attached from the right side. In ENGINE, the right side PMTs are dubbed pos (for positive), and the left side PMTs are dubbed neg (for negative). In some cases the forward layer is referred to as pr, for "Preshower", and the rest as ta, presumably for "Total Absorption".

The lead glass calorimeter of the retired SOS spectrometer was of identical construction, except the number of modules in layers 11. The analysis code for the SOS calorimeter mirrors that for the HMS calorimeter.

2 Input parameter files for the HMS calorimeter

In ENGINE, the primary geometry of the calorimeter is nominally described in **replay/PARAM/hcal.pos** file. The input parameters there are:

hcal_num_neg_columns - number of the layers with negative PMTs ($\leq$2);

hcal_1pr_zpos - Z position of the front layer (Preshower);

hcal_<i>ta_zpos - Z positions of the rear layers, i=2,3,4;

hcal_1pr_thick - thickness of the front layer (10 cm);

hcal_<i>ta_thick - thicknesses of the rear layers, i=2,3,4 (10 cm);

hcal_1pr_nr - number the of modules in the front layer (13);

hcal_<i>ta_nr - number of the modules in the rear layers, i=2,3,4 (13);

hcal_1pr_left, hcal_1pr_right - left and right end positions of the lead glass blocks in the front layer (±35 cm);

hcal_1pr_top - top positions of hcal_1pr_nr modules in the front layer (in HMS coordinate system);

hcal_<i>ta_left, hcal_<i>ta_right - left and right end positions of the lead glass blocks in the rear layers, i=2,3,4;

hcal_<i>ta_top - top positions of hcal_<i>ta_nr blocks in the rear layers, i=2,3,4;

In the data analysis, the energy deposition in a lead glass block is reconstracted by multiplying the raw ADC signal amplitude from an attached PMT by a constant factor. These constants, per channel, are obtained in the calorimeter calibration process, and are kept nominally in the **hcal.param** file, beside the other parameters. The content of the **hcal.param** file is:

hcal_slop - a slop parameter used in the calculation of the maximum allowed distance in the vertical direction between the particle track and attributed to it cluster of energy deposition in the calorimeter;

hcal_fv_test - a flag to activate the calorimeter fiducial volume test when tracking;

hcal_pos_cal_const - the calibration constants for the positive channels (see more on this below);

hcal_neg_cal_const - same for the negative channels;

hcal_pos_gain_ini - relative PMT gains per positive channel from the last calibration (kept 1);

hcal_neg_gain_ini - same for the negative channels;

hcal_pos_gain_cur - current relative gains per positive channel (kept at 1);

hcal_neg_gain_cur - same for the negative channels;

hcal_pos_ped_limit - initial high limits on the raw ADC signals for the selection of pedestal events, used in pedestal event analysis;

hcal_neg_ped_limit - same for the negative channels;

hcal_pos_gain_cor - corrections to the gains, ratios of the current and initial gains, per channel;

hcal_neg_gain_cor - same for the negative channels.

It should be noted here that, relative to the listed above initial purposes, the meaning and usage of the gain parameters are altered in the code. As the calibration procedure of the calorimeter provides abseolute calibration constants for a calibrated run, there is no need to use the relative gains hcal_pos_gain_ini, hcal_neg_gain_ini, hcal_pos_gain_cur and hcal_neg_gain_cur. Rather, the calibration constants are assigned to the gain correction constants (hcal_pos_gain_cor, hcal_neg_gain_cor), and the latter are used in the analysis code to convert the raw ADC amplitudes to the energy depositions in the modules. The initial and current gains are not present in the code any more, and are kept in the parameter files for historical reasons. The parameters hcal_pos_cal_const and hcal_neg_cal_const took role of the MeV to GeV conversion constants, for the calibration code provides the constants in MeV/(ADC channel) units, and therefore are kept at 0.001.

3 Calorimeter related constants

A number of calorimeter related constant parameters in the code have the following meanings:

hmax_cal_blocks = 52, total number of modules in the calorimeter, defined in the **hms_data_structures.cmn** file;

hmax_cal_rows = 13, number of the calorimeter rows (or number of modules per layer), defined in the **hms_data_structures.cmn** file;

hmax_cal_columns = 4, number of the calorimeter columns (or layers), defined in the **hms_data_structures.cmn** file;

hnclusters_max = 5, maximum number of the energy deposition clusters allowed in the calorimeter, defined in the **hms_calorimeter.cmn** file;

hntracks_max = 20, maximum number of tracks allowed in the calorimeter, defined in the **hms_data_structures.cmn** file.

4 Calorimeter related common blocks

The calorimeter related variables are grouped in a number of common blocks. Unless is stated otherwise below, they are found in the **Analysis/INCLUDE/ hms_calorimeter.cmn** file.

The common block hms_cal_parms comprises the primary geometrical parameters from the **hcal.pos** file.

The secondary geometrical parameters, which are derivatives of the primary parameters, are kept in the hms_geometry_cal block. This block is filled in h_init_cal subroutine. The variables are:

hcal_block_<i>size - block dimensions in i=x,y,z directions $(10x70x10cm^3)$;

hcal_block_<i>c (hmax_cal_blocks) - i=x,y,z, coordinates of the block centers;

hcal_<i>min, hcal_<i>max - boundaries of the calorimeter volume in i=x,y,z directions;

hcal_fv_<i>min, hcal_fv_<i>max - boundaries of the calorimeter's fiducial volume in i=x,y,z directions.

The hms_cal_const block holds the constants hcal_pos_cal_const and hcal_neg_cal_const. Both of them are arrays of HMAX_CAL_BLOCKS length. Note that, as it is explained above, these quantities are not the calibration constants, but MeV to GeV conversion constants (0.001).

The hms_cal_monitor block groups the relative gains and correction factors hcal_pos_gain_ini, hcal_neg_gain_ini, hcal_pos_gain_cur and hcal_neg_gain_cur from the hcal.param input file. Note that these quantities are not used in the code, as explained above.

The calorimeter ADC pedestal data are grouped in the hms_cal_pedestals block, which is filled in the h_calc_pedestal subroutine. The variables in the block are:

hcal_pos_ped_mean (hmax_cal_blocks) - mean pedestal values for the positive channels;

hcal_neg_ped_mean (hmax_cal_blocks) - same for the negative channels;

hcal_pos_ped_rms (hmax_cal_blocks) - RMS deviations of the pedestals for the positive channels;

hcal_neg_ped_rms (hmax_cal_blocks) - same for the negative channels;

hcal_pos_threshold (hmax_cal_blocks) - high limits on the calorimeter ADC signals, below which a signal is treated as a pedestal event; are typically 3 times the RMS deviations above the pedestal mean values.

The block hms_raw_cal, found in the INCLUDE/hms_data_structures.cmn, comprises calorimeter raw hit data. Two fields in the block, hcal_realadc_pos and hcal_realadc_neg are filled in the h_sparsify_cal subroutine, the rest is obtained from the signal decoding in the g_decode_fb_bank subroutine.

The block variables are:

hcal_column (hmax_cal_blocks) - raw hit column numbers;

hcal_row (hmax_cal_blocks) - raw hit row numbers;

hcal_adc_pos (hmax_cal_blocks) - ADC signals in the positive channels;

hcal_adc_neg (hmax_cal_blocks) - same in the negative channels;

hcal_realadc_pos (hmax_cal_blocks) - ADC signal amplitudes in the positive channels;

hcal_realadc_neg (hmax_cal_blocks) - same in the negative channel;

hcal_tot_hits - total number of the raw hits;

hcal_pos_hits - number of the positive channel raw hits (not used);

hcal_neg_hits - number of the negative channel raw hits (not used).

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

hcal_rows (hmax_cal_blocks) - row numbers of the sparsified hit modules;

hcal_cols (hmax_cal_blocks) - column numbers of the sparsified hit modules;

hcal_adcs_pos (hmax_cal_blocks) - ADC pulse heights in the positive channels of the hits;

hcal_adcs_neg (hmax_cal_blocks) - same for the negative channels;

hcal_num_hits - total number of the resultant sparsified hit modules.

The block hms_decoded_cal, found in the hms_data_structures.cmn file, comprises the calorimeter decoded data. It is filled in the h_trans_cal subroutine. The block variables are:

hblock_xc (hmax_cal_blocks) - X coordinates of the decoded hits;

hblock_zc (hmax_cal_blocks) - Z coordinates of the decoded hits;

hblock_de (hmax_cal_blocks) - energy depositions of the decoded hits, sums over the negative and the positive channels;

hcal_e<i> - total energy depositions in the layers i=1,2,3,4 from the decoded hits;

hcal_et - total energy deposition in the calorimeter from the decoded hits;

hcal_e<i>_pos - decoded energy depositions in the layers i=1,2 from the positive channels;

hcal_e<i>_neg - same from the negative channels;

hnhits_cal - total number of the decoded hits;

hblock_de_pos (hmax_cal_blocks) - decoded hit energy depositions from the positive channels;

hblock_de_neg (hmax_cal_blocks) - same from the negative channels.

The hms_clusters_cal block comprises data on the hit clusters. It is filled in the h_clusters_cal subroutine. The list of variables is:

hcluster_hit (hmax_cal_blocks) - cluster numbers for the hit modules;

hcluster_size (hnclusters_max) - numbers of the hits in the clusters;

hcluster_xc (hnclusters_max) - X (vertical) coordinates of the clusters;

hcluster_e<i> (hnclusters_max) - cluster energy depositions in the layers i=1,2,3,4;

hcluster_e<i>_pos (hnclusters_max) - cluster energy depositions in the layers i=1,2 from the positive channels;

hcluster_e<i>_neg (hnclusters_max) - same from thr negative channels;

hcluster_et (hnclusters_max) - total energy depositions in the clusters;

hnclusters_cal - total number of clusters.

The calorimeter track quantities are in the hms_tracks_cal block, which is filled in the h_tracks_cal subroutine. The variables are:

htrack_<i>c (hntracks_max) - x,y coordinates of the tracks at the front of calorimeter, i=x,y;

hcluster_track (hntracks_max) - numbers of the associated energy deposition clusters;

hntracks_cal - number of tracks for which a cluster was found.

The block hms_track_tests, found in the hms_data_structures.cmn file, holds per track particle ID information from the calorimeter. All the quantities in the block are corrected for the track Y coordinate at the front of the calorimeter. The block is filled in the h_cal subroutine. The variables are:

htrack_e<i> (hntracks_max) - track deposited energies in the layers i=1,2,3,4;

htrack_et (hntracks_max) - track total energy depositions in the calorimeter;

htrack_preshower (hntracks_max) - track energy depositions in the Preshower (equal htrack_e1);

hnblocks_cal (hntracks_max) - numbers of fired modules on the tracks;

htrack_e<i>_pos (hntracks_max) - track energy depositions in the first two layers (i=1,2) from the positive channels;

htrack_e<i>_neg (hntracks_max) - same from the negative channels.

The hms_cal_normalized block comprises the calorimeter main PID quantities. This block is filled in the h_physics subroutine. The variables are:

hscal_sum<i> - total energy depositions in the layers i=a,b,c,d; normalized to the best track momentum, not corrected for the impact point coordinate;

hsprsum - total energy deposition in the Preshower, a.k.a. the first layer; normalized to the best track momentum, not corrected for the impact point coordinate; amounts to hscal_suma.

hsshsum - total energy deposition in the calorimeter (summed over all the hits), normalized to the best track momentum; not corrected for the impact point coordinate.

hsprtrk - energy deposition in the Preshower from the best track, normalized to the track momentum, corrected for the impact point coordinate.

hsshtrk - energy deposition in the calorimeter from the best track, normalized to the track momentum, corrected for the impact point coordinate.

Another part of the particle ID information from the calorimeter is hold in the hms_physics_r4 block, found in the hms_data_structures.cmn file. This block is partially filled in the h_physics subroutine. The block variables are:

hstrack_et - total energy deposition in the calorimeter from the best track, normalized to the momentum;

hstrack_preshower_e - deposited in the Preshower energy from the best track, normalized to the momentum;

hstrack_e<i>_pos - energy depositions in the first two layers (i=1,2) from the positive channels, from the best track;

hstrack_e<i>_neg - same from the negative channels.

The first two variables take their values in the h_physics. The rest appeared to be not in use, and do not take any values.

Related to the calorimeter efficiency variables are grouped in the hcal_statistics block, found in the **hms_statistics.cmn** file. The block is filled in the h_cal_eff and h_cal_eff_shutdown subroutines. The variables in the block are:

hstat_cal_numevents - number of events used in the calculations of the calorimeter efficiency.

hstat_cal_trk (hmax_cal_columns, hmax_cal_rows) - number of times a track passed near the center of the block;

hstat_cal_hit (hmax_cal_columns, hmax_cal_rows) - number of times the module fired, and a track passed near the center of the block;

hstat_cal_trksum (hmax_cal_columns) - number of times a track passed near a block center in the layer;

hstat_cal_hitsum (hmax_cal_columns) - number of times a track passed near a fired block center in the layer;

hstat_cal_eff (hmax_cal_columns, hmax_cal_rows) - efficiency for the module;

hstat_cal_effsum (hmax_cal_columns) - efficiency over all the modules in the layer;

hstat_cal_slop - maximum allowed distance from the center of a block to a track for the efficiency calculations;

hstat_cal_maxchisq - maximum allowed chi squire for a track to be used in the efficiency calculations.

The last two parameters are given in the **replay/PARAM/htracking.param** parameter file.

The flag for the efficiency calculations, hbypass_cal_eff is stored in the hms_bypass_switches block (in the **INCLUDE/hms_bypass_switches.cmn** file).

Finally, the hms_cal_zero block found in the **hms_calorimeter.cmn** file, is apparently related to the calorimeter. It holds 6 arrays of hmax_cal_blocks dimension, which are not used in the code.

5 The code flow

5.1 The core calculations

The flow diagram of the calorimeter specific part of the ENGINE analysis program is presented in figures 1 and 2. The calorimeter subroutines are called in the initialization, event reconstruction and shutdown stages of the analysis process.

In the initialization stage the h_init_cal subroutine is called, where the secondary geometric parameters are derived from the primary parameters of the hms_cal_parms block, and the hms_geometry_cal block is filled. Here the

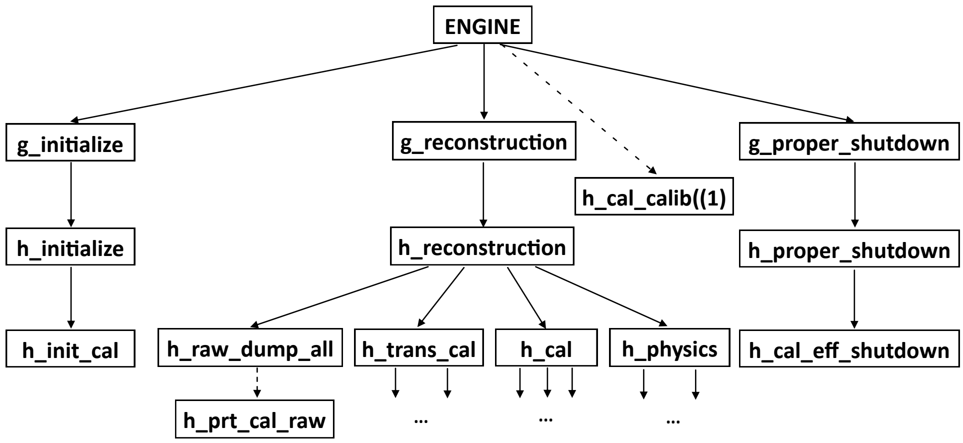

Figure 1: The code flow of the HMS calorimeter part of the ENGINE. The lower branches of the flow are shown separately in the Fig. 2. The mandatory calls are denoted by solid lines, the calls to the debugging subroutines are denoted by dashed lines, the dash-dotted lines indicate calls to the calorimeter calibration subroutine.

number of columns with negative PMTs (hcal_num_neg_columns) is initialized with 0, to be overwritten by the value specified in the input parameter file **hcal.pos** in the sequel.

In the event loop (event reconstruction stage), first the raw ADC signals for the calorimeter are read out and decoded in the g_decode_fb_bank subroutine, and the block hms_raw_cal gets partially filled. Each raw hit is determined by its column number and row number (hcal_column, hcal_row), and by ADC signals in the positive and the negative channels (hcal_adc_pos, hcal_adc_neg). Either hcal_adc_pos or hcal_adc_neg are non-negative. The total number of the raw hits hcal_tot_hits is also determined here.

The first calorimeter subroutine to be called in the event loop is h_trans_cal. It is called from the h_reconstruction, for decoding of the calorimeter raw signals. This at first calls the h_sparsify_cal, were the calorimeter raw hits get sparsified: the pedestal subtracted ADC signals per channel are calculated (hcal_realadc_pos and hcal_realadc_neg arrays in the hms_raw_cal block), and row, column numbers, ADC signal amplitudes of the sparsified hits are determined (hcal_rows, hcal_cols, hcal_adcs_pos, hcal_adcs_neg, content of the hms_sparsified_cal block). A hit is accepted if its signal amplitude either in the positive channel or in the negative channel is above the threshold

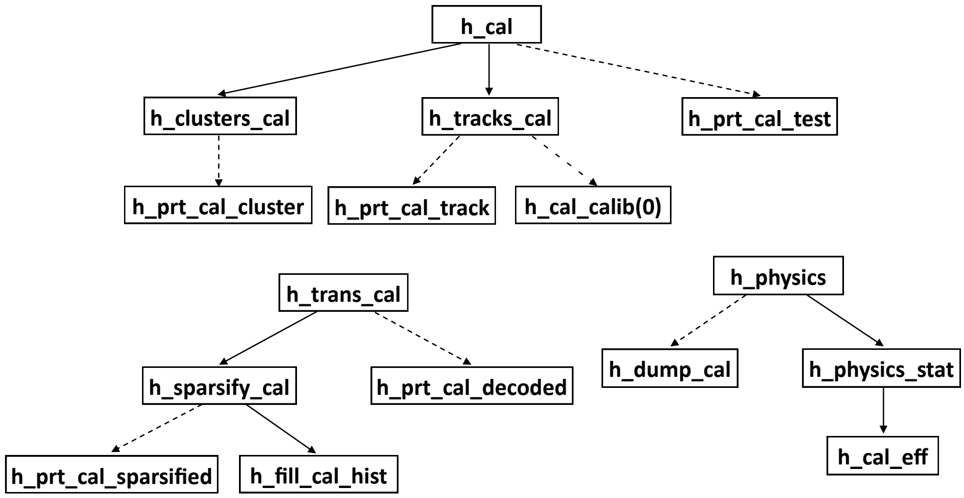

Figure 2: The lower branches of the code flow of the HMS calorimeter. The line type convention is same as in Fig. 1.

(hcal_pos_threshold, hcal_neg_threshold). The number of the sparsified hits hcal_num_hits is also determined there.

After the sparsification, the flow continues with decoding of the sparsified signals in the h_trans_cal, hence filling the hms_decoded_cal block: decoded hit X and Z coordinates (hblock_xc, hblock_zc arrays), and energy depositions (hblock_de, hblock_de_pos, hblock_de_neg) are determined; energy depositions in the layers hcal_e<i>, hcal_e<i>_pos, hcal_e<i>_neg, (i=1,2,3,4), and the total energy deposition in the calorimeter hcal_et are calculated. The hit energy is taken as a product of the sparsified ADC amplitude (hcal_adcs_pos, hcal_adcs_neg), the calibration and the relative gain correction constants (hcal_pos_cal_const, hcal_pos_gain_cor, hcal_neg_cal_const, hcal_neg_gain_cor) for the block. Note that the decoded energies are not corrected for the lateral coordinate.

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

The cluster is defined as a set of adjacent hit blocks which share common edge or corner. Starting from the hit number 1 for a seed, and the 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. This process goes on until all the hits are tagged with a cluster number.

Upon return from the subroutine the hms_clusters_cal block is filled. Each (sparsified) hit is tagged by a cluster number hcluster_hit. For each cluster, energy depositions per layer from the positive and the negative channels separately (hcluster_e<i>_pos, hcluster_e<i>_neg, i=1,2), and the sum of them as well (hcluster_e<i>, i=1,2,3,4), total energy deposition hcluster_et and energy weighted average X coordinate hcluster_xc are calculated. Note that the deposited energies are not coordinate corrected yet. The total number of the found clusters hclusters_cal is also determined.

Once done with the hit clustering, track and cluster matching is performed by a call to the h_tracks_cal subroutine. For each track, clusters are sought within a maximum distance in X direction from the track's position on the face of the calorimeter. The maximum distance is determined as half the thickness of the calorimeter block, plus the slop parameter hcal_slop. Among the found clusters, the closest to the track is associated. If the calorimeter fiducial volume test flag is on (hcal_fv_test$\neq$0) then the cluster matching is done only for the tracks passing through the calorimeter's fiducial volume. Upon exit from the h_tracks_cal, the hcluster_track pointer for each track takes the associate calorimeter cluster number (if a cluster is matched), and the number of matched tracks hntracks_cal is determined. These are part of the hms_track_cal block. This subroutine makes use of the focal plane tracking variables from the hms_focal_plane block.

Back from the track and cluster matching in the h_tracks_cal, the code proceeds with the track's calorimeter quantities in the h_cal subroutine. The block hms_track_tests is filled.

For each track, the energy depositions per calorimeter layer (htrack_e<i>, i=1,2,3,4), the total energy energy deposition in the calorimeter (htrack_et) and in the Preshower alone (htrack_preshower) are calculated from the energy depositions of the associated to the track cluster of hits. The energy depositions per layer from the positive and the negative side channels (htrack_e<i>_pos, i=1,2,3,4, and htrack_e<i>_neg, i=1,2) are also calculated. All the track energy depositions are corrected for the lateral coordinate of the track's projection on the face of the calorimeter. The corrections are different for the single PMT modules and for the double PMT modules. In the latter case, the energy depositions from the negative and the positive channels are corrected separately, then the sum of the two is taken as an energy deposition in a layer. This subroutine takes number of focal plane tracks hntracks_fp from hms_focal_plane block.

The particle ID quantities from the calorimeter, for the best track are determined in the h_physics subroutine. Here, related to the calorimeter part of the hms_physics_r4 block is filled (partially). The total energy deposition in the calorimeter, hstrack_et, and the deposited in the Preshower energy, hstrack_preshower_e, are duplicated from the track variables. Then the hms_cal_normalized block is filled: the normalized to the momentum of the best track sums of energy depositions in the layers, in the Preshower and in the calorimeter are evaluated from the decoded quantities, and the normalized energy depositions in the Preshower (hsprtrk) and in the calorimeter (hsshtrk) are determined from hstrack_preshower_e and hstrack_et quantities.

5.2 Debugging outputs

In each stage of the analysis, information on the calorimeter related quantities can be dumped out for the debugging purposes. The channel number for output, hlun_dbg_cal, and the flags to signal the data dumping are set in the **replay/PARAM/hdebug.param** file.

In the h_prt_cal_raw subroutine, the calorimeter raw hit data are dumped. This subroutine is called from the h_raw_dump_all. The call is triggered by the hdbg_raw_cal flag, found in the **hdebug.param** file. The total number of raw hits hcal_tot_hits is printed out at first. Then, for each hit, the hit number, the row and the column numbers of the hit module, and the ADC amplitude of the signal are printed out (hcal_column, hcal_row, hcal_adc_pos−hcal_pos_ped_mean). If the hit module is of two PMTs, then both the positive and the negative channel ADC amplitudes are printed out.

The sparsified raw hit data are dumped out in the h_prt_cal_sparsified subroutine, after the sparsification of the hits. The call to the subroutine is from the h_sparsify_cal, and is triggered by the hdbg_sparsified_cal flag, found in the **hdebug.param** input file. The total number of the sparsified hits (hcal_num_hits), and, for each hit, the hit number, the calorimeter row and column, the pedestal subtracted ADC amplitudes of the positive and the negative channels (hcal_rows, hcal_cols, hcal_adcs_pos, hcal_adcs_neg) are printed out.

The calorimeter decoded information is printed out by a call to the h_prt_cal_decoded subroutine from the h_trans_cal. The call is triggered by the hdbg_decoded_cal flag, set in the **hdebug_param** file. The total number of the decoded hits hnhits_cal is printed, then, for each hit, the hit number, the X and Z coordinates (in cm), and the energy deposition (in GeV) are printed out (hblock_xc, hblock_zc, hblock_de). Besides, for each calorimeter layer, the layer number and the summed energy deposition is printed out (hcal_e<i>, i=1,2,3,4, hcal_et).

The track related calorimeter quantities are printed out in the h_prt_cal_tracks subroutine, which is called from the h_tracks_cal, after the track and energy deposition cluster matching is performed. The call is triggered by the hdbg_tracks_cal flag. The total number of focal plane tracks hntracks_fp is printed, and, for each track, the track number, the number of the associated cluster of the energy deposition in the calorimeter, the track X and Y coordinates at the front of the calorimeter (in cm) are printed (hcluster_track, htrack_xc, htrack_yc). The total number of the calorimeter tracks hntracks_cal is also printed out.

The information on the clusters of energy depositions is dumped in the h_prt_cal_clusters subroutine, which is called from the h_clusters_cal. The flag hdbg_clusters_cal triggers the call. The total number of clusters hnclusters_cal is printed, then, for each hit, the hit number and the cluster number to which the hit belongs (hcluster_hit) are printed. Then the cluster information proper is printed out: the cluster number, its size (number of hits in it), the X coordinate, the energy depositions in the layers, the total energy deposition, the energy depositions in the first two layers from the positive and the negative channels (hcluster_size; hcluster_xc; hcluster_e<i>, i=1,2,3,4; hcluster_et; hcluster_e<i>_pos, hcluster_e<i>_neg, i=1,2). Note that the energy depositions are not corrected for the impact point.

The calorimeter PID quantities per track are printed out in the h_prt_cal_tests subroutine, which is called from the h_cal. The call is triggered by the hdbg_cal_tests flag. The total number of tracks in the spectrometer hntracks_fp is printed at first, followed by per track quantities: track number, number of fired modules on the track (same as the size of the associated cluster of energy deposition), and energy depositions per layer (hnblocks_cal, htrack_e<i>, i=1,2,3,4). Note that the energy depositions here are corrected for the track coordinate at the front of the calorimeter.

In the h_dump_cal subroutine, an information is printed out which can be used for calibration purposes. The subroutine is called from the h_physics, by the end of the event loop, after the best track is identified. The call is triggered by the hdebugdumpcal flag, found in the **hdebug.param** input file. The track is checked if is in the $\pm 10\%$ momentum bite of the spectrometer, and if is identified as an electron in the gas Čerenkov counter, i.e. if the gas Čerenkov signal is greater than 2. Then the ADC signal amplitude is dumped for each positive (negative) channel, along with the energy deposition in the module for the best track. The energy deposition is normalized to the momentum of the best track and is corrected for the impact point coordinate.

5.3 Efficiency calculations

The efficiency of the calorimeter is calculated in the two subroutines: h_cal_eff and h_cal_eff_shutdown. The calculations are triggered by initializing the flag h_bypass_cal_eff with 0 in the **hdebug.param** file.

In the h_cal_eff, tracks for the efficiency calculations are chosen, and the relevant variables are scored. A good electron track, with χ^2 hschi2perdeg less than the maximum allowed value hstat_cal_maxchisq, and with gas Čerenkov signal hcer_npe_sum greater than 2, is projected onto each of the 4 layers of the calorimeter. In each layer, the pass through block is identified, and the distance from the block center to the track is calculated. If the distance is less than the maximum allowed (hstat_cal_slop), the track counter hstat_cal_track for the block is scored. Next, the hit modules are checked. If the track is passed close to the center of the hit block (distance less than hstat_cal_slop), the hit counter hstat_cal_hit for the module is scored.

The efficiencies for the modules are calculated in the h_cal_eff_shutdown subroutine, as the ratios hstat_cal_hit/hstat_cal_track. The summed over the columns efficiencies are calculated as the ratios of the summed hit counters over the summed track counters.

In the h_cal_eff, the residual distances for the first layer can be histogammed, by establishing the histogram identification number hidcaldpos greater than 0, through the input parameter hcaldpos.

The h_cal_eff subroutine makes use of the hms_physics_r4 block for retrieving the focal plane track quantities, and the hcer_decoded_cer block for retrieving the gas Čerenkov sum signal. The hidcaldpos parameter is taken from the hms_id_histid block.

5.4 Calibration of the calorimeter

The calorimeter can be calibrated by using a dedicated code embedded in the analysis package. The calibration algorithm minimizes the variance of the detected energy in the calorimeter with respect to the calibration constants, subject to the constraint that the estimate is unbiased (relative to the energy of the primary particle). The code is distributed among several subroutines in the **Analyzer/HTRACKING/h_cal_calib.f** file. The calibration process is triggered by initializing the hdbg_tracks_cal flag with a negative value.

The main calibration subroutine, the h_cal_calib takes a single input parameter, named mode. When the mode is 0, accumulation of the data for the calibration is performed, otherwise the calibration proper takes place.

Accordingly, there are two calls to the h_cal_calib subroutine from the analysis code: with the input parameter 0, at the end of h_tracks_cal subroutine, after tracking is complete and all the track parameters are determined; and the second time at the end of the main subroutine, with the input parameter 1.

At the very first call, an output file named h_cal_calib.raw_data is opened for dumping raw data for the calibration. If the mode is 0, the file is updated with the selected data. Good quality single electron track events within the momentum acceptance of the spectrometer are chosen. The strict, somewhat redundant selection criteria are:

- single track at the focal plane, hntracks_fp = 1;
- single energy deposition cluster in the calorimeter, hnclusters_cal = 1;
- the cluster is associated with track, hntracks_cal = 1;
- the track momentum is within the HMS acceptance, |hdelta_tar| < 10%;
- good electron is detected in the gas Čerenkov counter, hcer_npe_sum > 4;
- the track is relativistic, the velocity from the timing measurement is at one, |hbeta-1| < 0.1.

The data for an event are saved in the h_cal_calib.raw_data, in the form of a "header" line, followed by a number of "hit" lines. The header line comprises the number of calorimeter hits for an event hcal_num_hits, the momentum of the track hp_tar, the track coordinates at the face of the calorimeter and the track slopes at the focal plane (htrack_xc, hxp_fp, htrack_yc, hyp_fp). Then, for each hit, the sparsified signal amplitudes for the positive and the negative channels (hcal_adcs_pos, hcal_adcs_neg), and the block number are saved.

If the input parameter mode is not 0, then the analysis of the selected data starts. The low and the high limits on the summed and normalized to the track momentum signal amplitudes are established by a call to the hcal_raw_thr subroutine at first. These thresholds are used in the sequel, in order to refine the calibration data further. Then the calibration algorithm proper is executed by a call to the hcal_clb_det subroutine.

The hcal_raw_thr subroutine takes 3 parameters: one input parameter for the channel number for the raw data file reading (lun), and two output parameters for the low and the high thresholds on the summed and normalized signal (thr_lo, thr_hi). For each event in the raw data file, the sum of the coordinate corrected signal amplitudes is taken, which is then normalized to the track momentum. The mean value and the RMS variation of the signal distribution are calculated, and the thresholds are determined as 3 sigma deviations from the mean value.

The calibration of the calorimeter takes place by calling the hcal_clb_det subroutine. The subroutine takes 4 input parameters: the channel number for reading the raw data file (lun), the run number (nrun), and the low and high thresholds on the summed and normalized raw signal (thr_lo, thr_hi). It reads in again events from the raw data file, by a call to an auxiliary subroutine h_get_data, and constructs the correlation matrix of the signals (qm), the vector of the average signals (q0), the vector of the average products of the signals and the energy of the primary electron (qe), and the average energy of the primary electron e0. Only events with summed and normalized signals within the thresholds (thr_lo, thr_hi) are taken into account in these calculations. As further calculations involve inversion of the correlation matrix qm, a refined copy of the matrix (aux), and the vectors as well (q0s, qes), are constructed in order to avoid the possible singularity problem in the inversion. The refined copies include only data from channels which has fired number of times above a threshold minf of 200.

Finally, the calibration constants are obtained by a call to the subroutine calib, with the refined data as input parameters. The inversion of the correlation matrix is done by means of a call to a copy of the standard CERNLIB subroutine SMXINV.

The resultant calibration constants are saved in the output file **PARAM/hcal.param.<run_number>**, in the format suitable to the standard **hcal.param** file of the ENGINE. By the end of the calibration, the code outputs an auxiliary data, which can be used for evaluation of the quality of the calibration. Namely, the total energy deposition in the calorimeter, the track momentum, and the coordinates at the front of the calorimeter are dumped on event by event basis in the output file h_cal_calib.cal_data.

6 Use of the calorimeter PID quantities

The track energy deposition htrack_et is used for selection of the best track in the h_select_best_track and h_select_best_track_using_scin subroutines. Among the other constraints, low and high limits on the htrack_et are applied in order to select the best track. The hsel_etmin and hsel_etmax limits are established in the **htracking.param** parameter file, and are part of the hms_chose_one_track_r4 block.

The calorimeter PID ntuples are filled in the h_ntuple_keep subroutine. The best track energy deposition hsshtrk is routinely used in the data analyses. The energy deposition in the Preshower hsprtrk can augment the PID capabilities of the calorimeter. The summed energy depositions hscal_sum<i>, hsprsum, hsshsum can be used for estimations of the coordinate dependences of the calorimeter signals, background levels and quality of the tracking, etc.