Reconstruction program for the highly granulated TESLA calorimeter with analog tile-HCAL

Long–writeup

DESY–FLC group by

V.L. Morgunov

March 15, 2003

"Topsy–Turvy World", Progress 1978, Painter V.A Noskov.

" I said it in Hebrew – I said it in Dutch –
I said it in German and Greek:
But I wholly forgot (and **it vexed me much**)
That English is what you speak ! "
/*L. Carroll, The Hunting of the Snark*/

1 Introduction

The program description was divided for several parts, namely:

0. Calorimeter Geometry and Cell Structure;
1. Particle–flow method and program concept;
2. Common structure of program;
3. Calorimeter reconstruction;
4. Propagation back to IP;
5. Subroutines related to BRAHMS;
6. Utility routines;
7. Common blocks description.

2 Calorimeter Geometry for TESLA Detector in BRAHMS

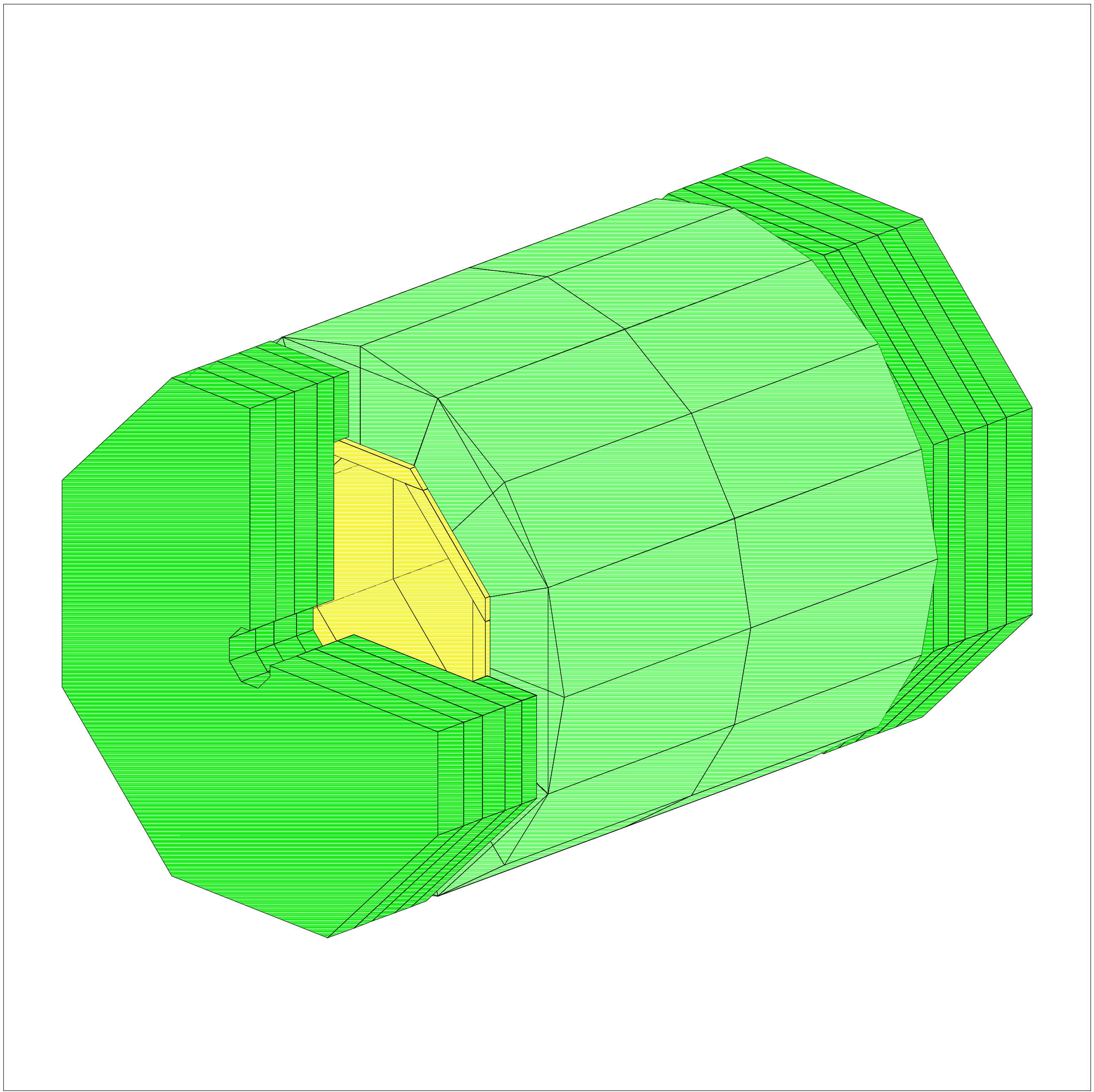

Figure 1: 3–D view of calorimeter for TESLA detector. ECAL is shown in yellow, HCAL – in green

Calorimeter for TESLA detector simulation in BRAHMS is very close to the engineering drawings of TESLA–TDR [**?**]. The BRAHMS geometry of whole calorimeter is shown at the Figures 1,2,3.

TESLA calorimeter consists of two main parts:

ECAL – electromagnetic Si–Tungsten fine granulated calorimeter (shown in yellow) and
HCAL – hadron Tile–Scintillator–Stainless–Steel calorimeter (shown in green).

ECAL barrel consists of 5 octagonal rings (40 modules), see Figure 5.

Sampling structure of ECAL consists of leading part; that is 30 layers of Tungsten (1.4 mm thickness and approximately of 1 x 1 meter in area, differ in depth), and 2.4 mm gap are filled with Si-detector (0.5 mm thickness) and two alveolus surrounded it. And a tail that is 10 layers of Tungsten (4.2 thickness) with the same structure of sensitive gap, see Figure 4.

BRAHMS geometry for barrel part of ECAL was translated from MOKKA geometry (main volume D000). In this geometry every piece of calorimeter is the separated volume.

Two ECAL end–caps (see Figure 5) have exactly the same longitudinal sampling structure as barrel, they are programmed as two octagonal cylinders (volumes ECA1 and ECA2 in accordance with two sampling structures).

HCAL barrel consists of 2 rings (16 modules each), the shape can be seen at Figures 1. Volumes HBA1 and HBA2 are joined into the nontrivial shape – inner octagonal, outer hexagonal cylinders.

HCAL End–Caps consist of 5 octagonal cylinders each (volumes HCA0 – HCA4).

All these volumes are filled with Stainless–Steel and divided into 38 or 53 (for end-cap) sampling structures that consists of 20 mm Stainless–Steel pales and 6.5 mm gap filled with 5 mm thickness scintillator plate (volumes SC–i, area is approximately 1 x 3 meters) and 1.5 mm gap with output fibers (volumes FB–i), see Figure 4.

The whole calorimeter consists of tree different sampling structures, two of ECAL and one of HCAL. That is used at the simulation step – to separate hit storage for the different sampling structures and at the reconstruction step – to get the energy conversion coefficients different for different sampling structures.

BRAHMS subroutines for calorimeter geometry are collected in **calobld_109.car**.

3 Calorimeter Electronic Cell Structure in BRAHMS

ECAL barrel:

Electronic cells in ECAL do not exist as the separate GEANT volumes, they are fictitious, and they are the simple divisions of each Silicon plates into the 1 x 1 centimeter cell starting from the Silicon plate corner. The address of the ECAL–barrel electronic cell in barrel is calculated in subroutine **CELLMAP**, supported by Ecole Polytechnique.

ECAL end–cap:

Volumes ECA1 and ECA2 are divided artificially into the quadrants in subroutine **Find_ECAL_cell** and each quadrant has its own coordinate system. The Silicon plates are divided into the 1 x 1 cm electronic cells in this coordinate system.

Hadron tile–scintillator calorimeter of TESLA TDR has a rather complicated scintillator light read–out, several output fibers from different tiles in depth of the sampling calorimeter structure are joined into one photodetector (so called electronic cell)..

BRAHMS calorimeter model follows this geometry/conditions, see Figure 8.

HCAL barrel: (see Figure 7 and Figure 9 for detail)

Electronic cells in HCAL have complicated shape due to the joining of a different number of scintillation tiles cited in the different sampling layers into one electronic channel.

Scintillator tiles and electronic cells are not exist as a GEANT volume in BRAHMS, they are created artificially.

Each scintillator volume (SCi) has the are size of about 3 x 1 meter, they are varied depends on the HCAL depth. It divided into tiles (from 5 x 5 cm to 20 x 20 cm), then the tiles from different sampling layers are joined in depth into one electronic cell. Sketch of one electronic cell volume is shown at Figure 8.

The database of electronic cell centers for one HCAL barrel module are calculated in subroutine **HBAGEOM**, it can be found in common /HBARGEOM/.

HCAL end–cap:

Volumes HCA0 – HCA4 are divided artificially into the quadrants in subroutine **Find_HCAP_cell** and each quadrant has it own coordinate system. Inside this coordinate system the scintillator plates divided into tiles of differ size and then few tiles in depth are joined into one electronic cell (see Figure 10 and Figure 11 as an example).

The database of electronic cell centers for one HCAL end–cap quadrant are calculated in subroutine **HCAP_CELL_DB**, it can be found in common /HCAPDB/.

The parameters for databases of electronic cells are shown in the Appendix.

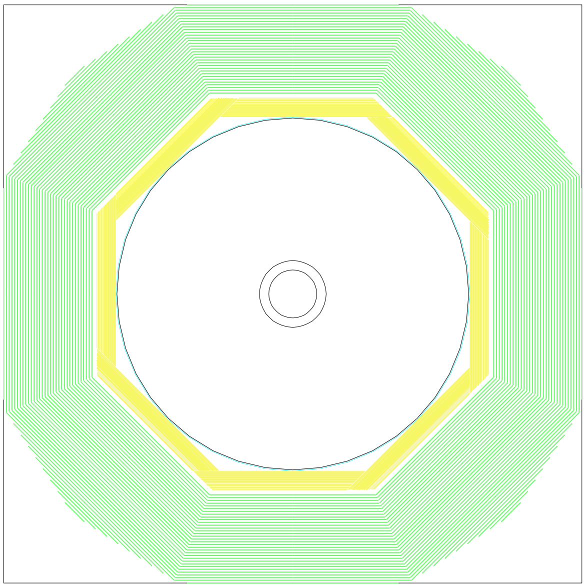

Figure 2: X–Y cut of calorimeter for TESLA detector

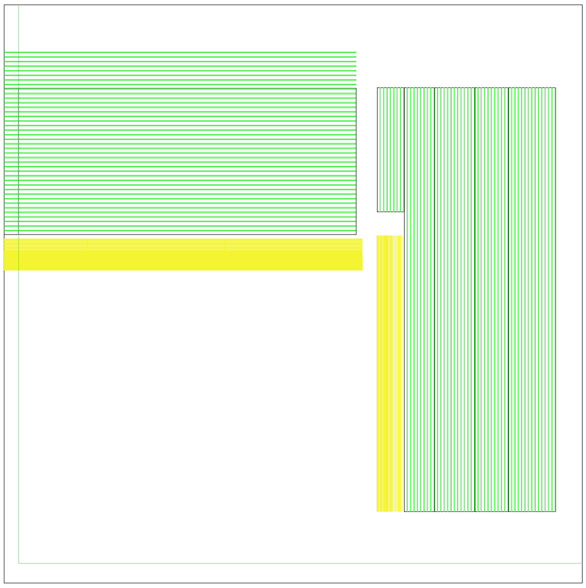

Figure 3: Quote of Y–Z cut of calorimeter for TESLA detector.

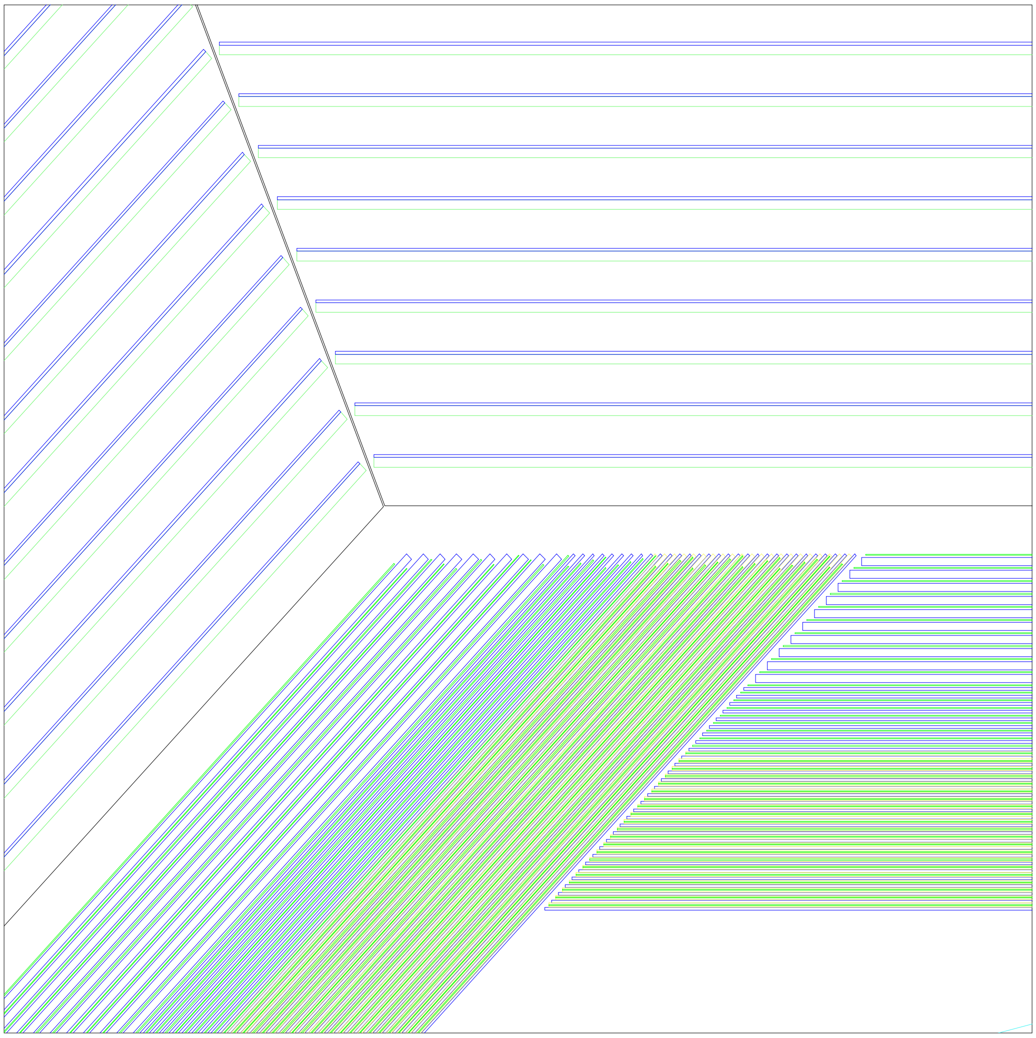

Figure 4: Sampling structures ECAL and HCAL

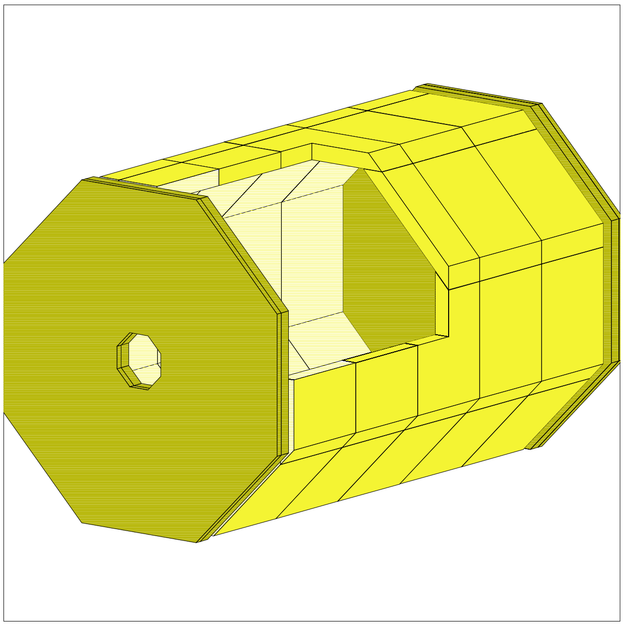

Figure 5: 3–D view of electromagnetic calorimeter

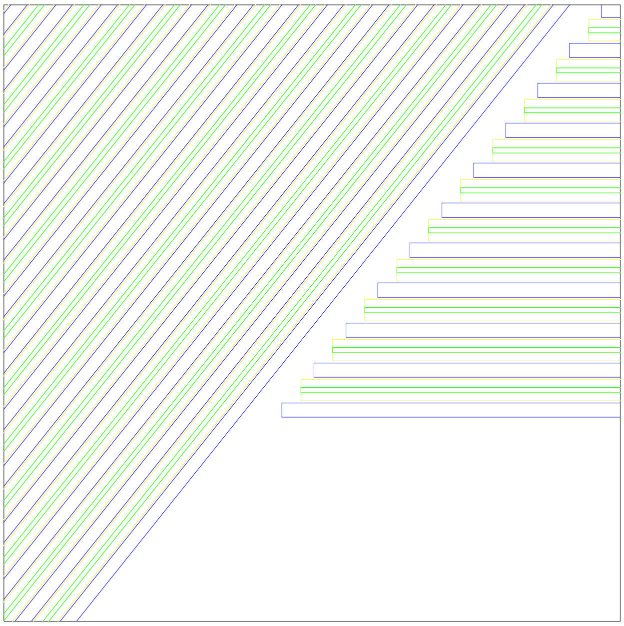

Figure 6: Sampling structure of electromagnetic calorimeter (1/8 corner). Tungsten is in blue, Si–detector is in green, alveolus are in yellow.

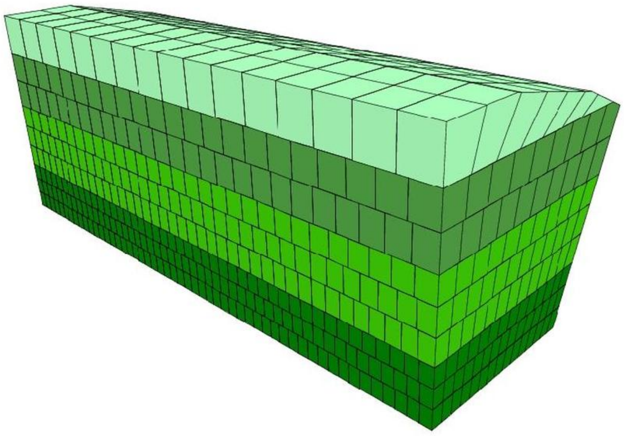

Figure 7: 3–D view of HCAL–barrel electronic cell structure for one barrel module

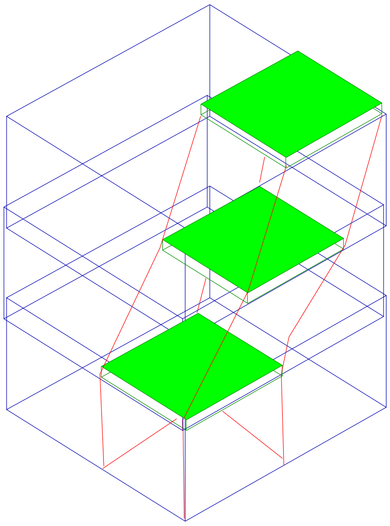

Figure 8: Sketch of spatial position of the scintillator tiles in one electronic cell, absorber plates are shown in blue, cell volume are in red. Sizes of tile are different in different sampling layers.

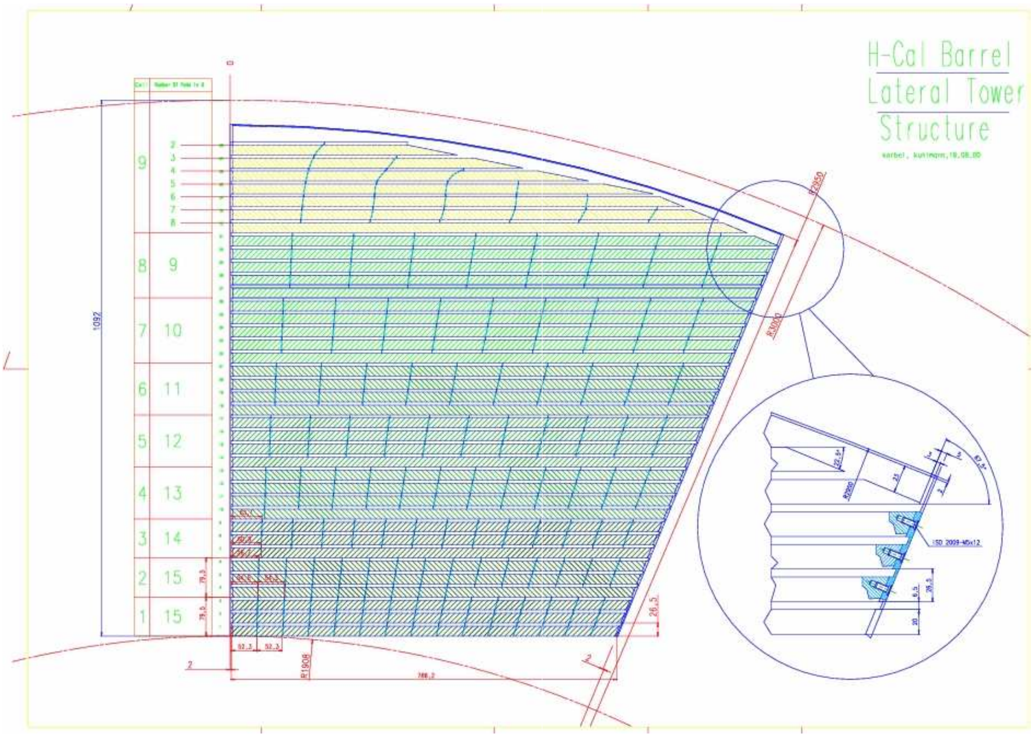

Figure 9: Engineering drawing of barrel module cell structure

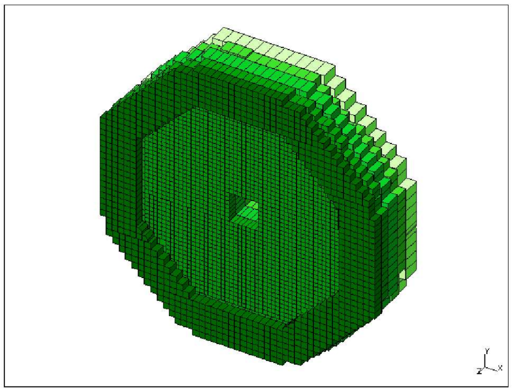

Figure 10: 3–D view of HCAL–End–Cap electronic cell structure

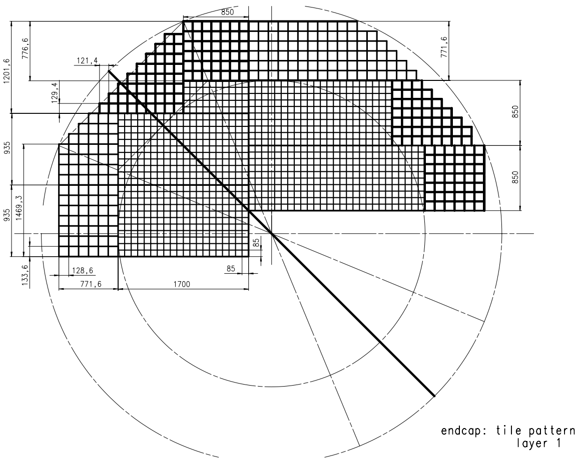

Figure 11: Structure of tiles in the HCAL–End–Cap module 1 (two quadrants).

4 Particle–Flow and Program Concept

The energy flux of stable particles at the interaction point (IP) can be presented as a "hedgehog" of vectors in the direction space $\vec{\Omega}$:

$$E_{IP}(\vec{\Omega}) = \sum_{i}^{N_{IP}} \delta(\vec{\Omega}_i)\sqrt{\vec{p}_i^2 + m_i^2} = \sum_{i}^{N_{IP}} \delta(\vec{\Omega}_i) E_i \tag{1}$$

where: N_{IP} is number of particles appearing at the IP with $\vec{p}_i$, m_i, E_i – the momentum, mass and energy of each particle.

The charged tracks change their initial direction due to the bending in the strong magnetic field that leads to jet energy flux distortion at the calorimeter inner surface.

The energy flux at the calorimeter surface $E_{face}(\vec{\Omega}, \vec{r})$ can be written as:

$$E_{face}(\vec{\Omega}, \vec{r}) = \hat{\mathcal{T}} E_{IP}(\vec{\Omega}) = \sum_{k}^{N_{face}} \delta(\vec{\Omega}_k) E_k(\vec{r}_k) \tag{2}$$

where: $\hat{\mathcal{T}}$ is the transport operator through the relatively small amount of matter (VTX + TPC); N_{face} is the number of particles that reached the calorimeter surface; $\vec{r}_k$ is the particle impact point at the calorimeter surface and $\vec{\Omega}_k$ is the particle direction at the calorimeter surface. The operator $\hat{\mathcal{T}}$ includes: small multiple scattering collisions and small energy losses, particle decay, and the effect of motion in the magnetic field for charged particles.

The energy deposition distribution in the calorimeter volume $E_{dep.}(\vec{r})$ can be written as:

$$E_{dep.}(\vec{r}) = \hat{\mathcal{S}} E_{face}(\vec{\Omega}, \vec{r}) \tag{3}$$

where: $\hat{\mathcal{S}}$ is the shower transformation operator. It produces the spatial energy distribution in calorimeter volume from the initial particle energy flux at the calorimeter surface.

The operator $\hat{\mathcal{S}}$ can be formally represented as a sum of operators of separated particle types (ID) with different behaviour in the calorimeter.

$$\hat{\mathcal{S}} = \hat{\mathcal{S}}_e + \hat{\mathcal{S}}_\gamma + \hat{\mathcal{S}}_\mu + \hat{\mathcal{S}}_{charg.had.} + \hat{\mathcal{S}}_{neut.had.} = \sum_{\text{ID}} \hat{\mathcal{S}}_{\text{ID}} \delta(\text{ID}) \tag{4}$$

The amplitude (A_{ic}) distribution in the calorimeter electronic cells can be written by the simplest operator $\hat{\mathcal{J}}$:

$$A_{ic}(\vec{r}_{ic}) = \hat{\mathcal{J}} E_{dep.}(\vec{r}) = \int_{V_{ic}} C_s \; E_{dep.}(\vec{r}) \; dV \tag{5}$$

where: C_s is the conversion coefficient that depends on the calorimeter sampling structure and particle ID;
V_{ic} – electronic cell volume with $\vec{r}_{ic}$ as a center.

After substitutions we have the set of equations for $A_{ic}(\vec{r}_{ic})$ that are measurable amplitudes at the center $\vec{r}_{ic}$ of electronic cell volume V_{ic}:

$$A_{ic}(\vec{r}_{ic}) = \sum_{\text{ID}} \sum_{k}^{N_{face}} \int_{V_{ic}} C_s \; \delta(\text{ID}) \; \delta(\vec{\Omega}_k) \; \hat{\mathcal{S}}_{\text{ID}} \sqrt{\vec{p}_k^2(\vec{r}_k) + m_k^2} \; dV \tag{6}$$

The number of such equations is equal to the number of calorimeter cells for event with non–zero amplitude in it.

The backward problem for this system of equations is the particle–flow reconstruction problem.
It was solved in some way for many different HEP detectors.

Analogs of such a kind of problem (successfully solved!) are the adjoint problem for the design of nuclear power reactors and the image reconstruction in the modern gamma–chamber in nuclear medicine.

Taking into account the definition of operator $\hat{\mathcal{S}}$ one can recognize well known calorimeter energy–flow formula in this equations, that is a sum of energy over all calorimeter cells: $\mathcal{E} = \mathcal{E}_e + \mathcal{E}_\mu + \mathcal{E}_{charg.had.} + \mathcal{E}_\gamma + \mathcal{E}_{neut.had.}$.

In reality any calorimeter cell can carry the signals from many initial particles!

A few more remarks on this problem:

These formulae are the strict mathematical definition of the bootstrap method for the fast simulation programs.

The equations can be rewritten as a sum of charged and neutral particles separately with the parameterization of shower operators $\hat{\mathcal{S}}$ to make an analytical investigation of its properties.

If one reduces the cell volume up to zero the overlapping still would take place for the shower volume.

At this stage of the investigation of the equations they are not solved directly, but as it was said early, they were solved indirectly in many HEP experiments.

TESLA calorimeter reconstruction program SNARK is one of such a solution.

The idea of solution is based on the shape of the shower operator $\hat{\mathcal{S}}_{\rm ID}$ straightforward. If one have the "exactly" (with very good accuracy) measured particle energy and its direction – so, one can take this knowledge and looks for the calorimeter hits along the predicted particle direction and separates this part of event from the others, that is clear but for the fine calorimeter cell structure only.

The algorithm starts from the finding the charged track core. The track core is the collection of the calorimeter hits which are at the distances less then one cell size from the extrapolation of the TPC measured track (helix curve) into the calorimeter volume.

Such a procedure works well for the muon track with energy more then 2 GeV (at 3 Tesla), for the lower energy a helix can be replaced by more sophisticated curve with taking into account the energy losses. It also works well for the primary ionization part of the hadron track (that is about 20 % of whole deposited energy in the hadron shower). It also works for the electromagnetic shower but due to the fact that the electromagnetic shower is rather short ($<$ 20 cm) and transversally compact in W-Si structure of TESLA ECAL.

The track core collects almost all hits for muon track, the main part of hadron track primary ionization and rather big part of the electromagnetic shower. The longitudinal distribution and average density of hits can be tested for different particle hypothesis even at the level of track core collection.

The next step of the procedure is based on the previous knowledge and estimations; it is the hit collection around the track core at the distances of two cell sizes – called first shell (see picture).

The particle identification is repeated for the new cluster that includes as a track core as a first shell and ... so on.

The iterative procedure is finished when the collected energy become to be "equal" to the input particle momentum. For the muon hypothesis procedure is stopped just after the first step. The word "equal" means equal inside the calorimeter energy resolution, more exactly it

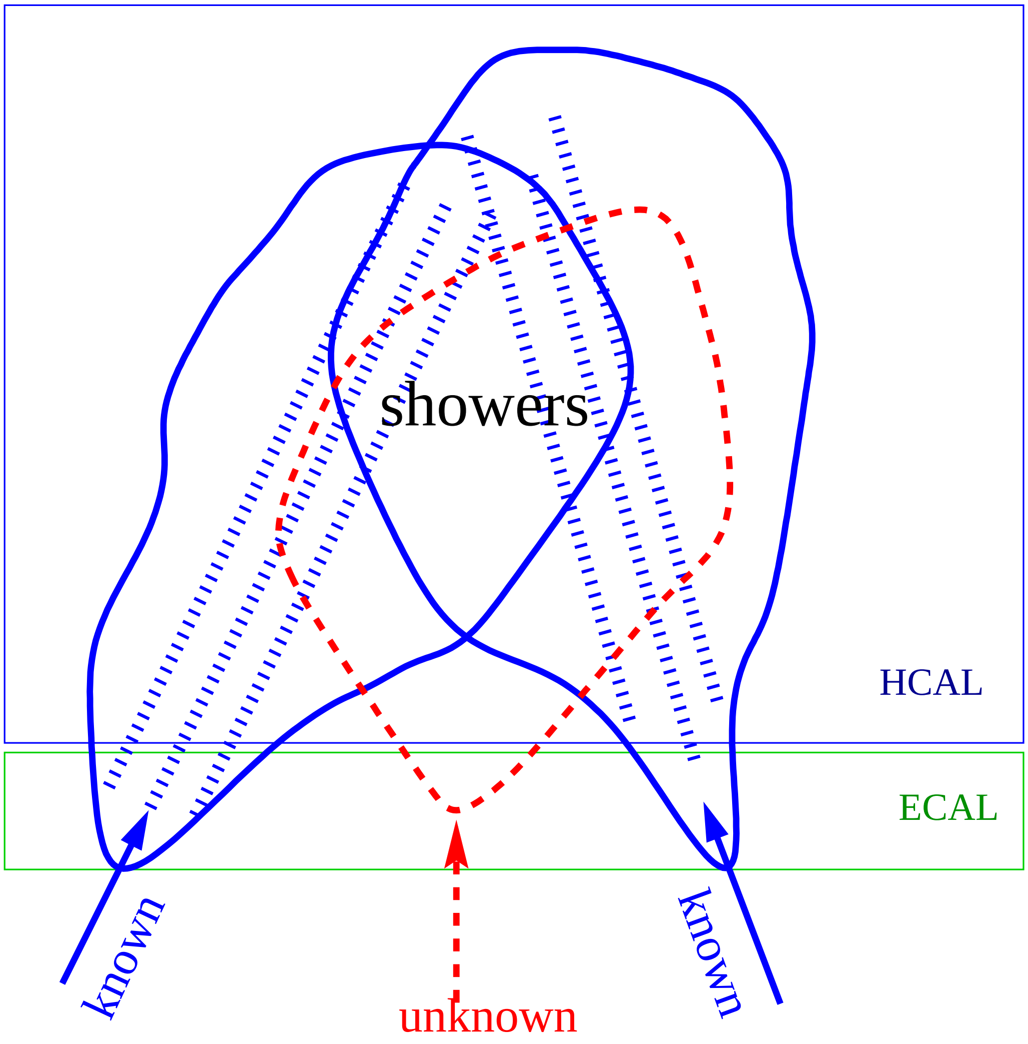

Figure 12: The well measured particle momentum substitutes the pretty random energy spread in the calorimeter volume. This leads to decreasing of the energy fluctuations in general. The vector subtraction from the overlapped showers is more effective in comparison with the scalar subtraction due to the implication of momentum vector.

is particle momentum plus/minus tuning parameters that depend on input energy, calorimeter sampling and particle hypothesis. We will not describe here in details all cases that program trying to resolve during the iterative procedure – its number is about 35 for now. There are different particle hypothesis and their combinations with taking into account the showers overlapping (one can see the text of program).

At the end of loop around charged tracks all hits belong to them are collected in clusters, labeled and they will not use more at the next step of the reconstruction.

The program starts the procedure to collect and separate the neutral part of event after the collecting all clusters for charged particles.

This algorithm applied the hit histograming technique in $\Theta - \Phi$ space to find at first the super–clusters then clusters inside the super–clusters.

Then it collects additional hits around the core of found clusters and it try to create the for every neutral particle as hadron as electromagnetic one.

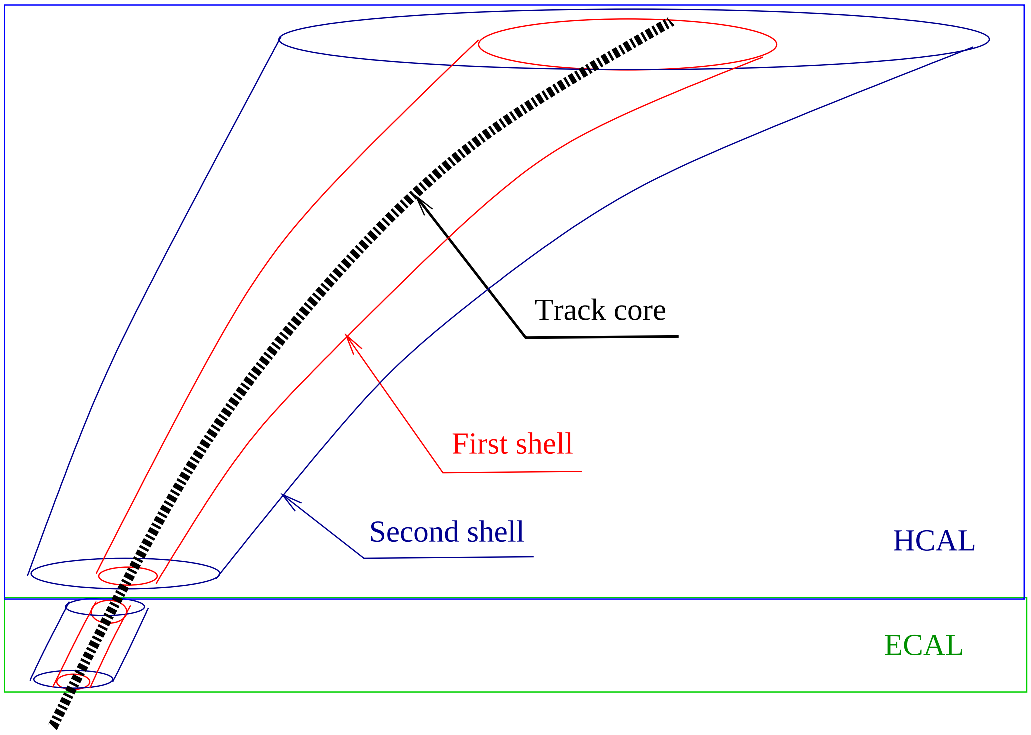

Figure 13: The Particle–Flow algorithm in the reconstruction is: to substitute carefully the particle energy measured by calorimeter for the energy injected by well measured charged tracks along the predicted shower development direction, for each individual charged particle in every event.

The super–clusters in $\Theta - \Phi$ coordinate space are constructed by combining all calorimeter hits which are within a certain distance Δ_1 in $\Theta - \Phi$ space. If two super clusters overlap, the energy inside the overlapped region is assigned to both super–clusters weighted by the total energy of the respective super–cluster. The properties of the super–cluster are calculated: energy, center position and angular radius.

Clusters inside the super–cluster are reconstructed by combining hits assigned to the super-cluster which are within a certain distance $\Delta_2 < \Delta_1$ in $\Theta - \Phi$ space. For the overlapping clusters the same method as above is applied. Then the cluster properties are calculated: energy, number of hits, hit density, center position, principal cluster axes, parameters of electromagnetic shower hypothesis.

Each cluster is assigned to a particle type based on its predicted and calculated shape parameters for gamma or hadron hypothesis. The remaining hits are joined to the closest cluster. The clusters whose main axes are overlapping within a certain window are merged into one cluster. Finally all cluster properties are recalculated.

This procedure is done for both ECAL and HCAL separately. ECAL and HCAL clusters are joined if they match a certain criteria.

Separation of overlapping clusters.

Assuming that the cluster profile in transverse direction can be described by an exponential law, the cluster shape in a logarithmic $\theta - \Phi$ space is simple a cone. If two clusters overlap, the energy of cells in the overlap region is assigned to both clusters, weighted by the linear distance to the respective cluster axis. In the current version of the software up to six clusters are allowed to overlap, at most three in one point.

5 Calorimeter Reconstruction Program

5.1 Program flow

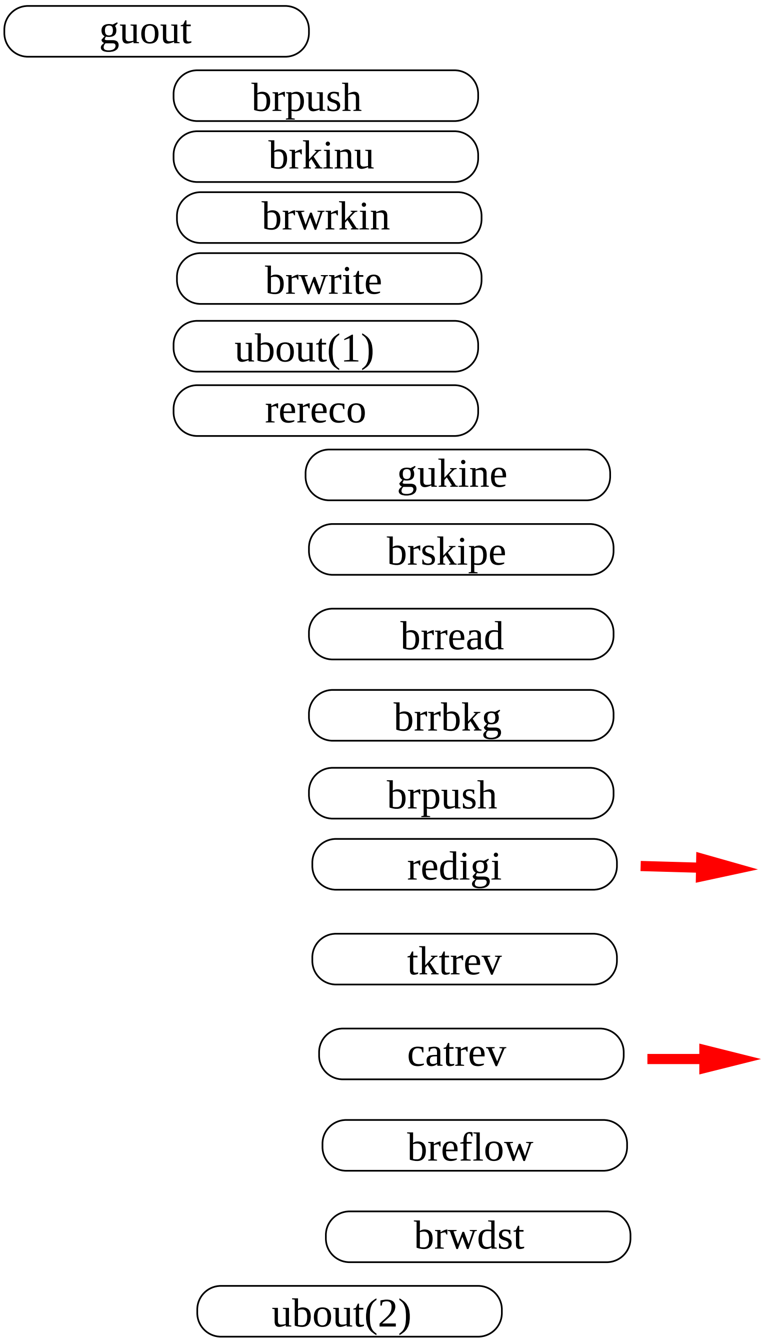

Figure 14: GEANT subroutine flow

5.2 Program flow

Every RUN

Reconstruction initialization

calor_reconst_ini

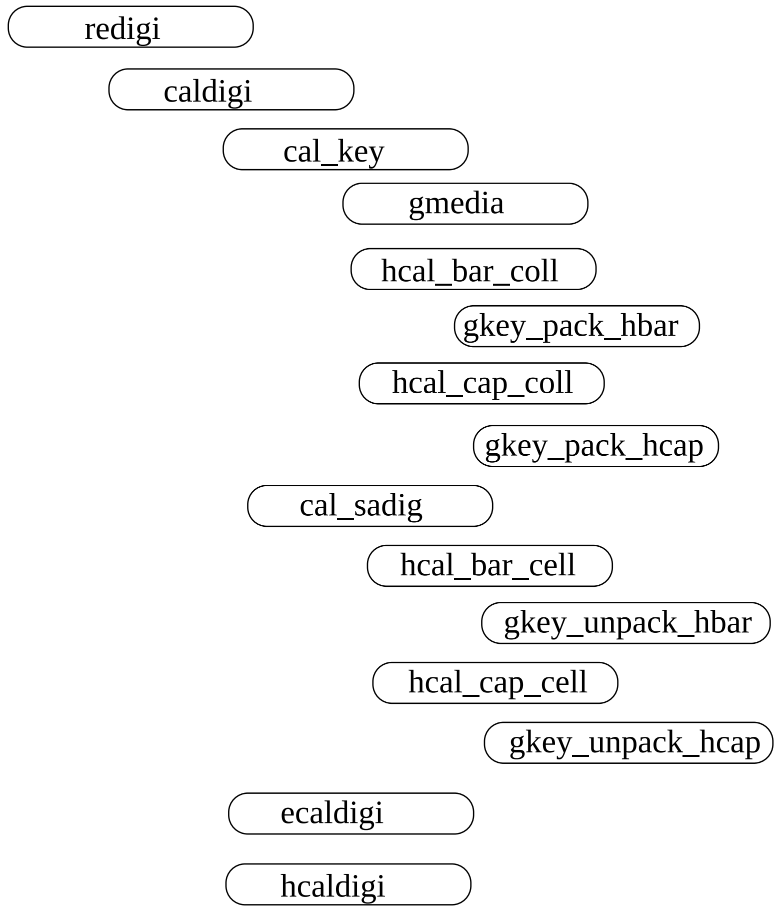

Figure 15: RECO subroutine flow

Every EVENT

Preparation

clearcomm
tpc_reconst_prep
ecal_reconst_prep
hcal_reconst_prep
    hcap_reconst_prep

Reconstruction

calor_reconst

Find clusters belongs to charged tracks

find_cluster_core
find_electron_clust
find_hadron_cluster
find_hadron_clust_3

Find clusters for neutral part of event

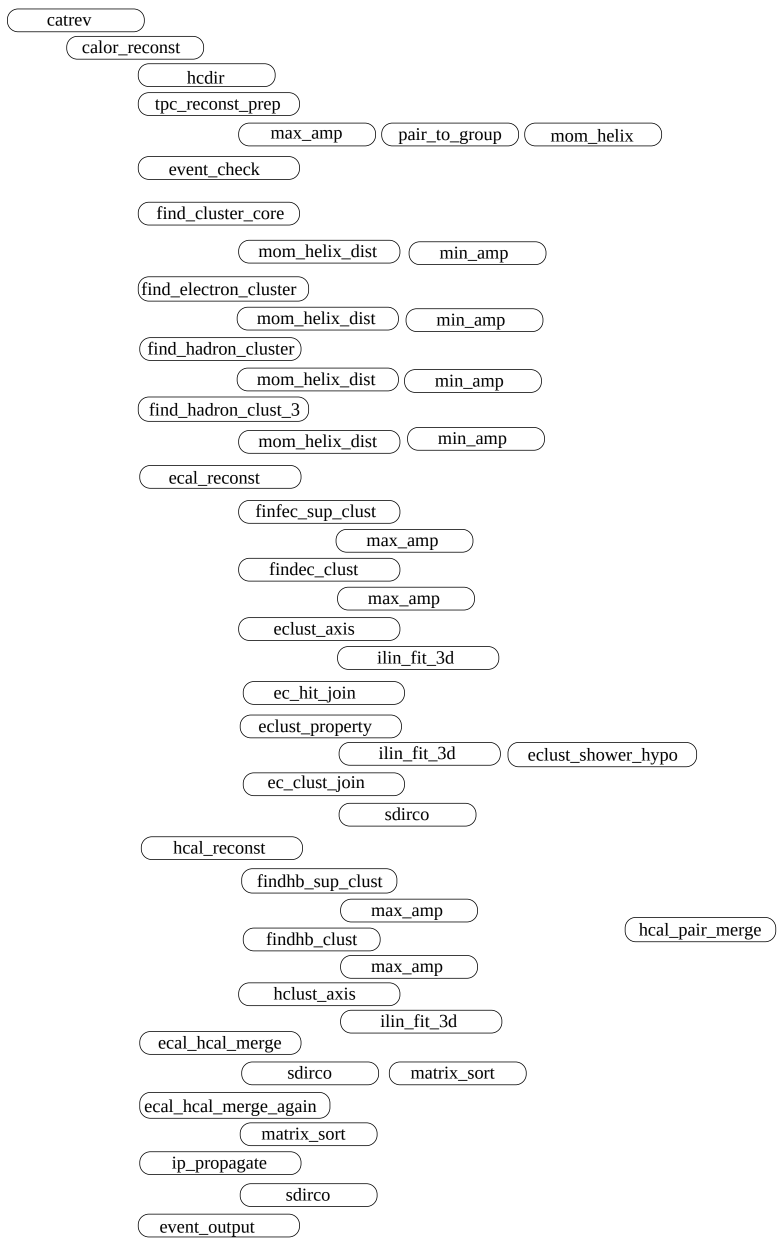

Figure 16: SNARK subroutine flow

ecal_reconst
- findec_sup_clust
- findec_clust
  - ec_clust_join
  - ec_hit_join
  - eclust_axsis
  - eclust_property
  - eclust_shower_hypo

hcal_reconst
- findhb_sup_clust
- findhb_clust
  - hclust_axsis
  - hcal_pair_merge

Join neutral clusters in both calorimeters

ecal_hcal_merge
ecal_hcal_merge_again

Event building at IP

ip_propagate
⇒ **Next event**

5.3 Initialization of Reconstruction Parameters

Subroutine CALOR_RECONST_INI
called once a run; makes an initialization of reconstruction parameters and counters. All reconstruction parameters are in **COMMON / RECONPARAM /.**

In the other environment as BRAHMS this subroutine can be called at any reasonable place.

Brief parameters description:

• **energy_coeff1, energy_coeff2, energy_coeff** – calibration coefficients are taken from muon run (10 GeV) to convert the hit amplitudes to physical scale [GEV] for two ECAL and one HCAL sampling structures;

• **sph_cutE_sup, sph_cutE_clu, sph_cut_sup, sph_cut_clu** – cuts for "angular" distances, used during super–cluster and cluster searching in ECAL and HCAL (see utility function **angle_dist**(Θ,Φ) that calculates an angular distance);

• **coeff_radE_sup, coeff_radE_clu, coeff_rad_sup, coeff_rad_clu** – additional parameters for cluster "angular" radius; it makes sph_cuts to be proportional to Log(Energy);

• **Cl_Dist_Merge** – if the distance between centers of clusters less then this parameter – try to MERGE them;

• **Cl_Dist_Merge_EM** – allow to merge if distance between ECAL cluster and HCAL cluster less then this parameter;

• **Dist_Merge** – minimal distance from line or helix to cluster to MERGE ECAL and HCAL;

• **EC_min_mu** – minimal energy for ECAL clusters to be merged

• **EC_max_mu** – maximal energy for muon can deposit to be recognized as ECAL cluster;

• **EM_Energy** – there are special condition for EM clusters – if ECAL Gamma quanta or electron has an energy greater then this it can produce signal in HACL;

• **x0_1ECAL, x0_2ECAL** – number of X0 in one sampling in first and second ECAL structures, used in subroutine EClust_Shower_Hypo;

• **R_moliere** – Moliere radius of ECAL first structure

• **theor_mu_densE, theor_mu_dens** – the average theoretical density in [MeV/hit] for the muon tracks; used in subroutines Eclust_axsis, Eclust_property and Hclust_axsis;

• **ec1_theor_dens, ec2_theor_dens, ec1_theor_RMS, ec2_theor_RMS** – coefficients to calculate predicted, energy dependent theoretical density in [MeV/hit] and its error for the electromagnetic and hadronic showers. Special electron and gamma runs were made. Afterwards the energy density and its RMS energy dependence were fitted with the simplest $theor\_dens = C1 + C2 * \sqrt{E_{clust}}$, $theor\_dens\_RMS = C3 + C4 * \sqrt{E_{clust}}$ formulae. Coefficients are used in subroutines Eclust_axsis, Eclust_property to appropriate cluster particle identification hypothesis in ECAL;

• **nindbeg_thr, nindend_thr** – thresholds used for particle identification in ECAL;

• **thresh_clust** – threshold distance for ECAL cluster joining; used in subroutine EC_CLUST_JOIN

• **thresh_hits** – threshold distance for ECAL hit joining; used in subroutine EC_HIT_JOIN

• **nhit_thrE, Esup_clustE_thr** – thresholds to create cluster from super-cluster ECAL

• **nhit_thr, Esup_clust_thr** – thresholds to create cluster from super-cluster HCAL

• **TPC_trk_Dist_Merge** – the largest allowed distance between tracks at the calorimeter face to be a group

• **TPC_Cl_Dist_Merge, TPC_Cl_Dist_MergeH** – if the distance between center of cluster and track is larger then this parameter it will not be calculated at all (save CPU time)

• **TPC_Dist_Merge, TPC_Dist_MergeH** – maximal distance from TPC particle to cluster to be MERGED (from strait line along TPC track direction to the center of ECAL/HCAL Cluster)

• **Energy_sigmaE** – actually this value is near by sigma for ECAL 12%/sqrt(E).

• **Energy_sigmaH** – actually this value is near by sigma for HCAL 50%/sqrt(E).

Parameters to choose the subtraction or substitution for the charged tracks with TPC momentum.

• –

5.4 Preparation of the TPC Track Information

TPC track information plays the significant role in the reconstruction with particle–flow technique. Hence program uses the parameters of charged track at the calorimeter face that taken from TPC reconstruction procedure.

Subroutine TPC_RECONST_PREP

calculates distances between charged track impact points at the calorimeter face and track helix hypothesis parameters; makes groups of tracks at the calorimeter face.

Input for this program is the arrays of $x(i)$, $y(i)$, $z(i)$ – coordinates of input point at the calorimeter inner surface, $P_x(i)$, $P_y(i)$, $P_z(i)$ – particle momentum at this point and $Q(i)$ – particle charge.

Now program used **COMMON / CALOINPUT /** that consists of an ideal TPC information.

Detailed actions:

• Subroutine initializes COMMONS.

• Sorts TPC tracks in decreasing order of their momentum.

• Calculates distance matrix for every two tracks at the calorimeter face. The simplest 3-d distance between two points at the calorimeter face is used now.

• Sorts distances between TPC charged tracks in increasing order and creates pairs of the nearest TPC tracks.

• Creates groups of the nearest tracks at the calorimeter face. The set of tracks is defined as a group if the distances between any pairs in group less then limit (see **TPC_trk_Dist_Merge**). The group configuration topologically described as *graf* by number of pairs and number of links between close tracks. Several group topologies exists: compact group – number of links more then number of pairs in group, extended group – number of links less then number of pairs in group, compact group with a tail and so on.

• Calculates coordinates and momentum for group of tracks.

• Creates the list of charged tracks that belong to some group.

• Creates the list of charged tracks that do not belong to any group in decreasing momentum order.

• Calculates parameters of predicted helix of charged particles in the calorimeter volume.

All calculated values are in /TPCTRACK/ and /CALOINPUT/ COMMONs.

5.5 Preparation of Electronic Cell Information

Subroutines ECAL_Reconst_Prep, HCAL_Reconst_Prep, HCAP_Reconst_Prep.

This part of program unpacks (if necessary) and fills the main input common blocks with coordinates and amplitudes of calorimeter hits.

To start the reconstruction **COMMON / ECreconst /** must be filled with:

NhitsE – Number of cells with non–zero amplitude in ECAL;

a_cellE(i) – Amplitude of calorimeter cell in physical units [GeV];

ny_layE(i) – Calorimeter longitudinal structure layer number;

x_cellE(i), y_cellE(i), z_cellE(i) – Cartesian coordinates of cell in [cm], the center of coordinate system suppose to be at the detector geometrical center (GEANT Master Reference Frame);

r_cellE(i), t_cellE(i), p_cellE(i) – Spherical coordinates of cell (R $[cm]$, $0 < \Theta < 180$, $0 < \Phi < 360$) [degree]; Zero of the Θ angle is at the Z+ axes.

The same information for HCAL – **COMMON / HCreconst /**: **Nhits, a_cell(i), ny_lay(i), x_cell(i), y_cell(i), z_cell(i), r_cell(i), t_cell(i), p_cell(i)** has to be filled also.

---

energy_coeff1, energy_coeff2, energy_coeff – coefficients are used in this programs to convert GEANT energy losses to the physical scale (see initialization parameters description).

Hierarchy of cell amplitude scales:

1. Amplitude read from ADC (integer);
2. Amplitude in the detector energy losses units [GeV] (do not include absorber);
3. Amplitude in physical scale [GeV] – include absorber losses.

The second term is generated by GEANT and it is remember as cell amplitude before digitization. The detector electronic noise and non-uniformity must be added at this level i.e. before digitization.

Level 3 is used in the reconstruction.

5.6 Subroutine Calor_Reconst

Steering routine for the event reconstruction in calorimeter. Description of program is made in the order of subroutines call.

The first part of reconstruction program creates clusters for charged particles in both calorimeters using the TPC information:

CALL FIND_CLUSTER_CORE
CALL FIND_ELECTRON_CLUSTER
CALL FIND_HADRON_CLUSTER

and extended search of hadron clusters:

CALL FIND_HADRON_CLUST_3

The second part of program finds clusters in ECAL and HCAL for the rest of hits – the treatment of the neutral part of events:

CALL ECAL_RECONST
CALL HCAL_RECONST

The last step is the cluster merging in both calorimeters:

CALL ECAL_HCAL_MERGE

And the second attempt to find HCAL cluster joined to ECAL cluster:

CALL ECAL_HCAL_MERGE_AGAIN.

5.6.1 Subroutine Find_Cluster_Core

Take TPC charged track and find the nearest hits to the predicted helix model of track.

This routine significantly used the calorimeter electronic cell structure database. It sets for the TESLA calorimeter – has to be rewritten for other calorimeter structures.

For every track it accumulates the information such as chisq, density, and it builds the particle hypothesis first approximation.

Detailed actions:

• Initialize COMMONS.

⇒ Loop along TPC tracks at the calorimeter face.

• Loops along ECAL and HCAL hits to select and collect hits at the one cell distance from predicted helix and distances from impact point and also calculate density of every collected hit. Track density calculation is made in the terms of one sensitive layer and it does not depends on the number of tiles in electronic cell of hadron calorimeter.

• Calculation of properties of collected track core cluster for further analysis, such as starting layer number along a track, final layer number, number of layers with one MIP in it, average density ...

• Analysis starts from the muon hypothesis chooser. The muon hypothesis is chosen when the core was no no gaps in ECAL and HCAL and if it has one MIP in every layer. All core hits are getting the hard assignment to this track.

• Next checked hypothesis is electron or positron. This hypothesis is chosen when the core has energy more then half of input energy (usual core energy is about 0.675–0.7 of input energy for e^+, e^- in W–Si ECAL structure), first part of ECAL has 25% of input energy , core has a big average density (property of electromagnetic shower), core energy in ECAL less then input energy and number of collected HCAL hits are less then 15 (electrons with energy about 250 GeV make about 15 hits in HCAL due to the longitudinal leakage). The hits are getting the soft assignment to this track. The hard assignment is made for the core hits if its energy is within $\pm 10\%/\sqrt{E}$ of input energy.

• Preparation of electron or positron hypothesis to the second iteration. There are few different cases were treated here for input energy less and more then 10 GeV (the electron shower with energy less then 10 GeV almost never can reach the HCAL).

• Next checked hypothesis is hadron that passed through the ECAL without hadron interaction. The hits are getting the soft assignment to this track. The hard assignment is made for the core hits if its energy is within $\pm 30\%/\sqrt{E}$ of input energy.

• Then hypothesis where the hadron track has a MIP in the first 10 layers of ECAL.

• Then hypothesis where the hadron track has "exact" energy in $\pm 20\%/\sqrt{E}$ range along helix prediction and the core energy is in $\pm 30\%/\sqrt{E}$ range around input energy.

• Core energy is MORE then input energy (overlapping of showers). Remark: the core cylinder is very narrow (1 cell) if the energy that collected in such a cylinder more then input momentum then it has to be made the subtraction of energy from every hit along this track. The rest of hits is deassigned. There are several cases are treated here with different shapes of hadronic and overlapped showers.

• The track with core energy less then input energy goes to the second iteration.

⇒ Next TPC tracks at the calorimeter face.

5.6.2 Subroutine Find_Electron_Clust

Taking into account the assignment of electron/positron hypothesis in the previous routine this routine takes TPC track and find cluster for it.

Detailed actions:

• Initialize COMMONS.

⇒ Loop along TPC tracks at the calorimeter face with electromagnetic hypothesis mark.

• Loops along ECAL and HCAL hits to select and collect hits at the six cell distances from predicted helix for ECAL and three cell distances for HCAL.

• Hits analysis along the track.

• Analysis starts from case when the collected cluster energy is in $\pm 14\%/\sqrt{E}$ range around input energy. All hits are getting the hard assignment to this track.

• Then if energy more then input $+14\%/\sqrt{E}$ – showers overlapping. The proportional subtraction and hard assignment were made for the cluster hits that were added at the second iteration to get exact energy of the cluster. The hard assignment for cluster core were made also.

• ADD HERE AN ADDITIONAL TREATMENT FOR SMALL ENERGY OF CLUSTER.

⇒ Next TPC tracks at the calorimeter face.

5.6.3 Subroutine Find_Hadron_Cluster

Take TPC charged track and find next to the nearest hits to the helix model of track (first shell).

• Initialize COMMONS.

⇒ Loop along TPC tracks at the calorimeter face.

• Loops along ECAL and HCAL hits to select and collect hits at the two cell distances from predicted helix for ECAL and three cell distances for HCAL.

• Hits analysis along the track.

• The case when it was not found any hits around the core (at the first shell), the core energy is less then input energy and the input energy is more then 5 GeV. Such a case has unknown nature and core cluster is voluntary replaced with exact energy if it is less then 3 GeV.

• Next case: the energy collected at the first shell plus track core is in $\pm 20\%/\sqrt{E}$ range around input energy. All hits are getting the hard assignment to this track.

• Next case is the hypothesis when the hadron track has a MIP in every layer of ECAL (track has no hadron interaction in ECAL), then energy has to be in $\pm 30\%/\sqrt{E}$ range. It has to be exactly muon-like energy deposition in ECAL with almost all energy deposition in HCAL. All hits are getting the hard assignment to this track.

• If a new ECAL hit collection energy is less then 0.8 of ECAL–MIP (= 1 GeV for now, taking into account the Landau tail) or no one ECAL hit was added. This means that the track was alone, there is not any gamma shower around it and three electronic cell sizes around helix is enough to collect all HCAL energy. Then energy has to be in $\pm 30\%/\sqrt{E}$ range. All hits are getting the hard assignment to this track.

•

•

⇒ Next TPC tracks at the calorimeter face.

5.6.4 Subroutine Find_Hadron_Clust_3

Find HADRON Cluster

THIRD ITERATION

5.6.5 Subroutine ECAL_Reconst

Program logic
Super-Cluster ===¿ similar to individual Jet
Cluster ===¿ similar to hadronic/(particle) shower
Sub-Cluster ===¿ similar to one hadronic shower sub-clusters
THE ASSUMPTION –¿ ONE SUPER-CLUSTER EXIST WHATEVER

Find Super-Clusters
CALL FindEC_Sup_Clust

Find Clusters in Super-Clusters
CALL FindEC_Clust(I)

Cluster Axis fit and first definition of properties
call Eclust_axsis(i,j)

Join hits to Clusters
CALL EC_HIT_JOIN

Re-Calculate of new property of cluster
CALL Eclust_property(i,0)

Join Clusters
CALL EC_CLUST_JOIN

Re-Calculate of new property of cluster
CALL Eclust_property(i,0)

Second Join hits to Clusters
CALL EC_HIT_JOIN

Re-Calculate of new property of cluster
CALL Eclust_property(i,0)

Second Join Clusters
CALL EC_CLUST_JOIN

Re-Calculate of new property of cluster
CALL Eclust_property(i,0)

Final property calculation
CALL Eclust_property(i,1)

5.6.6 Subroutine Findec_Sup_Clust

to find Super-Clusters

* To reduce an amount of information ZOOM method
is used in this program.
Working histogram has cells 5x5 degree now.
!!! tuning parameter !!!
it is filled with all event hits in preparation program.

0. Find 50 largest amplitudes in working histogram
1. Find the largest amplitude in 2-dim space Theta–Phi from 50-s
2. Find all hits at the defined circle around the maximal one
and mark all of them as 1 super-cluster
3. Repeat step 1
So, centers and number of super-clusters are found
4. Collect information about super-clusters
5. Repeat all steps for every cluster
6.

Find all neighbors for every Super-Cluster

For now –¿
Energy of super cluster center is at the a_maxENN(ladr_wE(i))
Theta and Phi th_maxENN(ladr_wE(i)) and ph_maxENN(ladr_wE(i))
Addresses of neighbors are at the Lsup_clsE(i,m),m=1,Isup_clsE(i)
Distances to neighbors are at the cl_dist_wE(i,j)

Assign hits to super-cluster with weight = amplitude of MAX

At the mixing zone – hit randomly belongs to cluster with cluster energy weight.

5.6.7 Subroutine Findec_Clust

The same technique as for Super_Cluster finding.

5.6.8 Subroutine Eclust_Axsis

user routine called from ECAL_RECONST to find 3-d Axes of Clusters, its properties and assignment to particle hypothesis

c Collect the consistence of hits in cell layers
nind0 = 0 ! Number of zero layers
nind1 = 0 ! Number of layers with one hit
nind2 = 0 ! Number of layers with two hits
nindn = 0 ! Number of layers with more then two hits
nindbeg = 0 ! Number of layers have zero at the beginning of ECAL
nindend = 0 ! Number of layers have zero at the end of ECAL
nhit_fst = 0 ! Number of hits at the first 20 layers
nhit_lst = 0 ! Number of hits at the last 20 layers
Calculation of Cluster coordinates and Find the Cluster Axis Find Cluster hit average Density in [MeV/hit] compare it with an average theoretical density for the electromagnetic shower

Choose the hypotheses for cluster

Track model calculation – Helix hypothesis

Copy Cluster information to COMMON Eclusters

5.6.9 Subroutine Ec_Hit_Join

Along Helix hypothesis

5.6.10 Subroutine Eclust_Property

The same as Cluster Axis

5.6.11 Subroutine Ec_Clust_Join

Along Helix hypothesis

5.6.12 Subroutine Eclust_Shower_Hypo

* Find Shower parameters and Chisq for particle hypothesis *
c Track model calculation – Shower Shape hypothesis *

c Coefficient taken from simulation see files
c ga1.dat, el1.dat and logbook p.165-167

5.6.13 Subroutine HCAL_Reconst

to reconstruct HCAL event

CALL FindHB_Sup_Clust

CALL FindHB_Clust(I)

call Hclust_axsis(i,j)

5.6.14 Subroutine Findhb_Sup_Clust

The same technique as FindEC_Sup_Clust.

5.6.15 Subroutine Findhb_Clust

The same technique as FindEC_Clust.

5.6.16 Subroutine Hclust_Axsis

The same technique as Eclust_Axsis.

5.7 Subroutine ECAL_HCAL_Merge

by distance from strait line that is the ECAL cluster axis, taking into account both clusters properties.

5.8 Subroutine ECAL_HCAL_Merge_Again

The same as ECAL_HCAL_Merge.

6 Subroutine IP_Propagate

It reflects the ideal TPC that can recognize all decays, gamma conversion, large scattering angles and multi–loop tracks with small P_t.

Creates four–vectors at the Interaction Point using TPC and cluster information; optionally.

Program is under development.

7 Subroutines related to reconstruction inside the BRAHMS

7.1 BRAHMS geometry and electronic cell databases

Subroutines Ecalgeom, Ecalendcap creates electromagnetic calorimeter geometry in BRAHMS.

Subroutines Hbageom, Hbarbld creates geometry in BRAHMS and electronic geometrical cell database for HCAL barrel.

Subroutines Hcapgeom, Hcapbld, Hcap_cell_db creates geometry in BRAHMS and electronic geometrical cell database for HCAL end-cap.

7.2 Hit collection packing and unpacking

GEANT hit and detector packages are not used due to the huge amount of the detector cells – it is no possible (not enough memory) to make HIT and DET databanks for every detector cells. Dynamic creation of ZEBRA banks for every separated cell can be used for this case but it will takes an additional CPU time. So, the special routines were written to collect hit information in the special common blocks.

These routines are called from **GUSTEP** to collect energy deposition information in detector at every GEANT step.

Subroutine Find_ECAL_Cell

Search for the ECAL calorimeter cell address and pack the geometrical keyword. This program called of subroutine **Cellmap** (was written in Ecole-Politechnik) for ECAL barrel to find the geometrical indexes. Put a new cell amplitude into the common.

Subroutine Find_HBAR_Cell

Search for the HCAL barrel calorimeter cell address and pack the geometrical keyword. Put a new cell amplitude into the common.

Subroutine Find_HCAP_Cell

Search for the HCAL end-cap calorimeter cell address and pack the geometrical keyword. Put a new cell amplitude into the common.

8 Utility routines

File Rec_Util.f

Subroutine Ilin_Fit_3d (rs,xs,ys,zs,cx,cy,cz,nnpt,ind,rpt,xpt,ypt,zpt,wpt,icos,chisq,ierror)
User routine to find parameters for 3–D line through the array of 3–D points with energy as a weight – **cluster axes**.

There are several models can be applied to find the main cluster axes. One of them used 3–D fit to find the line with the smallest χ^2 of distances from that line to all cluster points. Another one used something similar to the trust definition in PYTHIA. *(The programs are available optionally)* Here it was chosen the tensor of inertia model of cluster as solid body. Actually program finds the main tensor axis.

```
*******************************************************************
*     Called old CERNLIB routines to find matrix eigenvalues      *
*******************************************************************
c  rs,xs,ys,zs -- weighted cluster center INPUT  = OUTPUT
c                                - radius and coordinates
c  cx,cy,cz    -- direct cosines  INPUT  = zero approximation
c                                 OUTPUT = result of fit
c   nnpt        -- number of points in fit array
c   ind         -- number of sampling coordinate
c   rpt,xpt,ypt,zpt -- arraies of points to fit - radius and coordinates
c   wpt         -- array of weights of poits
c   icos        -- number of axis the closest to cosines zero approximation
c   chisq       -- chi-sqare OUTPUT
c   ierror      -- .NOT. zero if something wrong during the fit procedure
c----------------------------------------------------------------------
```

SUBROUTINE GKEY_PACK_HBAR
(IMOD,IX,IY,IZ,KEYG)

```
c-----------------------------------------------------------------------
c              Valid for Barrel part of Hadron Calorimeter
c-----------------------------------------------------------------------
c  Input -->
c         IMOD  --> module number 1 -- 32
c         IX    --> X-coordinate cell number in module coordinate system
c         IY    --> Y-coordinate cell number in module coordinate system
c         IZ    --> Z-coordinate cell number in module coordinate system
C  Output  -->  KEYG -- PACKED GEOMETRICAL KEY FOR CALORIMETER HITS
c-----------------------------------------------------------------------
```

SUBROUTINE GKEY_UNPACK_HBAR
(KEYG,IMOD,IX,IY,IZ)

```
c-----------------------------------------------------------------------
c             Valid for Barrel part of Hadron Calorimeter
c-----------------------------------------------------------------------
C  Input  -->  KEYG -- PACKED GEOMETRICAL KEY FOR CALORIMETER HITS
c
c  Output -->
c        IMOD  --> module number 1 -- 32
c        IX    --> X-coordinate cell number in module coordinate system
c        IY    --> Y-coordinate cell number in module coordinate system
c        IZ    --> Z-coordinate cell number in module coordinate system
c-----------------------------------------------------------------------
```

SUBROUTINE GKEY_PACK_HCAP
(K_quad,K_xreg,K_yreg,K_zreg,IX,IY,IZ,KEYG)

```
c-----------------------------------------------------------------------
c             Valid for EndCap part of Hadron Calorimeter
c-----------------------------------------------------------------------
c  Input -->
C        K\_quad--> Quadrant number
C        K\_xreg--> Number of X region of quadrant
C        K\_yreg--> Number of Y region of quadrant
C        K\_zreg--> Number of Z region of quadrant
c        IX    --> X-coordinate cell number in module coordinate system
c        IY    --> Y-coordinate cell number in module coordinate system
c        IZ    --> Z-coordinate cell number in module coordinate system
C  Output  -->  KEYG -- PACKED GEOMETRICAL KEY FOR CALORIMETER HITS
c-----------------------------------------------------------------------
```

SUBROUTINE GKEY_UNPACK_HCAP
(KEYG,K_quad,K_xreg,K_yreg,K_zreg,IX,IY,IZ)

```
c-----------------------------------------------------------------------
c             Valid for EndCap part of Hadron Calorimeter
c-----------------------------------------------------------------------
C  Input  -->  KEYG -- PACKED GEOMETRICAL KEY FOR CALORIMETER HITS
c
c  Output -->
C        K\_quad--> Quadrant number
C        K\_xreg--> Number of X region of quadrant
C        K\_yreg--> Number of Y region of quadrant
C        K\_zreg--> Number of Z region of quadrant
c        IX    --> X-coordinate cell number in module coordinate system
c        IY    --> Y-coordinate cell number in module coordinate system
c        IZ    --> Z-coordinate cell number in module coordinate system
c-----------------------------------------------------------------------
```

Subroutine Mom_Helix_Dist (xp,yp,zp,cx,cy,cz,p,qq,xhi,yhi,zhi,dist)
3–D distance from hit to helix track model with momentum p(cx,cy,cz) at point r(xp,yp,zp)

```
C.    ****************************************************************
C.    *  input: xp,yp,zp particle 3-d position  [cm]                 *
C.    *         cx,cy,cz momentum direction cosines                  *
C.    *         qq,p  - charge and momentum  [GeV/c]                 *
C.    *         xhi,yhi,zhi - HIT  coordinates                       *
C.    *  output:  dist - 3-d distance from helix to hit-point        *
C.    *          in field Bz30 = 3 Tesla                             *
C.    *          in field Bz40 = 4 Tesla                             *
C.    ****************************************************************
```

Method:

Let us write some the simplest definitions:
the distance along Z is $z = \Theta b$ and distance along 2-dim circle is $a = \Theta R$ from these two equation and taken into account that the same point has the same time to reach it one can derive in some approximation:
$t = D\Theta Rm/p_{xy} = D\Theta bm/p_z$; where: m is particle mass:
then we have $b = R(p_z/p_{xy})$

Some definitions:

$$p_{xy} = \sqrt{p_x{}^2 + p_y{}^2}; \quad R = \frac{\sqrt{p_{xy}}}{|q\,B|}; \quad b = \frac{p_z}{|q\,B|};$$

$$x_c = x_p + \frac{p_y}{q\,B}; \quad y_c = y_p - \frac{p_x}{q\,B}$$

$$R^2 + b^2 = \frac{px^2 + py^2}{|q\,B|^2} + \frac{pz^2}{|q\,B|^2}$$

Helix equation with axes along Z and with zero Θ at the point x_p, y_p, z_p exactly:

$$\begin{cases} x = \cos{(q\,\Theta)}(x_p - x_c) - \sin{(q\,\Theta)}(y_p - y_c) + x_c; \\ y = \cos{(q\,\Theta)}(y_p - y_c) + \sin{(q\,\Theta)}(x_p - x_c) + y_c; \\ z = b\,\Theta + z_p; \end{cases}$$

$$dx = \frac{\partial x}{\partial \Theta}; \quad dy = \frac{\partial y}{\partial \Theta}; \quad dz = \frac{\partial z}{\partial \Theta};$$

Equation of normal plane to helix through the hit point:

$$(x_h - x)dx/(R^2 + b^2) + (y_h - y)dy/(R^2 + b^2) + (z_h - z)dz/(R^2 + b^2) = 0$$

After substitutions:

$$\begin{aligned} &-p_y\,q\,Bx_h \sin(q\,\Theta)\,|q^2\,B^2| + p_x\,q\,Bx_h \cos(q\,\Theta)\,|q^2\,B^2| + \\ &p_y\,q\,Bx_p \sin(q\,\Theta)\,|q^2\,B^2| - p_x\,q\,Bx_p \cos(q\,\Theta)\,|q^2\,B^2| + \\ &p_y^2 \sin(q\,\Theta)\,|q^2\,B^2| + p_x\,q\,By_h \sin(q\,\Theta)\,|q^2\,B^2| + \\ &p_y\,q\,By_h \cos(q\,\Theta)\,|q^2\,B^2| - p_x\,q\,By_p \sin(q\,\Theta)\,|q^2\,B^2| - \\ &p_y\,q\,By_p \cos(q\,\Theta)\,|q^2\,B^2| + p_x^2 \sin(q\,\Theta)\,|q^2\,B^2| - \\ &p_z\,q\,B^2 z_h\,|q\,B| + p_z^2 q\,B^2\Theta + p_z\,q\,B^2 z_p\,|q\,B| = 0 \end{aligned}$$

The second order approximation (Tailor raw) of equation of normal plane to helix passes through the hit point near by $\Theta = 0$ is

$$\left(\frac{\left(-\frac{(x_h - x_p)\,p_x}{B} - \frac{(y_h - y_p)\,p_y}{B} + \frac{(z_h - z_p)\,p_z}{|q\,B|}\right) B^2}{{p_x}^2 + {p_y}^2 + {p_z}^2}\right)
+ \left(\frac{\left(\frac{(x_h - x_p)\,p_y}{q\,B} - \frac{(y_h - y_p)\,p_x}{q\,B}\right) B^2}{{p_x}^2 + {p_y}^2 + {p_z}^2} - 1\right)\Theta
+ \frac{\left(\frac{1}{2}\frac{(x_h - x_p)\,p_x}{B} + \frac{\frac{1}{2}(y_h - y_p)\,p_y}{B}\right) B^2\,\Theta^2}{{p_x}^2 + {p_y}^2 + {p_z}^2} = 0$$

The solution of this equation gives the 3–D nearest to the hit point at the helix with rather good accuracy.

Subroutine Mom_Helix_Dist_Array (xp,yp,zp,cx,cy,cz,p,qq,nh,xhi,yhi,zhi,dist)
The same as Mom_Helix_Dist but for array of hits.

Subroutine Helix_dist
(xs,ys,zs,xh,yh,zh,x0,y0,rad,b,phi0,dphi,xm,ym,zm,dist)

```
C.
C.    ******************************************************************
C.    *                                                                *
C.    *      Distance from arbitrary point to  Helix                   *
C.    *     Helix path through the center of coordinate system         *
C.    *                                                                *
C.    *  input: xs,ys,zs - cluster center coordinates                  *
C.    *         cx,cy,cz - direction cosines from cluster fit          *
C.    *         xh,yh,zh - arbitrary poit coordinates                  *
C.    *         x0, y0, rad - circle helix parameter                   *
C.    *         b helix slope                                          *
C.    *         phi0 angle at the helix at 0,0,0 point                 *
C.    *         dphi angle between 0,0,0 and center of cluster         *
C.    *               along circle                                     *
C.    *                                                                *
C.    *   output: xm,ym,zm - point at the helix closests to hit        *
C.    *           dist - distance from helix to hit                    *
C.    *                                                                *
C.    ******************************************************************
```

SUBROUTINE Mom_Helix
(xp,yp,zp,cx,cy,cz,p,q,x0,y0,d0,rad,b,phi0,dphi,z0)

```
C.
C.    ******************************************************************
C.    *                                                                *
C.    *           Helix track model calculation                        *
C.    *  for momentum p(cx,cy,cz) at point r(xp,yp,zp)                 *
C.    *          in field Bz30 = 3 Tesla                               *
C.    *          in field Bz40 = 4 Tesla                               *
C.    *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!*
C.    *      This program has a mistake in the d0 definition           *
C.    *       it was used one more approach d0 lies at Z=0             *
C.    *!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!*
C.    *                                                                *
C.    *                                                                *
C.    *  input: xp,yp,zp particle 3-d position  [cm]                   *
C.    *         cx,cy,cz momentum direction cosines                    *
C.    *         q,p  - charge and momentum  [GeV/c]                    *
C.    *  output: x0, y0, rad - circle helix parameter                  *
C.    *          d0 - P.C.A. distance                                  *
C.    *          b helix slope                                         *
C.    *          phi0 angle at the helix circle at P.C.A.              *
C.    *          dphi angle between P.C.A. and xp,yp,zp                *
C.    *               along circle                                     *
C.    *                                                                *
C.    ******************************************************************
```

SUBROUTINE Clust_Helix
(xs,ys,zs,CX,CY,CZ,x0,y0,rad,b,phi0,dphi,pxy,p)

```
C.
C.    ******************************************************************
C.    *                                                                *
C.    *           Helix track model calculation                        *
C.    *       goes through the xs,ys,zs and cx,cy,cz  and 0,0,0        *
C.    *                                                                *
C.    *       Author : V.L. Morgunov                                   *
C.    *       created  25-Oct-2000  see helix.mws                      *
C.    *                                                                *
C.    *    Helix path through the center of coordinate system          *
C.    *                                                                *
C.    *  input: xs,ys,zs cluster center coordinates                    *
C.    *         cx,cy,cz direction cosines from cluster fit            *
C.    *  output: x0, y0, rad - circle helix parameter                  *
C.    *          b helix slope                                         *
C.    *          phi0 angle at the helix circle at 0,0,0 point         *
C.    *          dphi angle between 0,0,0 and center of cluster        *
```

```
C.    *              along circle                                   *
C.    *          pxy - transversal to field momentum                *
C.    *          p   - derived momentum                             *
C.    *                                                             *
C.    ***************************************************************
```

SUBROUTINE helix_geant_draw
(charge,xp,yp,zp,cx,cy,cz,p,icol)

```
C.    ***************************************************************
C.    *   Calling sequence CALL GHELX3(FIELDM*CHARGE,STEP,VECT,VOUT)   *
C.    *                                                             *
C.    *  Tracking routine in a constant field oriented along axis 3    *
C.    *  Tracking is performed with a conventional helix step method   *
C.    *   input                                                     *
C.    *     STEP = arc length of the step asked                     *
C.    *     VECT = input vector (position,direction cos and momentum)  *
C.    *     CHARGE=  electric charge of the particle                *
C.    *   output                                                    *
C.    *     VOUT = same as VECT after completion of the step        *
C.    *  made from   GHELX3                                         *
c.    *  Authors    R.Brun, M.Hansroul,  Rewriten  V.Perevoztchikov    *
C.    ***************************************************************
C.    ---------------------------------------------------------------
C       units are kgauss,centimeters,gev/c
C.    ---------------------------------------------------------------
```

SUBROUTINE Mhelix_draw
(x0,y0,r,zs,phi0,dphi,icol)

```
C.
C.    ***************************************************************
C.    *                                                             *
C.    *       Draw a helix track model                              *
C.    *       goes through the xs,ys,zs and cx,cy,cz                *
C.    *                                                             *
C.    *       Rewriten from   Author : P.Zanarini                   *
c     *   Input parameters:                                         *
c     *         x0,y0,r  Helix center coordinates and its radius    *
c     *         Zs    -- shift from zero point to cluster center    *
c     *         Phi   -- initial angle to draw                      *
c     *         dPhi  -- angle to draw                              *
c     *   All of them are calculated by Clust\_Helix subroutine     *
C.    ***************************************************************
C.
```

SUBROUTINE Mcircle
(R,x0,y0,z0,icol)

```
C.
C.    *****************************************************************
C.    *       Draw a circle of radius R centered on x0,y0,z0          *
C.    *      Rewriten from (GEANT 321) Author : P.Zanarini            *
C.    *****************************************************************
C.
```

SUBROUTINE draw_cross
(X,Y,Z,SIZSYM,icol)

```
C.    *****************************************************************
C.    *       Draw an HIT point at (X,Y,Z)                            *
c.    *    Was made from GDRHIT (GEANT 321)                           *
C.    *****************************************************************
```

SUBROUTINE draw_hit
(X,Y,Z,SIZSYM,icol)

```
C.    *****************************************************************
C.    *       Draw an HIT point at (X,Y,Z)                            *
c.    *    Was made from GDRHIT (GEANT 321)                           *
C.    *****************************************************************
```

---

subroutine Point_proj
(key,x0,y0,z0,anx,any,anz,n,x,y,z,xx,yy)

```
c-------------------------------------------------------
c   Project of set of n points {x,y,z} to the reference
c  plate and convert its coordinates to the frame at that plate
c-------------------------------------------------------
c   INPUT
c          key = 0 -- calculate transfer matrix
c          key = 1 -- transfer points to the reference plate
c          x0, y0, z0 -- point at reference plate
c          anx,any,anz -- direct vecrtor of reference plate
c          n - array of x,y,z point to convert
c   OUTPUT
c         -- xx,yy,zz resulting array of new coordinates
c-------------------------------------------------------
```

---

subroutine PAIR_TO_GROUP
(npair,ia,ib,ngr,mgrup,kgrup,lgrup)

```
C.**************************************************************
c   INPUT
C     Npair       -- number of track pairs
c     Ia, Ib      -- track addresses lists of close-by pairs
c----------------------------------------------------------------
c   OUTPUT
c     Ngrup       -- number of groups found
c     Mgrup(i)    -- number of members in group
c     Kgrup(i)    -- number of pairs belong to group
c     Lgrup(i,j)  -- Track adresses of group
C.**************************************************************
c   Lgrup has to be exactly equal of the same variable in tpctrack.inc
C.**************************************************************
c     No more then 50 groups and 20 members in any group
c-----------------------------------------------------------
```

Subroutine Point_line
(xp,yp,zp,cx,cy,cz,xv,yv,zv,xm,ym,zm)

```
c-------------------------------------------------------------------
c       Point at line that closest to point near by line
c-------------------------------------------------------------------
c       xp, yp, zp -- point is at the line cx,cy,cz
c       xv, yv, zv -- point that is out of line
c       xm, ym, zm -- closest point at the line
c-------------------------------------------------------------------
```

Function sdist
(xp,yp,zp,cx,cy,cz,xv,yv,zv)

```
c-------------------------------------------------------------------
c        Distance from line to point
c-------------------------------------------------------------------
c       xp, yp, zp -- point is at the line
c       xv, yv, zv -- point is out of line
```

---

subroutine Matrix_sort
(a,n,m,k,i,j)

```
*****************************************************************
*     Utility routine to find K smallest values                 *
*    from N x M values of array A and their addresses           *
*     INPUT  A -- Array with tested numbers                     *
*            N -- first index size in A                         *
*            M -- second indexs size in A                       *
*            K -- number of needed minimal numbers              *
*     OUTPUT I -- list of first indexes in increasing order of value*
*            J -- list of second indexes in increasing order of val.*
*****************************************************************
```

---

subroutine Min_amp
(n,a,m,b,ia)

```
*****************************************************************
*     Utility routine to find M smallest values                 *
*    from N amplitudes array and thier addresses                *
*          Can be used if M << N                                *
*     in opposite case better use SORTZV (CERN-lib)             *
*     INPUT                                                     *
*            N -- initial array size                            *
*            A -- Array with tested numbers                     *
*            M -- Number of needed min numbers                  *
*    OUTPUT                                                     *
*            B -- Sorted array                                  *
*            Ia -- addresses of sorted numbers in initial array *
*****************************************************************
```

---

subroutine Max_amp
(n,a,m,b,ia)

```
*****************************************************************
*     Utility routine to find M biggest values                  *
*    from N amplitudes array and thier addresses                *
*          Can be used if M << N                                *
*     in opposite case better use SORTZV (CERN-lib)             *
*     INPUT  N -- initial array size                            *
*            A -- Array with tested numbers                     *
*            M -- Number of needed min numbers                  *
*            B -- Sorted array                                  *
*            Ia -- addresses of sorted numbers in initial array *
*****************************************************************
```

---

function angle_dist
(t1,p1,t2,p2)

```
******************************************************************
*    Calculate angle between two vectors at sphere               *
* Input                                                          *
*   T(i) and P(i) --  spherical coordinates of vectors [degree]  *
* Output                                                         *
*    angle\_dist -- Angle between vectors  [degree]               *
******************************************************************
```

---

subroutine hit_move_rot

```
(rcyl,rs,xs,ys,zs,cx,cy,cz,xhi,yhi,zhi,xmr,ymr,zmr)
******************************************************************
*     user routine called from Find\_HB\_clust                     *
*  to find points at the cylinder surface around  Super-Clusters *
*  in the coordinate system with axes of Super-Cluster as Z-axes *
c      See MAPLE file cyleq2.mws                                 *
******************************************************************
```

---

subroutine hit_move
(rcyl,rs,xs,ys,zs,cx,cy,cz,xhi,yhi,zhi,xm,ym,zm)

```
******************************************************************
*     user routine called from Find\_HB\_clust                     *
*  to find points at the cylinder surface around  Super-Clusters *
c=========================================================
c      See MAPLE file cyleq2.mws
c  Input parameters
c   rcyl -- Radius of cylinder around Super-Cluster
c   xs,ys,zs -- Center of Super-Cluster
c   rs,cx,cy,cz -- it's radius and Direct COSines
c   xhi,yhi,zhi -- Hit coordinates
c  Output parameters
c   xm,ym,zm -- Projection of hit to the cylinder surface
c               around Super-Cluster
c=========================================================
      rr = 1.0/rs
c  Coordinate of point at the Super-Cluster axes
c     that is perpendicular to Hit point
```

---

subroutine hit_rot
(cx,cy,cz,xm,ym,zm,xmr,ymr,zmr)

```
*******************************************************************
*     user routine called from Find\_HB\_clust                        *
*  to find points in the coordinate system                          *
*     with axes of Super-Cluster as Z-axes                          *
c      See MAPLE file cyleq2.mws
c  Input parameters
c   Rotation matrix to the coordinate system
c     with axes of Super-Cluster as Z-axes
c==========================================================
c  Direction of S-Cl axes become to be +Z' axes
c    and X' axes has a + signum in old coordinate system
c   if is(cy>0.0) then e11:=cy/sqrt(cx^2+cy^2);
c                 else e11:=-cy/sqrt(cx^2+cy^2); fi;
c   if is(cy>0.0) then e12:=-cx/sqrt(cx^2+cy^2);
c                  else e12:=cx/sqrt(cx^2+cy^2); fi;
c   e13 := 0;
c   e21 := e13*e32-e12*e33;
c   e22 :=  e11*e33-e13*e31;
c   e23 := e12*e31-e11*e32;
c   e31 := cx;
c   e32 := cy;
c   e33 := cz;
c    Rotation matrix is inverse to defined above
c==========================================================
```

---

function coosh
(x,step)

```
*******************************************************************
*  Shift coordinate to the center of cell with width = STEP        *
*******************************************************************
```

SUBROUTINE FIT_LONG_SHOWER
(n,x,y,a,b,c,chi)

```
c==========================================================================
c               Utility routine
c       Chi--square method for calculation of
c       Longitudilal Electromagnetic Shower model
c       parameters A,B,C in E(x) = C*(X^A)*exp(-B*X)
c           See MAPLE file shower.mws
```

```
c       N    --- number of points
c       X(i) --- distance along shower
c       Y(i) --- amplitudes along shower
c       W(i) --- weight
c       Chi  --- Chi-square in the histogram space but not the logarithmic one
c        it is some model dependent chisqare with distribution around unit
c==========================================================================
```

SUBROUTINE FIT_LINEAR_S
(n,x,y,s,a,b)

```
c==========================================================================
c               Utility routine
c       Chi--square method for calculation of
c       parameters A,B in linear equation Y(x) = A*X + B
c           See MAPLE file linear.mws
c       N    --- number of points
c       X(i) --- distance along shower
c       Y(i) --- amplitudes along shower
c       S(i) --- error of amplitude
c==========================================================================
*   Here have to be added -- correct zero amplitudes account
*      and   chi-square calculations for fitted curve
c==========================================================================
```

SUBROUTINE FIT_LINEAR_W
(n,x,y,w,a,b)

```
c==========================================================================
c               Utility routine
c       Chi--square method for calculation of
c       parameters A,B in linear equation Y(x) = A*X + B
c           See MAPLE file linear\_w.mws
c       N          ---  number     of points
c       X(i), Y(i) ---  coordinate of point
c       W(i)       ---  weight     of point
c==========================================================================
*   Here have to be added -- correct zero amplitudes account
*      and   chi-square calculations for fitted curve
c==========================================================================
```

SUBROUTINE FIT_EXP
(n,x,y,s,a,b)

```
c===========================================================================
c               Utility routine
c       Chi--square method for calculation of
c       parameters A,B in equation Y(x) = A*exp(-B*X)
c       Or Transversal Electromagnetic Shower model
c           See MAPLE file linear.mws
c       N    --- number of points
c       X(i) --- distance along shower
c       Y(i) --- amplitudes along shower
c       S(i) --- error of amplitude
c===========================================================================
*   Here have to be added -- correct zero amplitudes account
*      and   chi-square calculations for fitted curve
c===========================================================================
```

SUBROUTINE SDIRCO
(X1,Y1,Z1,X2,Y2,Z2,CX,CY,CZ)

```
c------------------------------------------------------------------------
C          Line equation THROUGH THE TWO POINTS
c------------------------------------------------------------------------
```

9 Event analysis, Jet Finder and Mass Reconstruction

Optional part of reconstruction program:
Subroutine RECONST_ANALYZ
Subroutine CALOR_ANALYZ

10 Event Printing and Drawing

Optional part of reconstruction program:
Subroutine CALOR_DRAW
Subroutine Eclust_Draw
Subroutine Hclust_Draw

11 Common blocks description

COMMON / RECONPARAM / fill up in subroutine CALOR_RECONST_INI:

```
C ----------------------------------------------------------------------
      real*4  energy_coeff1, energy_coeff2, energy_coeff
      real*4  sph_cutE_sup, coeff_radE_sup
      real*4  sph_cutE_clu, coeff_radE_clu
      real*4  sph_cut_sup,  coeff_rad_sup
      real*4  sph_cut_clu,  coeff_rad_clu
      real*4  Cl_Dist_Merge, Dist_Merge
      real*4  EC_min_mu, EC_max_mu
      real*4  EM_Energy, Cl_Dist_Merge_EM
      real*4  x0_1ECAL, x0_2ECAL
      real*4  R_moliere, theor_mu_densE, theor_mu_dens
      real*4  thresh_clust, thresh_hits
      real*4  TPC_trk_Dist_Merge
      real*4  TPC_Cl_Dist_Merge, TPC_Dist_Merge
      real*4  TPC_Cl_Dist_MergeH, TPC_Dist_MergeH
      real*4  Energy_sigmaE, Energy_sigmaH
      real*4  Esup_clustE_thr,  Esup_clust_thr
      integer nhit_thrE, nhit_thr, nindbeg_thr, nindend_thr
      common/reconparam/ energy_coeff1, energy_coeff2, energy_coeff
     $, sph_cutE_sup, coeff_radE_sup
     $, sph_cutE_clu, coeff_radE_clu
     $, sph_cut_sup,  coeff_rad_sup
     $, sph_cut_clu,  coeff_rad_clu
     $, Cl_Dist_Merge, Dist_Merge
     $, EC_min_mu, EC_max_mu
     $, EM_Energy, Cl_Dist_Merge_EM
     $, x0_1ECAL, x0_2ECAL
     $, R_moliere, theor_mu_densE, theor_mu_dens
     $, nindbeg_thr, nindend_thr
     $, thresh_clust, thresh_hits
     $, nhit_thrE, Esup_clustE_thr
     $, nhit_thr, Esup_clust_thr
     $, TPC_trk_Dist_Merge
     $, TPC_Cl_Dist_Merge, TPC_Dist_Merge
     $, TPC_Cl_Dist_MergeH, TPC_Dist_MergeH
     $, Energy_sigmaE, Energy_sigmaH
C ----------------------------------------------------------------------
      SAVE /RECONPARAM/
*************************************************************************
```

COMMON / CALOINPUT /
consists of TPC reconstruction information – can be filled with reconstructed TPC information in subroutine **TPC_Reconst_Prep**.

NTRKTPC – maximal allowed array size,
ninp1 – number of tracks at IP,
lpart1 – particle ID at IP,
lvert1 – vertex number (GEANT),
xmass1 – particle mass,

xcharge1 – particle charge,
xc1, yc1, zc1 – vertex coordinates,
cx1, cy1, cz1 – momentum direct cosines,
pm1 – particle momentum,
ninp2, lpart2, xmass2, xcharge2, xc2, yc2, zc2, cx2, cy2, cz2, pm2 – the same track parameters but at the calorimeter face, except,
lvert2 – address of mother particle at IP.
XHTPC, YHTPC, RHTPC, BHTPC, RHTPC, BHTPC, PHTPC, DPHTPC, ZHTPC, DHTPC – parameters of track helix hypothesis prediction (see subroutine **Mom_Helix** in **rec_util.f**).

```
C ---------------------------------------------------------------------
      integer NTRKTPC
      integer ninp0,lpart0,lvert0
      integer ninp1,lpart1,lvert1
      integer ninp2,lpart2,lvert2
C ---------------------------------------------------------------------
      real*4 xmass0,xcharge0,xc0,yc0,zc0,cx0,cy0,cz0,pm0
      real*4 xmass1,xcharge1,xc1,yc1,zc1,cx1,cy1,cz1,pm1
      real*4 xmass2,xcharge2,xc2,yc2,zc2,cx2,cy2,cz2,pm2
      real*4 XHTPC,YHTPC,RHTPC,BHTPC,PHTPC,DPHTPC,ZHTPC,DHTPC
C ---------------------------------------------------------------------
      PARAMETER (NTRKTPC=400)
C ---------------------------------------------------------------------
      common /caloinput/
     $  ninp0,lpart0(NTRKTPC),lvert0(NTRKTPC)
     $, xmass0(NTRKTPC),xcharge0(NTRKTPC)
     $, xc0(NTRKTPC),yc0(NTRKTPC),zc0(NTRKTPC)
     $, cx0(NTRKTPC),cy0(NTRKTPC),cz0(NTRKTPC),pm0(NTRKTPC)
     $, ninp1,lpart1(NTRKTPC),lvert1(NTRKTPC)
     $, xmass1(NTRKTPC),xcharge1(NTRKTPC)
     $, xc1(NTRKTPC),yc1(NTRKTPC),zc1(NTRKTPC)
     $, cx1(NTRKTPC),cy1(NTRKTPC),cz1(NTRKTPC),pm1(NTRKTPC)
     $, ninp2,lpart2(NTRKTPC),lvert2(NTRKTPC)
     $, xmass2(NTRKTPC),xcharge2(NTRKTPC)
     $, xc2(NTRKTPC),yc2(NTRKTPC),zc2(NTRKTPC)
     $, cx2(NTRKTPC),cy2(NTRKTPC),cz2(NTRKTPC),pm2(NTRKTPC)
     $, XHTPC(NTRKTPC),  YHTPC(NTRKTPC),  RHTPC(NTRKTPC)
     $, BHTPC(NTRKTPC),  PHTPC(NTRKTPC), DPHTPC(NTRKTPC)
     $, ZHTPC(NTRKTPC),  DHTPC(NTRKTPC)
C ---------------------------------------------------------------------
        save /caloinput/
C     documentation for CALOINPUT commong block

C     V.L. Morgunov
C     MC information for the event
C
c     This part of common is filled in the GUKINE
```

```
C     ninp0          number of MC tree particles
C     lpart0(i)         particle ID code for this particle
C     lvert0(i)         Address of particle in HEP record
C     xmass0(i)         MC mass of particle
C     xcharge0(i)       charge of particle
C     xc0(i)            vertex(1) of particle
C     yc0(i)            vertex(2)
C     zc0(i)            vertex(3)
C     cx0(i)            px/p
C     cy0(i)            py/p
C     cz0(i)            pz/p
C     pm0(i)            p      all momenta are tree momenta
C
c     This part of common is filled in the TK?????

C     charged track information at the point of closest approach to
C     the IP
C     ninp1         number of charged tracks (equals to number of TK banks)
C     lpart1(i)         particle ID (if MC)
C     lvert1(i)         MC vertex number
C     xmass1(i)         mass of particle
C     xcharge1(i)        charge (-1., 0., 1., or 10. if unknown)
C     xc1(i)            -d0 * sin(phi)
C     yc1(i)             d0 * cos(phi)
C     zc1(i)             z0
C     pm1(i)             abs(p)
C     cx1(i)             sin(theta) * cos(phi)
C     cy1(i)             sin(theta) * sin(phi)
C     cz1(i)             cos(theta)

C     charged track information at the face of the calorimeter

c     This part of common is filled in the TK?????

C     the information from the previous block is repeated, but moved to
C     the calorimeter face.
C     ninp2         number of charged tracks (equals to number of TK banks)
C     lpart2(i)         particle ID (if MC)
C     lvert2(i)         MC vertex number
C     xmass2(i)         mass of particle
C     xcharge2(i)        charge (-1., 0., 1., or 10. if unknown)
c             Coordinates at the Calorimeter face
C     xc2(i)            -d0 * sin(phi)
C     yc2(i)             d0 * cos(phi)
C     zc2(i)             z0
c             Momentum at the Calorimeter face
C     pm2(i)             abs(p)
C     cx2(i)             sin(theta) * cos(phi)
C     cy2(i)             sin(theta) * sin(phi)
```

```
C     cz2(i)             cos(theta)
```

COMMON / TPCTRACK /

```
cd    +KEEP,TPCTRACK.
*******************************************************************
*   This common collects an information about track grops
*            at the calorimeter face
*******************************************************************
C ----------------------------------------------------------------------
      integer ind_pm0,ind_pm1,ind_pm2,kk_tpc,ktpc1,ktpc2
      integer ngr,mgrup,kgrup,lgrup
      integer  n_tr_gr,la_tr_gr,n_tr_not_gr,la_tr_not_gr
C ----------------------------------------------------------------------
      real*4 TPC_distance,xgr,ygr,zgr,cxgr,cygr,czgr,grmom
C ----------------------------------------------------------------------
      Common /TPCTRACK/ ind_pm0(ntrktpc+1),ind_pm1(ntrktpc+1),
     &  ind_pm2(ntrktpc+1)
     $, kk_tpc, TPC_distance(ntrktpc,ntrktpc)
     $, ktpc1(500), ktpc2(500)
     $, ngr, mgrup(100), kgrup(100), lgrup(100,50)
     $, xgr(100),ygr(100),zgr(100),cxgr(100)
     $, cygr(100),czgr(100),grmom(100)
     $, n_tr_gr,la_tr_gr(250), n_tr_not_gr,la_tr_not_gr(250)
C ----------------------------------------------------------------------
        save /tpctrack/
*************************************************************************
*               Filled in tpc_reconst_prep.f
*************************************************************************
*  ind_pm*(ninp*)  -- index for /caloinput/ COMMON for track
*                     in decreasing momentum order
* -----------------------------------------------------------------------
*  kk_tpc              -- number of rasonable distances in matrix
*  TPC_distance(k1,k2) -- collect kk_tpc distances of pairs of tracks
*                         with distance less then TPC_trk_Dist_Merge
*      ktpc1(kk_tpc)   --  first index of TPC_distance matrix
*                                  in increasing distance order
*      ktpc2(kk_tpc)   --  second index of TPC_distance matrix
*                                  in increasing distance order
* -----------------------------------------------------------------------
*   ngr  --  number of group of tracks at the calorimeter face
*   mgrup(k) -- number of members in group k in array lgrup(k,i)
*   kgrup(k) -- number of close-by pairs in group k
*   lgrup(k,i) -- addresses of i-track in the /caloinput/ common
*                 that belongs to k-group
*******************************************************************
*   lgrup has to be equivalent to Lgrup in Pair_to_group parameter value
*******************************************************************
*   xgr,ygr,zgr -- coordinates of group center
```

```
*   cgr,cgr,cgr -- direction of group of tracks
*   grmom       -- momentum of group of tracks
*   n_tr_gr, la_tr_gr() -- List of tracks belong to any group
*   n_tr_not_gr,la_tr_not_gr()-- List of tracks do not belong to any group
**************************************************************************
```

INCLUDE caloampl.inc

There are several common blocks consist of geometrical keywords and raw amplitudes (hits) for all calorimeters and it's parts. Used in **ECAL_RECONST_PREP** and **HCAL_RECONST_PREP** to fill **COMMON / ECRECONST /** and **/ HCRECONST /** with starting reconstruction amplitudes and coordinates.

All commons are filled during GEANT simulation run or from reading the event file if the simulation was in separate run.

COMMONs

/myBec1/ – ECAL barrel first sampling structure – 1.4 mm W,
/myBec1/ – ECAL barrel second sampling structure – 4.2 mm W,
/myBhca/ – HCAL barrel sampling structure
/myEPec1/ – ECAL endcap +Z first sampling structure – 1.4 mm W,
/myEPec2/ – ECAL endcap +Z second sampling structure – 4.2 mm W,
/myEPhca/ – HCAL endcap +Z sampling structure
/myEMec1/ – ECAL endcap -Z first sampling structure – 1.4 mm W,
/myEMec2/ – ECAL endcap -Z second sampling structure – 4.2 mm W,
/myEMhca/ – HCAL endcap -Z sampling structure

Where:
MBEC1 – number of stored words in common, **KBEC1** or **BBEC1** – geometrical keyword or hit amplitude

Subroutine Clearcomm has to be called before reading or generation of next event for hits common blocks cleaning. This subroutine also sets hits counters to initial value.

```
C ---------------------------------------------------------------------
C                 BARREL
C ---------------------------------------------------------------------
      PARAMETER (LBEC1=40000)
C ---------------------------------------------------------------------
      common/myBec1/MBEC1, BBEC1(LBEC1)
      DIMENSION   KBEC1(LBEC1)
      equivalence(KBEC1(1),BBEC1(1))
        save /myBec1/
C ---------------------------------------------------------------------
      PARAMETER (LBEC2=10000)
C ---------------------------------------------------------------------
      common/myBec2/MBEC2, BBEC2(LBEC2)
      DIMENSION   KBEC2(LBEC2)
      equivalence(KBEC2(1),BBEC2(1))
        save /myBec2/
C ---------------------------------------------------------------------
      PARAMETER (LBHCA=10000)
C ---------------------------------------------------------------------
```

```
      common/myBhca/MBHCA, BBHCA(LBHCA)
      DIMENSION   KBHCA(LBHCA)
      equivalence(KBHCA(1),BBHCA(1))
        save /myBhca/
C ---------------------------------------------------------------------
C                 ENDCAP Z+
C ---------------------------------------------------------------------
      PARAMETER (LEPEC1=40000)
C ---------------------------------------------------------------------
      common/myEPec1/MEPEC1,BEPEC1(LEPEC1)
      DIMENSION   KEPEC1(LEPEC1)
      equivalence(KEPEC1(1),BEPEC1(1))
        save /myEPec1/
C ---------------------------------------------------------------------
      PARAMETER (LEPEC2=10000)
C ---------------------------------------------------------------------
      common/myEPec2/MEPEC2,BEPEC2(LEPEC2)
      DIMENSION   KEPEC2(LEPEC2)
      equivalence(KEPEC2(1),BEPEC2(1))
        save /myEPec2/
C ---------------------------------------------------------------------
      PARAMETER (LEPHCA=10000)
C ---------------------------------------------------------------------
      common/myEPhca/MEPHCA,BEPHCA(LEPHCA)
      DIMENSION   KEPHCA(LEPHCA)
      equivalence(KEPHCA(1),BEPHCA(1))
        save /myEPhca/
C ---------------------------------------------------------------------
C                 ENDCAP Z-
C ---------------------------------------------------------------------
      PARAMETER (LEMEC1=40000)
C ---------------------------------------------------------------------
      common/myEMec1/MEMEC1,BEMEC1(LEMEC1)
      DIMENSION KEMEC1(LEMEC1)
      equivalence(KEMEC1(1),BEMEC1(1))
        save /myEMec1/
C ---------------------------------------------------------------------
      PARAMETER (LEMEC2=10000)
C ---------------------------------------------------------------------
      common/myEMec2/MEMEC2,BEMEC2(LEMEC2)
      DIMENSION KEMEC2(LEMEC2)
      equivalence(KEMEC2(1),BEMEC2(1))
        save /myEMec2/
C ---------------------------------------------------------------------
      PARAMETER (LEMHCA=10000)
C ---------------------------------------------------------------------
      common/myEMhca/MEMHCA,BEMHCA(LEMHCA)
      DIMENSION KEMHCA(LEMHCA)
      equivalence(KEMHCA(1),BEMHCA(1))
```

```
      save /myEMhca/
C ----------------------------------------------------------------------
```

COMMON / TRACKCORE /

```
+KEEP,TRACKCORE.
c======================================================================
c          Core track information collection
c======================================================================
c      Arrayes index is track number in main TPC common
c======================================================================
       integer NTRCKCORE
C ----------------------------------------------------------------------
      PARAMETER (NTRCKCORE=400)
C ----------------------------------------------------------------------
       real  chisq,chisqe,chisqh
       real  enrgy,enrgye,enrgyh
       real  enr,enre,enrh
       real  enr2,enre2,enrh2
       real  enr3,enre3,enrh3
       real  e_loss_ave_e,e_loss_ave_h
       real  e_loss_ave_e2,e_loss_ave_h2
       real  e_loss_ave_e3,e_loss_ave_h3
       real  e_sume_1,e_sume_2
       integer n_used_track
       integer nhit_ecal,nhit_hcal
       integer nhit_ecal2,nhit_hcal2
       integer nhit_ecal3,nhit_hcal3
       integer i_no_first_hit
       integer i_last_lay_e,i_no_cont_e,i_no_cont_ea
       integer i_last_lay_h,i_no_cont_h,i_no_cont_ha
       integer i_track_start_e,i_track_start_h
       integer n_lay_mips_e,lay_mip_e,lay_mip_ea
       integer n_lay_mips_h,lay_mip_h,lay_mip_ha
       integer n_hite_1,n_hite_2
c----------------------------------------------------------------------
       common /trackcore/ n_used_track(NTRCKCORE)
     $, enr(NTRCKCORE),enre(NTRCKCORE),enrh(NTRCKCORE)
     $, enr2(NTRCKCORE),enre2(NTRCKCORE),enrh2(NTRCKCORE)
     $, enr3(NTRCKCORE),enre3(NTRCKCORE),enrh3(NTRCKCORE)
     $, enrgy(NTRCKCORE),enrgye(NTRCKCORE),enrgyh(NTRCKCORE)
     $, e_loss_ave_e(NTRCKCORE),e_loss_ave_h(NTRCKCORE)
     $, e_loss_ave_e2(NTRCKCORE),e_loss_ave_h2(NTRCKCORE)
     $, e_loss_ave_e3(NTRCKCORE),e_loss_ave_h3(NTRCKCORE)
     $, nhit_ecal(NTRCKCORE),nhit_hcal(NTRCKCORE)
     $, nhit_ecal2(NTRCKCORE),nhit_hcal2(NTRCKCORE)
     $, nhit_ecal3(NTRCKCORE),nhit_hcal3(NTRCKCORE)
     $, chisq(NTRCKCORE),chisqe(NTRCKCORE),chisqh(NTRCKCORE)
```

```
     $, i_no_first_hit(NTRCKCORE),i_no_cont_ea(NTRCKCORE)
     $, i_last_lay_e(NTRCKCORE),i_no_cont_e(NTRCKCORE)
     $, i_last_lay_h(NTRCKCORE),i_no_cont_h(NTRCKCORE)
     $, i_no_cont_ha(NTRCKCORE),lay_mip_ea(NTRCKCORE)
     $, i_track_start_e(NTRCKCORE),i_track_start_h(NTRCKCORE)
     $, n_lay_mips_e(NTRCKCORE),lay_mip_e(NTRCKCORE)
     $, n_lay_mips_h(NTRCKCORE),lay_mip_h(NTRCKCORE)
     $, lay_mip_ha(NTRCKCORE)
     $, e_sume_1(NTRCKCORE),e_sume_2(NTRCKCORE)
     $, n_hite_1(NTRCKCORE),n_hite_2(NTRCKCORE)
c-----------------------------------------------------------------------
       save /trackcore/
c-----------------------------------------------------------------------
c         Common description
c-----------------------------------------------------------------------
*     n_used_track  - Number of logical branch this track treated in
c        At the core collection step
*     enr           - Energy of cluster collected along track in calorimeter
*     enre          - Energy of cluster collected along track in ECAL
*     enrh          - Energy of cluster collected along track in HCAL
c        At the second iteration
*     enr2          - Energy of cluster collected along track in calorimeter
*     enre2         - Energy of cluster collected along track in ECAL
*     enrh2         - Energy of cluster collected along track in HCAL
c        At the third iteration
*     enr3          - Energy of cluster collected along track in calorimeter
*     enre3         - Energy of cluster collected along track in ECAL
*     enrh3         - Energy of cluster collected along track in HCAL
c        After all iterations
*     enrgy         - Energy of cluster collected along track in calorimeter
*     enrgye        - Energy of cluster collected along track in ECAL
*     enrgyh        - Energy of cluster collected along track in HCAL
c
*     e_loss_ave_e  -
*     e_loss_ave_h  -
*     e_loss_ave_e2 -
*     e_loss_ave_h2 -
*     e_loss_ave_e3 -
*     e_loss_ave_h3 -
*     nhit_ecal     -
*     nhit_hcal     -
*     nhit_ecal2    -
*     nhit_hcal2    -
*     nhit_ecal3    -
*     nhit_hcal3    -
*     chisq         -
*     chisqe        -
*     chisqh        -
*     i_no_first_hit -
```

```
*     i_no_cont_ea  -
*     i_last_lay_e  -
*     i_no_cont_e   -
*     i_last_lay_h  -
*     i_no_cont_h   -
*     i_no_cont_ha  -
*     lay_mip_ea    -
*     i_track_start_e -
*     i_track_start_h -
*     n_lay_mips_e  -
*     lay_mip_e     -
*     n_lay_mips_h  -
*     lay_mip_h     -
*     lay_mip_ha    -
*     e_sume_1      -
*     e_sume_2      -
*     n_hite_1      -
*     n_hite_2      -
c=======================================================================
```

COMMON / ECLUSTERS /

```
+KEEP,ECLUSTERS.
**************************************************************************
*     The main common for neutral cluster collection in ECAL
**************************************************************************
      integer NECLUST,NEclusters,LESUP,LECLU
      integer NEHITS,NEHITS1,NEHITS2
      integer NEHITSO,NEBEG,NEEND,IHYPECL
      integer McluH
C ---------------------------------------------------------------------
      real*4 ENECL,ECLCHISQ,DNECL,EN1ECL,EN2ECL
      real*4 XECL_fst,YECL_fst,ZECL_fst
      real*4 XECL,YECL,ZECL,CXECL,CYECL,CZECL
      real*4 RECL,TECL,PECL
C ---------------------------------------------------------------------
      PARAMETER (NECLUST=500)
C ---------------------------------------------------------------------
       common/Eclusters/ NEclusters,LESUP(NECLUST),LECLU(NECLUST)
     $,NEHITS(NECLUST),NEHITS1(NECLUST),NEHITS2(NECLUST)
     $,NEHITSO(NECLUST), NEBEG(NECLUST),  NEEND(NECLUST)
     $,IHYPECL(NECLUST),  ENECL(NECLUST),ECLCHISQ(NECLUST)
     $, DNECL(NECLUST), EN1ECL(NECLUST) ,EN2ECL(NECLUST)
     $,  XECL_fst(NECLUST),YECL_fst(NECLUST),ZECL_fst(NECLUST)
     $,  XECL(NECLUST),   YECL(NECLUST),   ZECL(NECLUST)
     $, CXECL(NECLUST),  CYECL(NECLUST),  CZECL(NECLUST)
     $,  RECL(NECLUST),   TECL(NECLUST),   PECL(NECLUST)
     $, McluH(NECLUST)
```

```
C ---------------------------------------------------------------------
      save /ECLUSTERS/
************************************************************************
*      ECAL  Neutral Clusters Common Block description
C ---------------------------------------------------------------------
*  NEclusters - Number of ECAL neutral clusters collected at this common
*  LESUP      - pointer to Super-cluster address in the COMMON/ECrecres/
*  LECLU      - pointer to Cluster address in the COMMON/ECrecres/
*  NEHITS     - Number of hits belongs to this cluster
*  NEHITSO    - Number of hits belongs to this cluster (working variable)
*  NEBEG      - Number of layers have zero at the begining of ECAL
*  NEEND      - Number of layers have zero at the end of ECAL
*  IHYPECL    - Hypothesis number for this cluster,
*                 see COMMON/caloevent/ for the hypothesis description
*  ENECL      - Cluster energy
*  ECLCHISQ   - Chi^2 from the inetria tensor definition (Sum of squared
*               distances of hits from the main principal axes)
*  DNECL      - Cluster average density [MeV/hit]
*---------------------------------------------------------------------
*  XECL,YECL,ZECL - cluster center of gravity
*  CXECL,CYECL,CZECL - direct Cosine for the principal axis
*  RECL       - 3-D radius of center of gravity of cluster
*  TECL       - Theta angle of center of gravity of cluster
*  PECL       - Phi angle of center of gravity of cluster
* ---------------------------------------------------------------------
*  NEHITS1    - Number of hits belongs to this cluster is at the
*                 first part of ECAL (1 - 10 layers)
*  EN1ECL     - Cluster energy in the first part of ECAL (1 - 10 layers)
*  XECL_fst, YECL_fst,ZECL_fst  - Center of gravity of the first part
*                                  of ECAL (1 - 10 layers)
*  NEHITS2    - Number of hits belongs to this cluster is at the
*                 last part of the ECAL (11 - 40 layers)
*  EN2ECL     - Cluster energy in the second part of ECAL (11 - 40 layers)
* ---------------------------------------------------------------------
*  McluH      - HCAL cluster address if merged exists (= 0 if not exists)
************************************************************************
```

COMMON / ECRECONST /

```
+KEEP,ECRECONST.
*************************************************************************
*          Main common for ECAL hit collection
*************************************************************************
      integer NLADE,L_cellE,ny_layE,Lsup_clustE,L_clustE,Lsub_clustE
      integer NHITSE,L_ch_trackE
C ---------------------------------------------------------------------
      real*4 a_cellE,x_cellE,y_cellE,z_cellE,r_cellE,t_cellE,p_cellE
      real*4 summ_barrel_ecal1,summ_barrel_ecal2,summ_barrel_ecal
      real*4 summ_ECAPP_ecal1, summ_ECAPP_ecal2, summ_ECAPP_ecal
```

```
      real*4 summ_ECAPM_ecal1, summ_ECAPM_ecal2, summ_ECAPM_ecal
      real*4 summ_ECAL
C ------------------------------------------------------------------------
      PARAMETER (NLADE=30000)
C ------------------------------------------------------------------------
      COMMON/ECreconst/
     $       L_cellE(nladE), a_cellE(nladE),    ny_layE(nladE)
     $ ,     x_cellE(nladE), y_cellE(nladE),    z_cellE(nladE)
     $ ,     r_cellE(nladE), t_cellE(nladE),    p_cellE(nladE)
     $ , Lsup_clustE(nladE),L_clustE(nladE),L_ch_trackE(nladE)
     $ , NHITSE,summ_barrel_ecal1,summ_barrel_ecal2,summ_barrel_ecal
     $ ,         summ_ECAPP_ecal1, summ_ECAPP_ecal2, summ_ECAPP_ecal
     $ ,         summ_ECAPM_ecal1, summ_ECAPM_ecal2, summ_ECAPM_ecal
     $ ,         Summ_ECAL
C ------------------------------------------------------------------------
       save /ecreconst/
c===========================================================================
c          Description of COMMON /ECreconst/
c               HIT/CELL INFORMATION
c===========================================================================
c    NhitsE --------- number of hits in this common and in ECAL
c===========================================================================
c    summ_barrel_ecal1 --- Whole Energy deposited in ECAL-barrel 1 struct
c    summ_barrel_ecal2 --- Whole Energy deposited in ECAL-barrel 2 struct
c    ....... and so on
c---------------------------------------------------------------------------
c        L_cellE -- Cell Status word collect all information
c                       related to cell assignment
c   L_cellE =
c         i x 1 -- Super_Cluster number
c              +
c       i x 100 -- Cluster number in super-cluster or individual cluster
c
c          if L_cellE = 0 then  -- it is an individual hit
c             (no one reconstruction or recognition program assigned it.
c---------------------------------------------------------------------------
c    L_ch_trackE - charged track address this hit belongs to (=0 if not)
c---------------------------------------------------------------------------
c    a_cellE -------- Amplitude of hit (cell)
c    ny_layE -------- Absorber layer number of hit (cell)
c    x,y,z_cellE ---- Cartezian coordinates of hit (cell)
c    r,t,p_cellE ---- Spherical coordinates of hit (cell)
c---------------------------------------------------------------------------
c          Working arraies
c---------------------------------------------------------------------------
c    Lsup_clustE ---- super-cluster address this hit (cell) belongs to
c    L_clustE ------- cluster address this hit (cell) belongs to
***************************************************************************
```

COMMON / HBRECONST /

```
+KEEP,HBRECONST.
************************************************************************
*          Main common for HCAL hit collection
************************************************************************
C ---------------------------------------------------------------------
      integer NLAD,L_cell,ny_lay,Lsup_clust,L_clust,Lsub_clust,NHITS
      integer L_ch_track
C ---------------------------------------------------------------------
      real*4 a_cell,x_cell,y_cell,z_cell,r_cell,t_cell,p_cell
      real*4 summ_barrel, summ_hcapP, summ_hcapM, Summ_HCAL
C ---------------------------------------------------------------------
      PARAMETER (NLAD=4000) ! maximum allowed number of hits in HCAL
C ---------------------------------------------------------------------
      common/HBreconst/
     $       L_cell(nlad), a_cell(nlad),    ny_lay(nlad)
     $ ,     x_cell(nlad), y_cell(nlad),    z_cell(nlad)
     $ ,     r_cell(nlad), t_cell(nlad),    p_cell(nlad)
     $ , Lsup_clust(nlad),L_clust(nlad),L_ch_track(nlad)
     $ , NHITS,summ_barrel, summ_hcapP, summ_hcapM, Summ_HCAL
C ---------------------------------------------------------------------
       save /hbreconst/
c=======================================================================
c          Description of COMMON /HBreconst/
c              HIT/CELL INFORMATION
c=======================================================================
c    Nhits --------- number of hits in HCAL
c-----------------------------------------------------------------------
c    summ_barrel --- Whole Energy deposited in HCAL-barrel
c    summ_hcapP ---- Whole Energy deposited in HCAL CAP + Z
c    summ_hcapM ---- Whole Energy deposited in HCAL CAP - Z
c-----------------------------------------------------------------------
c    L_ch_track - charged track address this hit belongs to (=0 if not)
c-----------------------------------------------------------------------
c       L_cell -- Cell Status word collects all information
c                      related to cell assignment
c   L_cell =
c      i x 1 -- Super_Cluster number
c          +
c      i x 100 -- Cluster number in super-cluster or individual cluster
c
c         if L_cell = 0 then  -- it is an individual hit
c            (no one reconstruction or recognition program assigned it.
c-----------------------------------------------------------------------
c    a_cell -------- Amplitude of hit (cell)
c    ny_lay -------- Absorber layer number of hit (cell)
c    x,y,z_cell ---- Cartezian coordinates of hit (cell)
```

```
c    r,t,p_cell ---- Spherical coordinates of hit (cell)
c-------------------------------------------------------------------------
c          Working arraies
c-------------------------------------------------------------------------
c    Lsup_clust ---- super-cluster address this hit (cell) belongs to
c    L_clust ------- cluster address this hit (cell) belongs to
c=========================================================================
```

COMMON / HCLUSTERS /

```
+KEEP,HCLUSTERS.
*************************************************************************
*     The main common for neutral cluster collection in HCAL
*************************************************************************
C ----------------------------------------------------------------------
      integer NHCLUST,NHCLusters,LHSUP,LHCLU
      integer NHHITS,NHHITS1,NHHITS2
      integer NHHITSO,NHBEG,NHEND,IHYPHCL
      integer McluE
C ----------------------------------------------------------------------
      real*4 ENHCL,HCLCHISQ,DNHCL,EN1HCL,EN2HCL
      real*4 XHCL,YHCL,ZHCL,CXHCL,CYHCL,CZHCL
      real*4 RHCL,THCL,PHCL
C ----------------------------------------------------------------------
      PARAMETER (NHCLUST=200)
C ----------------------------------------------------------------------
       common/HCLusters/ NHCLusters,LHSUP(NHCLUST),LHCLU(NHCLUST)
     $,NHHITS(NHCLUST),NHHITS1(NHCLUST),NHHITS2(NHCLUST)
     $,NHHITSO(NHCLUST), NHBEG(NHCLUST),  NHEND(NHCLUST)
     $,IHYPHCL(NHCLUST), ENHCL(NHCLUST),HCLCHISQ(NHCLUST)
     $, DNHCL(NHCLUST), EN1HCL(NHCLUST) ,EN2HCL(NHCLUST)
     $,  XHCL(NHCLUST),   YHCL(NHCLUST),   ZHCL(NHCLUST)
     $, CXHCL(NHCLUST),  CYHCL(NHCLUST),  CZHCL(NHCLUST)
     $,  RHCL(NHCLUST),   THCL(NHCLUST),   PHCL(NHCLUST)
     $, McluE(NHCLUST)
C ----------------------------------------------------------------------
       save /HCLUSTERS/
************************************************************************
*      HCAL  Neutral Clusters Common Block description
C ----------------------------------------------------------------------
*  NHclusters - Number of HCAL neutral clusters collected at this common
*  LHSUP      - pointer to Super-cluster address in the COMMON/HCrecres/
*  LHCLU      - pointer to Cluster address in the COMMON/HCrecres/
*  NHHITS     - Number of hits belongs to this cluster
*  NHHITSO    - Number of hits belongs to this cluster (working variable)
*  NHBEG      - Number of layers have zero at the begining of HCAL
```

```
*  NHEND      - Number of layers have zero at the end of HCAL
*  IHYPHCL    - Hypothesis number for this cluster,
*                  see COMMON/caloevent/ for the hypothesis description
*  ENHCL      - Cluster energy
*  HCLCHISQ   - Chi^2 from the inetria tensor definition (Sum of squared
*               distances of hits from the main principal axes)
*  DNHCL      - Cluster average density [MeV/hit]
*---------------------------------------------------------------------
*  XHCL,YHCL,ZHCL - cluster center of gravity
*  CXHCL,CYHCL,CZHCL - direct Cosine for the principal axis
*  RHCL       - 3-D radius of center of gravity of cluster
*  THCL       - Theta angle of center of gravity of cluster
*  PHCL       - Phi angle of center of gravity of cluster
* ---------------------------------------------------------------------
*  NHHITS1    - Number of hits belongs to this cluster is at the
*                 first part of HCAL (1 - 10 layers)
*  EN1HCL     - Cluster energy in the first part of HCAL (1 - 10 layers)
*  XHCL_fst,YHCL_fst,ZHCL_fst  - Center of gravity of the first part
*                                 of HCAL (1 - 10 layers)
*  NHHITS2    - Number of hits belongs to this cluster is at the
*                 last part of the HCAL (11 - 40 layers)
*  EN2HCL     - Cluster energy in the second part of HCAL (11 - 40 layers)
* ---------------------------------------------------------------------
*  McluE      - ECAL cluster address if merged exists (= 0 if not exists)
************************************************************************
```

COMMON / ECRECRES /

```
+KEEP,ECRECRES.
*************************************************************************
*      Intermediate common for the cluster find procedure in ECAL
*******************************************************************
C ---------------------------------------------------------------------
      integer NSUPE,NCLUE,NHsup_clustE, NH_clustE
      integer Ic_clustE,Isup_clsE,Iclu_clsE
      integer Lsup_clsE,Lclu_clsE,Nsup_clustE,N_clustE
C ---------------------------------------------------------------------
      real*4 Esup_clustE,E_clustE
      real*4 STTsup_clustE,STT_clustE
      real*4 SPTsup_clustE,SPT_clustE
      real*4 STBsup_clustE,STB_clustE
      real*4 SPBsup_clustE,SPB_clustE
      real*4 Rsup_clustE,  R_clustE
      real*4 Tsup_clustE,  T_clustE
      real*4 Psup_clustE,  P_clustE
      real*4 Xsup_clustE,  X_clustE
      real*4 Ysup_clustE,  Y_clustE
      real*4 Zsup_clustE,  Z_clustE
```

```
C ---------------------------------------------------------------------
       PARAMETER (NSUPE=70, NCLUE=500)
C ---------------------------------------------------------------------
       COMMON/ECrecres/ Nsup_clustE,N_clustE(NSUPE)
     $, NHsup_clustE(NSUPE), NH_clustE(NCLUE,NSUPE)
     $,  Esup_clustE(NSUPE),  E_clustE(NCLUE,NSUPE)
     $,STTsup_clustE(NSUPE),STT_clustE(NCLUE,NSUPE)
     $,SPTsup_clustE(NSUPE),SPT_clustE(NCLUE,NSUPE)
     $,STBsup_clustE(NSUPE),STB_clustE(NCLUE,NSUPE)
     $,SPBsup_clustE(NSUPE),SPB_clustE(NCLUE,NSUPE)
     $,  Rsup_clustE(NSUPE),  R_clustE(NCLUE,NSUPE)
     $,  Tsup_clustE(NSUPE),  T_clustE(NCLUE,NSUPE)
     $,  Psup_clustE(NSUPE),  P_clustE(NCLUE,NSUPE)
     $,  Xsup_clustE(NSUPE),  X_clustE(NCLUE,NSUPE)
     $,  Ysup_clustE(NSUPE),  Y_clustE(NCLUE,NSUPE)
     $,  Zsup_clustE(NSUPE),  Z_clustE(NCLUE,NSUPE)
     $,                      iC_clustE(NCLUE,NSUPE)
     $    ,Isup_clsE(NSUPE), Iclu_clsE(NCLUE)
     $ ,Lsup_clsE(NSUPE,NSUPE),Lclu_clsE(NCLUE,NCLUE)
C ---------------------------------------------------------------------
       save /ecrecres/
c=====================================================================
c     Nsup_clust  -- Number of super-clusters found
c     N_clust     -- Number of clusters found
c     NH***_clust -- Number of hits in super-cluster N
c     E***_clust,S***_clust -- Energy and Sizes of sup, clusters
c     R          -- Energy radius of cluster (see program for definition)
c     T,P        -- Spherical coordinates of sup, clusters
c     X,Y,Z      -- Cartezian coordinates of sup, clusters
c          Working array
c     I***_cls   -- number of close neibours
c     L***_cls   -- addresses of close neibours
************************************************************************
```

COMMON / HBRECRES /

```
+KEEP,HBRECRES.
************************************************************************
*      Intermediate common for the cluster find procedure in HCAL
****************************************************************
C ---------------------------------------------------------------------
      integer NSUP,NCLU,NHsup_clust,NH_clust,Nsup_clust,N_clust
      integer Isup_cls,Iclu_cls,Lsup_cls,Lclu_cls
C ---------------------------------------------------------------------
      real*4  Esup_clust,  E_clust
      real*4 STTsup_clust,STT_clust
      real*4 SPTsup_clust,SPT_clust
      real*4 STBsup_clust,STB_clust
      real*4 SPBsup_clust,SPB_clust
```

```
      real*4 Rsup_clust,  R_clust
      real*4 Tsup_clust,  T_clust
      real*4 Psup_clust,  P_clust
      real*4 Xsup_clust,  X_clust
      real*4 Ysup_clust,  Y_clust
      real*4 Zsup_clust,  Z_clust
C ---------------------------------------------------------------------
       PARAMETER (NSUP=50, NCLU=200)
C ---------------------------------------------------------------------
       common/HBrecres/ Nsup_clust,N_clust(NSUP)
     $ , NHsup_clust(NSUP), NH_clust(NCLU,NSUP)
     $ ,  Esup_clust(NSUP),  E_clust(NCLU,NSUP)
     $ ,STTsup_clust(NSUP),STT_clust(NCLU,NSUP)
     $ ,SPTsup_clust(NSUP),SPT_clust(NCLU,NSUP)
     $ ,STBsup_clust(NSUP),STB_clust(NCLU,NSUP)
     $ ,SPBsup_clust(NSUP),SPB_clust(NCLU,NSUP)
     $ ,  Rsup_clust(NSUP),  R_clust(NCLU,NSUP)
     $ ,  Tsup_clust(NSUP),  T_clust(NCLU,NSUP)
     $ ,  Psup_clust(NSUP),  P_clust(NCLU,NSUP)
     $ ,  Xsup_clust(NSUP),  X_clust(NCLU,NSUP)
     $ ,  Ysup_clust(NSUP),  Y_clust(NCLU,NSUP)
     $ ,  Zsup_clust(NSUP),  Z_clust(NCLU,NSUP)
     $     ,Isup_cls(NSUP),      Iclu_cls(NCLU)
     $ ,Lsup_cls(NSUP,NSUP),Lclu_cls(NCLU,NCLU)
C ---------------------------------------------------------------------
       save /hbrecres/
c=====================================================================
c     Nsup_clust  -- Number of super-clusters found
c     N_clust     -- Number of clusters found
c     NH***_clust -- Number of hits in super-cluster N
c     E***_clust,S***_clust -- Energy and Sizes of sup, clusters
c     R          -- Energy radius of cluster (see program for definition)
c     T,P        -- Spherical coordinates of sup, clusters
c     X,Y,Z      -- Cartezian coordinates of sup, clusters
c          Working array
c     I***_cls   -- number of close neibours
c     L***_cls   -- addresses of close neibours
***********************************************************************
```

COMMON / ECRECWORK /

```
+KEEP,ECRECWORK.
*****************************************************************
*      Working common for the cluster find procedure
*****************************************************************
      integer Nhit_maxE,ladrNNE,ih_maxENN,jh_maxENN,ladr_wE
      integer idworkE,nx_wEork_h,ny_wEork_h,nn_wEorkh
C ---------------------------------------------------------------------
      real*4 a_maxENN,r_maxENN,ph_maxENN,th_maxENN,a_wE
```

```
      real*4 xh_wEork_st,yh_wEork_st,h_wEork
C ----------------------------------------------------------------------
      PARAMETER (Nhit_maxE=500)
C ----------------------------------------------------------------------
      common/ecrecwork/  ladrNNE(Nhit_maxE)
     $ , a_maxENN(Nhit_maxE), r_maxENN(Nhit_maxE)
     $ ,ih_maxENN(Nhit_maxE),jh_maxENN(Nhit_maxE)
     $ ,ph_maxENN(Nhit_maxE),th_maxENN(Nhit_maxE)
     $ ,a_wE(500),ladr_wE(500)
     $ ,idworkE,nx_wEork_h,ny_wEork_h
     $ ,nn_wEorkh ,xh_wEork_st,yh_wEork_st
     $ ,h_wEork(100000)
C ----------------------------------------------------------------------
       save /ecrecwork/
*******************************************************************
```

COMMON / HBRECWORK /

```
+KEEP,HBRECWORK.
**********************************************************************
*      Working common for the cluster find procedure in HCAL
*******************************************************************
C ----------------------------------------------------------------------
      integer Nhit_max,ladrNN,ih_maxNN,jh_maxNN,ladr_w
      integer idwork,nx_work_h,ny_work_h,nn_workh
C ----------------------------------------------------------------------
      real*4 a_maxNN,r_maxNN,ph_maxNN,th_maxNN
      real*4 xh_work_st,yh_work_st,h_work,a_w
C ----------------------------------------------------------------------
      PARAMETER (Nhit_max=200)
C ----------------------------------------------------------------------
       common/hbrecwork/  ladrNN(Nhit_max)
     $ , a_maxNN(Nhit_max), r_maxNN(Nhit_max)
     $ ,ih_maxNN(Nhit_max),jh_maxNN(Nhit_max)
     $ ,ph_maxNN(Nhit_max),th_maxNN(Nhit_max)
     $ ,a_w(500),ladr_w(500)
     $ ,idwork,nx_work_h,ny_work_h
     $ ,nn_workh ,xh_work_st,yh_work_st
     $ ,h_work(100000)
C ----------------------------------------------------------------------
       save /hbrecwork/
c======================================================================
c     PARAMETER Nhit_max is allowed for searching clusters
c     ladrNN      -- addresses of maximal amplitudes
C     a,l,i,j,p,t -- amplitude and coordinate information
c                    for Nhit_max hits
**********************************************************************
```

COMMON / CALOEVENT /

Final event information – reconstructed objects four–vectors at IP.

```
+KEEP,CALOEVENT.
************************************************************************
*         Main output common after the reconstruction
************************************************************************
      integer NpjE,NpjH,Npj,NspvE,NspvH,Nspv,NpvE,NpvH,Npv
      integer ipvect,lpvect
C ----------------------------------------------------------------------
      real*4 PjetsE,SuPvectE,PvectE
      real*4 PjetsH,SuPvectH,PvectH
      real*4 Pjets, SuPvect, Pvect
C ----------------------------------------------------------------------
      PARAMETER (LPVECT=500)    ! 500 4-vectors is allowed to store
C ----------------------------------------------------------------------
      COMMON/caloevent/ Npv, Pvect(8,LPVECT)
      dimension ipvect(8,LPVECT)
      equivalence (pvect(1,1),ipvect(1,1))
C ----------------------------------------------------------------------
       save /caloevent/
C ----------------------------------------------------------------------
c   Record of particle four-vectors (i) after the reconstruction.
c   Momentum given: - for charged at PCA, for neutrals at IP
C ----------------------------------------------------------------------
C              Pvect(1,i) = P_x
C              Pvect(2,i) = P_y
C              Pvect(3,i) = P_z
C              Pvect(4,i) = sqrt(P**2 + M**2)
C              iPvect(5,i) - Reconstructed object hypothesis identificator
c                                (see below)
C              Pvect(6,i)
c                 = 1.0         Direct IP connection for charged track
c                 = 4.0         particle has hits in ECAL and in HCAL
c                                   Photon - like
c                 = 5.0         particle has hits in ECAL and in HCAL
c                                   Hadron - like
c                 = 6.0         particle has hits in ECAL and in HCAL
c                                   or Hadron - like or photon - like
c                 = 7.0         particle has hits in HCAL only
c                                   Hadron - like
C              iPvect(7,i)  - pointer to -->
c                  for charged - track number from common /CALOINPUT/,
C                              "charged track information at the point
c                                of closest approach to the IP"
C                  for neutral - cluster number in common/Eclusters/ or
c                                              in common/Hclusters/
C              Pvect(8,i) = object charge
C ----------------------------------------------------------------------
```

```
C Reconstructed object hypothesis identificator and
C            color in calorimeter drawings
C ----------------------------------------------------------------------
c           1 - neutral hadron like
c           2 - e+ or e- like
c           3 - mu+ or mu- like
c           4 - charged hadron like
c           5 - this particle has no ID, or any hypothesis was not assigned
c           6 - photon like
c   1 - black, 2 - red, 3 - green, 4 - blue, 5 - yellow, 6 - pink.
*************************************************************************
```

COMMON /HBARGEOM /

```
+KEEP,HBARGEOM.
*************************************************************************
*     Geomerty and electronic cell database for HCAL Barrel
*************************************************************************
c---------------------------------------------------------
      integer ny_cell,ny_sampl,max_nx,max_nz
      integer nx_pad,ny_pad,nz_pad,nxp,nyp,nzp
      integer nswitch,nelect,nlay1,nlay2
c---------------------------------------------------------
      real Y_had_bar1,Y_had_bar2,Y_had_bar3,Z_size
      real t_st_steel,t_scint,t_fiber_gap,t_wall,toll_gap
      real x0_hbar, y0_hbar, x1_hbar, y1_hbar
      real x2_hbar, y2_hbar, x3_hbar, y3_hbar
      real x_plate_size,z_plate_size,y_plate_cent
      real x_sh_mod1,x_sh_mod2,y_sh_mod1,y_sh_mod2,z_sh_mod
      real x_step,y_step,z_step
      real x_center,y_center,z_center
c---------------------------------------------------------
      parameter (ny_cell=9)
      parameter (max_nx=15)
      parameter (max_nz=52)
      parameter (ny_sampl=40)
c---------------------------------------------------------
      common /hbargeom/Y_had_bar1,Y_had_bar2,Y_had_bar3,Z_size
     $,t_st_steel,t_scint,t_fiber_gap,t_wall,toll_gap
     $,x0_hbar, y0_hbar, x1_hbar, y1_hbar
     $,x2_hbar, y2_hbar, x3_hbar, y3_hbar
     $,x_sh_mod1,x_sh_mod2,y_sh_mod1,y_sh_mod2,z_sh_mod
     $,x_plate_size(ny_sampl),z_plate_size,y_plate_cent(ny_sampl)
     $,nx_pad(ny_cell),ny_pad(ny_cell),nz_pad(ny_cell),nlay1,nlay2
     $,nxp(ny_sampl),nyp(ny_sampl),nzp(ny_sampl)
     $,nswitch(ny_sampl),nelect(ny_cell)
c       Pad size for barrel
```

```
     $,x_step(ny_sampl),y_step,z_step(ny_sampl)
c       Electronic cell center for barrel
     $,x_center(max_nx,ny_cell,max_nz)
     $,y_center(max_nx,ny_cell,max_nz)
     $,z_center(max_nx,ny_cell,max_nz)
c-----------------------------------------------------------
        save /hbargeom/
c-----------------------------------------------------------
c     Electronic cell structure for barrel
c     MUST be initialize in subroutine HBAGEOM
c-----------------------------------------------------------
c  Y_cell  number =  1  2  3  4  5  6  7  8  9
c      data nx_pad /15,15,14,13,12,11,10, 9, 8/
c      data ny_pad / 3, 3, 3, 4, 4, 4, 5, 5, 9/
c      data nz_pad /52,50,45,40,35,30,26,22,16/
C      data nlay1/32/, nlay2/8/
c-----------------------------------------------------------
**************************************************************************
```

COMMON / HCAPDB /

```
+KEEP,HCAPDB.
***********************************************************************
c    Geometrical database for HCAL ENDCAP electronic cells
***********************************************************************
       integer nz_0,nz_1,nz_2,nz_3,nz_4,nz_all
       integer kz_0,kz_1,kz_2,kz_3,kz_4
       integer kzmax_0,kzmax_1,kzmax_2,kzmax_3,kzmax_4
c----------------------------------------------------------------------
       integer nx_01, nx_02, nx_03
       integer ny_01, ny_02, ny_03
c----------------------------------------------------------------------
       integer nx_11a,nx_11b,nx_11
       integer nx_12a,nx_12b,nx_12, nx_13
       integer ny_11, ny_11a,ny_11b
       integer ny_12, ny_12a,ny_12b,ny_13
c----------------------------------------------------------------------
       integer nx_21a,nx_21b,nx_21
       integer nx_22a,nx_22b,nx_22, nx_23
       integer ny_21, ny_21a,ny_21b
       integer ny_22, ny_22a,ny_22b,ny_23
c----------------------------------------------------------------------
       integer nx_31a,nx_31b,nx_31
       integer nx_32a,nx_32b,nx_32,nx_33
       integer ny_31,ny_31a,ny_31b
       integer ny_32,ny_32a,ny_32b,ny_33
c----------------------------------------------------------------------
       integer nx_41,nx_42,nx_43
```

```
      integer ny_41,ny_42, ny_43
c-----------------------------------------------------------------------
*                    8      9     12     10     14)
      Parameter (kz_0=4,kz_1=3,kz_2=4,kz_3=5,kz_4=7)
      Parameter (nz_0=2,nz_1=3,nz_2=3,nz_3=2,nz_4=2)
      Parameter (nz_all=nz_0+nz_1+nz_2+nz_3+nz_4)
c-----------------------------------------------------------------------
      Parameter (nx_01=8,  nx_02=13, nx_03=17)
      Parameter (ny_01=7,  ny_02=7,  ny_03=6)
c-----------------------------------------------------------------------
      Parameter (nx_11a=27,nx_11b=6, nx_11=nx_11a+nx_11b,
     $           nx_12a=22,nx_12b=8, nx_12=nx_12a+nx_12b, nx_13=17)
      Parameter (ny_11=10, ny_11a=10,ny_11b=7,
     $           ny_12=10, ny_12a=10,ny_12b=7,ny_13=6)
c-----------------------------------------------------------------------
      Parameter (nx_21a=22,nx_21b=5,nx_21=nx_21a+nx_21b,
     $           nx_22a=18,nx_22b=7,nx_22=nx_22a+nx_22b, nx_23=14)
      Parameter (ny_21=8,  ny_21a=8,ny_21b=6,
     $           ny_22=8,  ny_22a=8,ny_22b=6,  ny_23=5)
c-----------------------------------------------------------------------
      Parameter (nx_31a=19,nx_31b=4,nx_31=nx_31a+nx_31b,
     $           nx_32a=16,nx_32b=6,nx_32=nx_32a+nx_32b, nx_33=10)
      Parameter (ny_31=7,  ny_31a=7,ny_31b=5,
     $           ny_32=7,  ny_32a=7,ny_32b=5, ny_33=4)
c-----------------------------------------------------------------------
      Parameter (nx_41=15, nx_42=14,nx_43=10)
      Parameter (ny_41=4,  ny_42=4, ny_43=4)
c-----------------------------------------------------------------------
cc MVL     REAL PHI1,DPHI,RNPDV,RNZ,ZC0,RMI1,RMA1,ZC01,RMI2,RMA2
cc MVL     REAL RMI3,RMA3,Z_C1,Z_C2,Z_C3,Z_C4,ZPOL,ZPLE
cc MVL      REAL Zpos0,Zpos1,Zpos2,Zpos3,Zpos4
      REAL  hcap_z_centr,bk_sampl
      REAL  r_magn,r_iner_base,r_iner_hole,gap_zero
      REAL sampling, first_shift, ang
      REAL shift_hca0,shift_hca1,shift_hca2,shift_hca3,shift_hca4
      REAL    dx_01,   dx_02,   dx_03,   dy_01   ,dy_02,   dy_03
      REAL x_siz_01,x_siz_02,x_siz_03,y_siz_01,y_siz_02,y_siz_03
      REAL  stx_c01, stx_c02, stx_c03, sty_c01, sty_c02, sty_c03
      REAL   dx_01a,  dx_02a,x_siz_03a
      REAL    dx_11,   dx_12,   dx_13,   dy_11   ,dy_12,   dy_13
      REAL x_siz_11,x_siz_12,x_siz_13,y_siz_11,y_siz_12,y_siz_13
      REAL  stx_c11, stx_c12, stx_c13, sty_c11, sty_c12, sty_c13
      REAL   dx_11a,  dx_11b,  dx_12a,  dx_12b
      REAL x_siz_11a, x_siz_11b, x_siz_12a, x_siz_12b
      REAL stx_c11a,stx_c11b,stx_c12a,stx_c12b
      REAL sty_c11a,sty_c11b,sty_c12a,sty_c12b
      REAL    dx_21,   dx_22,   dx_23,   dy_21   ,dy_22,   dy_23
      REAL x_siz_21,x_siz_22,x_siz_23,y_siz_21,y_siz_22,y_siz_23
      REAL  stx_c21, stx_c22, stx_c23, sty_c21, sty_c22, sty_c23
```

```
      REAL  dx_21a,  dx_21b,  dx_22a,  dx_22b
      REAL x_siz_21a, x_siz_21b, x_siz_22a, x_siz_22b
      REAL stx_c21a,stx_c21b,stx_c22a,stx_c22b
      REAL sty_c21a,sty_c21b,sty_c22a,sty_c22b
      REAL    dx_31,   dx_32,   dx_33,   dy_31   ,dy_32,   dy_33
      REAL x_siz_31,x_siz_32,x_siz_33,y_siz_31,y_siz_32,y_siz_33
      REAL  stx_c31, stx_c32, stx_c33, sty_c31, sty_c32, sty_c33
      REAL   dx_31a,  dx_31b,  dx_32a,  dx_32b
      REAL x_siz_31a, x_siz_31b, x_siz_32a, x_siz_32b
      REAL stx_c31a,stx_c31b,stx_c32a,stx_c32b
      REAL sty_c31a,sty_c31b,sty_c32a,sty_c32b
      REAL    dx_41,   dx_42,   dx_43,   dy_41   ,dy_42,   dy_43
      REAL x_siz_41,x_siz_42,x_siz_43,y_siz_41,y_siz_42,y_siz_43
      REAL  stx_c41, stx_c42, stx_c43, sty_c41, sty_c42, sty_c43
       REAL tx_01,tx_02,tx_03,ty_01,ty_02,ty_03
       REAL tx_11,   tx_12,   tx_13,   ty_11,   ty_12,   ty_13
       REAL tx_11a,  tx_11b,  tx_12a,  tx_12b
       REAL ty_11a,  ty_11b,  ty_12a,  ty_12b
       REAL tx_21,   tx_22,   tx_23,   ty_21,   ty_22,   ty_23
       REAL tx_21a,  tx_21b,  tx_22a,  tx_22b
       REAL ty_21a,  ty_21b,  ty_22a,  ty_22b
       REAL tx_31,   tx_32,   tx_33,   ty_31,   ty_32,   ty_33
       REAL tx_31a,  tx_31b,  tx_32a,  tx_32b
       REAL ty_31a,  ty_31b,  ty_32a,  ty_32b
       REAL tx_41,   tx_42,  tx_43,   ty_41,   ty_42,   ty_43
       REAL x_cen_01, y_cen_01, z_cen_01, x_cen_02, y_cen_02
       REAL z_cen_02, x_cen_03, y_cen_03, z_cen_03

       REAL x_cen_11, y_cen_11, z_cen_11, x_cen_12
       REAL y_cen_12, z_cen_12, x_cen_13, y_cen_13, z_cen_13
       REAL x_cen_21, y_cen_21, z_cen_21, x_cen_22, y_cen_22
       REAL z_cen_22, x_cen_23, y_cen_23
       REAL z_cen_23, x_cen_31, y_cen_31, z_cen_31
       REAL x_cen_32, y_cen_32, z_cen_32, x_cen_33
       REAL y_cen_33, z_cen_33, x_cen_41, y_cen_41
       REAL z_cen_41, x_cen_42, y_cen_42
       REAL z_cen_42, x_cen_43, y_cen_43, z_cen_43
c-----------------------------------------------------------------------
      COMMON/HCAPDB/
c-----------------------------------------------------------------------
ccc       Pad size and coordinates database for endcap
c-----------------------------------------------------------------------
     $ r_magn,r_iner_base,r_iner_hole,gap_zero
     $,   sampling, bk_sampl, first_shift
     $,shift_hca0,shift_hca1,shift_hca2,shift_hca3,shift_hca4
     $,kzmax_0,kzmax_1,kzmax_2,kzmax_3,kzmax_4
     $,    dx_01,   dx_02,   dx_03,   dy_01   ,dy_02,   dy_03
     $, x_siz_01,x_siz_02,x_siz_03,y_siz_01,y_siz_02,y_siz_03
     $,  stx_c01, stx_c02, stx_c03, sty_c01, sty_c02, sty_c03
```

```
     $,    tx_01,   tx_02,   tx_03,   ty_01,   ty_02,   ty_03
     $,    dx_01a,  dx_02a,x_siz_03a
c
     $,    dx_11,   dx_12,   dx_13,   dy_11   ,dy_12,   dy_13
     $, x_siz_11,x_siz_12,x_siz_13,y_siz_11,y_siz_12,y_siz_13
     $,  stx_c11, stx_c12, stx_c13, sty_c11, sty_c12, sty_c13
     $,    tx_11,   tx_12,   tx_13,   ty_11,   ty_12,   ty_13
c
     $,    dx_11a,  dx_11b,  dx_12a,  dx_12b
     $, x_siz_11a, x_siz_11b, x_siz_12a, x_siz_12b
     $,  stx_c11a,stx_c11b,stx_c12a,stx_c12b
     $,    tx_11a,  tx_11b,  tx_12a,  tx_12b
     $,  sty_c11a,sty_c11b,sty_c12a,sty_c12b
     $,    ty_11a,  ty_11b,  ty_12a,  ty_12b
c
     $,    dx_21,   dx_22,   dx_23,   dy_21   ,dy_22,   dy_23
     $, x_siz_21,x_siz_22,x_siz_23,y_siz_21,y_siz_22,y_siz_23
     $,  stx_c21, stx_c22, stx_c23, sty_c21, sty_c22, sty_c23
     $,    tx_21,   tx_22,   tx_23,   ty_21,   ty_22,   ty_23
c
     $,    dx_21a,  dx_21b,  dx_22a,  dx_22b
     $, x_siz_21a, x_siz_21b, x_siz_22a, x_siz_22b
     $,  stx_c21a,stx_c21b,stx_c22a,stx_c22b
     $,    tx_21a,  tx_21b,  tx_22a,  tx_22b
     $,  sty_c21a,sty_c21b,sty_c22a,sty_c22b
     $,    ty_21a,  ty_21b,  ty_22a,  ty_22b
c
     $,    dx_31,   dx_32,   dx_33,   dy_31   ,dy_32,   dy_33
     $, x_siz_31,x_siz_32,x_siz_33,y_siz_31,y_siz_32,y_siz_33
     $,  stx_c31, stx_c32, stx_c33, sty_c31, sty_c32, sty_c33
     $,    tx_31,   tx_32,   tx_33,   ty_31,   ty_32,   ty_33
c
     $,    dx_31a,  dx_31b,  dx_32a,  dx_32b
     $, x_siz_31a, x_siz_31b, x_siz_32a, x_siz_32b
     $,  stx_c31a,stx_c31b,stx_c32a,stx_c32b
     $,    tx_31a,  tx_31b,  tx_32a,  tx_32b
     $,  sty_c31a,sty_c31b,sty_c32a,sty_c32b
     $,    ty_31a,  ty_31b,  ty_32a,  ty_32b
c
     $,    dx_41,   dx_42,   dx_43,   dy_41   ,dy_42,   dy_43
     $, x_siz_41,x_siz_42,x_siz_43,y_siz_41,y_siz_42,y_siz_43
     $,  stx_c41, stx_c42, stx_c43, sty_c41, sty_c42, sty_c43
     $,    tx_41,   tx_42,  tx_43,   ty_41,   ty_42,   ty_43
c----------------------------------------------------------------------
ccc       Electronic cell center for endcap
c----------------------------------------------------------------------
     $, x_cen_01(nx_01,ny_01,nz_0)
     $, y_cen_01(nx_01,ny_01,nz_0)
     $, z_cen_01(nx_01,ny_01,nz_0)
```

```
     $, x_cen_02(nx_02,ny_02,nz_0)
     $, y_cen_02(nx_02,ny_02,nz_0)
     $, z_cen_02(nx_02,ny_02,nz_0)
     $, x_cen_03(nx_03,ny_03,nz_0)
     $, y_cen_03(nx_03,ny_03,nz_0)
     $, z_cen_03(nx_03,ny_03,nz_0)
c
     $, x_cen_11(nx_11,ny_11,nz_1)
     $, y_cen_11(nx_11,ny_11,nz_1)
     $, z_cen_11(nx_11,ny_11,nz_1)
     $, x_cen_12(nx_11,ny_12,nz_1)
     $, y_cen_12(nx_11,ny_12,nz_1)
     $, z_cen_12(nx_11,ny_12,nz_1)
     $, x_cen_13(nx_13,ny_13,nz_1)
     $, y_cen_13(nx_13,ny_13,nz_1)
     $, z_cen_13(nx_13,ny_13,nz_1)
c
     $, x_cen_21(nx_21,ny_21,nz_2)
     $, y_cen_21(nx_21,ny_21,nz_2)
     $, z_cen_21(nx_21,ny_21,nz_2)
     $, x_cen_22(nx_22,ny_22,nz_2)
     $, y_cen_22(nx_22,ny_22,nz_2)
     $, z_cen_22(nx_22,ny_22,nz_2)
     $, x_cen_23(nx_23,ny_23,nz_2)
     $, y_cen_23(nx_23,ny_23,nz_2)
     $, z_cen_23(nx_23,ny_23,nz_2)
c
     $, x_cen_31(nx_31,ny_31,nz_3)
     $, y_cen_31(nx_31,ny_31,nz_3)
     $, z_cen_31(nx_31,ny_31,nz_3)
     $, x_cen_32(nx_32,ny_32,nz_3)
     $, y_cen_32(nx_32,ny_32,nz_3)
     $, z_cen_32(nx_32,ny_32,nz_3)
     $, x_cen_33(nx_33,ny_33,nz_3)
     $, y_cen_33(nx_33,ny_33,nz_3)
     $, z_cen_33(nx_33,ny_33,nz_3)
c
     $, x_cen_41(nx_41,ny_41,nz_4)
     $, y_cen_41(nx_41,ny_41,nz_4)
     $, z_cen_41(nx_41,ny_41,nz_4)
     $, x_cen_42(nx_42,ny_42,nz_4)
     $, y_cen_42(nx_42,ny_42,nz_4)
     $, z_cen_42(nx_42,ny_42,nz_4)
     $, x_cen_43(nx_43,ny_43,nz_4)
     $, y_cen_43(nx_43,ny_43,nz_4)
     $, z_cen_43(nx_43,ny_43,nz_4)
     $, hcap_z_centr(nz_all)
c-----------------------------------------------------------------------
      SAVE /HCAPDB/
```

****

COMMON / /

COMMON / /

COMMON / /

COMMON / /

COMMON / TPCMERGE /

hbargeom.inc hcapdb.inc reconparam.inc

12 List of reconstruction programs

calor_reconst.f
- calor_reconst_ini.f
- clearcomm.f
- ec_clust_join.f
- ec_hit_join.f
- ecal_hcal_merge.f
- ecal_hcal_merge_again.f
- ecal_reconst.f
- eclust_axsis.f
- eclust_draw.f
- eclust_property.f
- eclust_shower_hypo.f
- find_cluster_core.f
- find_electron_clust.f
- find_hadron_clust_3.f
- find_hadron_cluster.f
- findec_clust.f
- findec_sup_clust.f
- findhb_clust.f
- findhb_sup_clust.f
- hcal_pair_merge.f
- hcal_reconst.f
- hclust_axsis.f
- hclust_draw.f

ilin_fit_3d.f
ip_propagate.f
rec_util.f
reconst_analyz.f
tpc_calo_merge.f
tpc_reconst_prep.f

---

13 List of analysis programs

calor_analyz.f
calor_analyz_2jet.f
calor_analyz_4jet.f
calor_analyz_h0.f
calor_analyz_w.f
calor_analyz_z0.f
calor_analyz_z0_gamm.f
calor_analyz_zh_ww.f
calor_draw.f
calor_print.f
make_n_jets.f

14 List of reconstruction preparation programs

ecal_reconst_prep.f
hcal_reconst_prep.f
hcap_reconst_prep.f

15 List of GEANT simulation level programs

cellmap.f
ecalendcap.f
ecalgeom.f
find_ecal_cell.f
find_hbar_cell.f
find_hcap_cell.f
g4g3ecal.f
gdtrak.f
gdxyz.f
guhadr.f
gukine.f
guout.f
guphad.f
gustep.f
gutrev.f
gxint.f
gxphys.f

hbageom.f
hbarbld.f
hcap_cell_db.f
hcapbld.f
hcapgeom.f
ufiles.f
ugeom.f
uginit.f
uglast.f
uhinit.f

---

ilin_fit_3d
rec_util
calor_draw
eclust_draw
hclust_draw
calor_print
reconst_analyz
event_analyz
event_analyz_2jet
event_analyz_4jet
event_analyz_h0
event_analyz_w
event_analyz_z0
event_analyz_z0_gamm
event_analyz_zh_ww
make_n_jets
ugeom
ecalgeom
ecalendcap
hbageom
hbarbld
hcapgeom
hcapbld
hcap_cell_db
cellmap
find_ecal_cell
find_hbar_cell
find_hcap_cell
gustep
gukine
guout
gdtrak
gdxyz
guhadr
guphad
gutrev
gxint
gxphys

ufiles
uginit
uglast
uhinit

Appendix

16 Electronic Cell Database Parameters

```
 ==================== HBARREL geometry ============
 Shift of center of energy -1.20214069cm
------------------------------------------------------------
 Inner Calorimeter Barrel Radius=  170.cm
 Outer Calorimeter Barrel Radius=  300.cm
 ECAL thickness                    20.7999992cm
 Inner HCAL barrel Radius=         190.800003cm
 Outer HCAL barrel Radius=         296.800018cm
 Z\_size of  HCAL barrel=           265.850006cm
 HCAL barrel wall thickness=       0.200000003cm
 -------   Electronic structure --------------
 Nlay   X   Z  N\_ch  Sum
   1   15   52  780  780
   2   15   50  750 1530
   3   14   45  630 2160
   4   13   40  520 2680
   5   12   35  420 3100
   6   11   30  330 3430
   7   10   26  260 3690
   8    9   22  198 3888
   9    8   16  128 4016
------------------------------------------------------------
               HCAL Barrel Module sizes
------------------------------------------------------------
 Tollerance gap to magnet vessel=  0.5cm in radius
 Y1\_HCAL\_barrel=  190.800003; Y2=  84.8000031; Y3\_HCAL\_barrel=  296.800018
 Sqrt( X2**2 + y2**2)=  298.291992
       X        Y
     0.000    0.000
    78.992    0.000
   114.117   84.800
     0.000  108.700
------------------------------------------------------------
 Shifts in HBAR1 -48.2773056 -42.4000015 -132.725006
 Shifts in HBAR2 -28.5293159 -96.75 -132.725006
------------------------------------------------------------
 Scintillator plate Z size =  265.450012
```

```
------------------------------------------------------------
  Ny  Y\_pad     X\_size  N\_x\_pad  X\_step  N\_z\_pad  Z\_step N\_pad
------------------------------------------------------------
  1   2.250     79.404   15      5.294      52    5.105    780
  2   4.900     80.502   15      5.367      52    5.105    780
  3   7.550     81.599   15      5.440      52    5.105    780
---- Electronic channels in layer 1 = 780 --------
 Average size 3  72.6002884  4.17169762  27.3963337  5.23415089
  4  10.200     82.697   15      5.513      50    5.309    750
  5  12.850     83.795   15      5.586      50    5.309    750
  6  15.500     84.892   15      5.659      50    5.309    750
---- Electronic channels in layer 2 = 750 --------
 Average size 3  78.5928726  4.2834568  29.6576862  5.44588709
  7  18.150     85.990   14      6.142      45    5.899    630
  8  20.800     87.088   14      6.221      45    5.899    630
  9  23.450     88.185   14      6.299      45    5.899    630
---- Electronic channels in layer 3 = 630 --------
 Average size 3  97.2398148  4.59848452  36.6942711  6.05757952
 10  26.100     89.283   13      6.868      40    6.636    520
 11  28.750     90.381   13      6.952      40    6.636    520
 12  31.400     91.478   13      7.037      40    6.636    520
 13  34.050     92.576   13      7.121      40    6.636    520
---- Electronic channels in layer 4 = 520 --------
 Average size 4  123.006905  4.97328329  46.4176979  6.81305361
 14  36.700     93.674   12      7.806      35    7.584    420
 15  39.350     94.771   12      7.898      35    7.584    420
 16  42.000     95.869   12      7.989      35    7.584    420
 17  44.650     96.967   12      8.081      35    7.584    420
---- Electronic channels in layer 5 = 420 --------
 Average size 4  159.64801  5.42485142  60.2445297  7.76173496
 18  47.300     98.064   11      8.915      30    8.848    330
 19  49.950     99.162   11      9.015      30    8.848    330
 20  52.600    100.260   11      9.115      30    8.848    330
 21  55.250    101.357   11      9.214      30    8.848    330
---- Electronic channels in layer 6 = 330 --------
 Average size 4  212.547668  5.96786261  80.206665  8.95581722
 22  57.900    102.455   10     10.245      26   10.210    260
 23  60.550    103.553   10     10.355      26   10.210    260
 24  63.200    104.650   10     10.465      26   10.210    260
 25  65.850    105.748   10     10.575      26   10.210    260
 26  68.500    106.845   10     10.685      26   10.210    260
---- Electronic channels in layer 7 = 260 --------
 Average size 5  283.136108  6.56646729  106.843796  10.3365269
 27  71.150    107.943    9     11.994      22   12.066    198
 28  73.800    109.041    9     12.116      22   12.066    198
 29  76.450    110.138    9     12.238      22   12.066    198
 30  79.100    111.236    9     12.360      22   12.066    198
 31  81.750    112.334    9     12.482      22   12.066    198
---- Electronic channels in layer 8 = 198 --------
```

```
 Average size 5  391.293427  7.31421185  147.657883  12.1514559
 32  84.400    113.431    8     14.179      16   16.591    128
 33  87.050    101.264    7     14.466      16   16.591    112
 34  89.700     88.611    6     14.768      16   16.591     96
 35  92.350     75.958    5     15.192      16   16.591     80
 36  95.000     63.305    4     15.826      16   16.591     64
 37  97.650     50.651    3     16.884      16   16.591     48
 38 100.300     37.998    2     18.999      16   16.591     32
 39 102.950     25.345    1     25.345      16   16.591     16
 40 105.600     12.692    1     12.692      16   16.591     16
---- Electronic channels in layer 9 = 128 --------
 Average size 8  737.362732  9.03428459  278.250061  16.68083
-----------------------------------------------------------
-----------------------------------------------------------
 Whole number of pads / module =   14442
 Total area of Barrel tiles / module =   940016.375 cm^2
 Whole number of electronic channels/module =   4016
-----------------------------------------------------------
 Whole number of pads in a calorimeter barrel  =   462144
 Total area of Barrel tiles =   30080524. cm^2
 Whole number of electronic channels in barrel =   128512
-----------------------------------------------------------
-----------------------------------------------------------
 PHI1, DPHI, NPDV, NZ   =   0.  360.  8.  2.
 RMI1, RMA1, RMI2, RMA2 =   203.880005  276.  203.880005  276.
 RMI1, RMA1             =   30.  276.
 ZC0, ZC01, ZC1, ZC2    =   0.  21.2000008  23.8500004  31.7999992
 ZC3, ZC4, ZPOL, ZPLE   =   26.5  37.0999985  0.5  0.150000006
-----------------------------------------------------------
-----------------------------------------------------------
 PHI1, DPHI, NPDV, NZ   =   0.  360.  8.  2.
 RMI1, RMA1, RMI2, RMA2 =   203.880005  276.  203.880005  276.
 RMI1, RMA1             =   30.  276.
 ZC0, ZC01, ZC1, ZC2    =   0.  21.2000008  23.8500004  31.7999992
 ZC3, ZC4, ZPOL, ZPLE   =   26.5  37.0999985  0.5  0.150000006
 Z shifts for cructures=
 282.600006  303.799988  327.649994  359.450012  385.950012
-----------------------------------------------------------
 277.163849  114.805031
 Shift of center of energy  0.cm
---------------------------------------------------
       Z coordinates of center of electronic cells
  287.900   298.500   307.775   315.725   323.675   332.950
  343.550   354.150   366.075   379.325   395.225   413.775
---------------------------------------------------
              X        Layer 0        Y
    Shift      Size   Step     Shift   Size    Step
                        Region 1
 8  235.00    72.16   12.03    0.00   84.81   12.12   7
```

```
 8  210.95   72.16   12.03    0.00   84.81   12.12   7
                          Region 2
13  121.34  167.50   12.88   84.81   90.19   12.88   7
13  198.65  167.50   12.88   84.81   90.19   12.88   7
                          Region 3
17    0.00  144.81   12.07  175.00   72.16   12.03   6
17    0.00  205.14   12.07  175.00   72.16   12.03   6
---------------------------------------------------
 Area 0 =   52455.9336  65081.1094  101013.766 =   218550.812 cm^2
---------------------------------------------------
---------------------------------------------------
               X        Layer 1        Y
    Shift      Size   Step     Shift   Size    Step
                          Region 1
27    0.00  230.00    8.52    0.00   84.81    8.48  10
 6  230.00   77.16   12.86    0.00   84.81   12.12   7
                          Region 2
22    0.00  187.41    8.52   84.81   84.81    8.48  10
 8  187.41  102.89   12.86   84.81   84.81   12.12   7
                          Region 3
17    0.00  219.74   12.93  169.61   77.55   12.93   6
---------------------------------------------------
 Area 1 =   234442.562  200532.141  201033.344 =   636008.062 cm^2
---------------------------------------------------
---------------------------------------------------
               X        Layer 2        Y
    Shift      Size   Step     Shift   Size    Step
                          Region 1
22    0.00  230.00   10.45    0.00   84.81   10.60   8
 5  230.00   77.16   15.43    0.00   84.81   14.13   6
                          Region 2
18    0.00  188.18   10.45   84.81   84.81   10.60   6
 7  188.18  108.03   15.43   84.81   84.81   14.13   6
                          Region 3
14    0.00  217.15   15.51  169.61   77.55   15.51   5
---------------------------------------------------
 Area 2 =   312588.5  262178.469  173220.062 =   747987. cm^2
---------------------------------------------------
---------------------------------------------------
               X        Layer 3        Y
    Shift      Size   Step     Shift   Size    Step
                          Region 1
19    0.00  230.00   12.11    0.00   84.81   12.12   7
 4  230.00   77.16   19.29    0.00   84.81   16.96   5
                          Region 2
16    0.00  193.68   12.11   84.81   84.81   12.12   7
 6  193.68  115.75   19.29   84.81   84.81   16.96   5
                          Region 3
10    0.00  193.88   19.39  169.61   77.55   19.39   4
```

```
----------------------------------------------------
 Area 3 =   260491.328  213333.75  15036.4746 =   488861.562 cm^2
----------------------------------------------------
----------------------------------------------------
               X        Layer 4        Y
    Shift     Size   Step     Shift   Size    Step
                        Region 1
15    0.00  307.16   20.48    0.00   82.39   20.60   4
                        Region 2
14    0.00  286.69   20.48   82.39   82.39   20.60   4
                        Region 3
10    0.00  204.78   20.48  164.78   82.39   20.60   4
----------------------------------------------------
 Area 4 =   354292.75  295243.938  200765.812 =   850302.5 cm^2
----------------------------------------------------
----------------------------------------------------
   Number of scintillation pads in one quadrant
----------------------------------------------------
 Layer     Reg 1     Reg 2     Reg 3    Total
----------------------------------------------------
   0        360       392       696      1448
----------------------------------------------------
   1       2808      2349       783      5940
----------------------------------------------------
   2       2472      2052       720      5244
----------------------------------------------------
   3       1530      1270       340      3140
----------------------------------------------------
   4        840       700       476      2016
----------------------------------------------------
----------------------------------------------------
Total      8010      6763      3015     17788
----------------------------------------------------
 Total area of EndCap tiles for one quadrant =   2941710. cm^2
----------------------------------------------------
 Total number of pads in calorimeter EndCap = 142304
 Total area of EndCap tiles =   23533680. cm^2
----------------------------------------------------
----------------------------------------------------
     Number of electronic cells in one quadrant
----------------------------------------------------
 Layer     Reg 1     Reg 2     Reg 3    Total
----------------------------------------------------
   0         90        98       174       362
----------------------------------------------------
   1        936       783       261      1980
----------------------------------------------------
   2        618       513       180      1311
----------------------------------------------------
```

```
   3      306      254       68      628
 --------------------------------------------------
   4      120      100       68      288
 --------------------------------------------------
 --------------------------------------------------
Total     2070     1748      751     4569
 --------------------------------------------------
  Total number of electronic cells in  calorimeter EndCap = 36552
 --------------------------------------------------
  Z-POSITION  284.600006
  Z-POSITION  287.25
  Z-POSITION  289.899994
  Z-POSITION  292.550018
  Z-POSITION  295.200012
  Z-POSITION  297.850006
  Z-POSITION  300.5
  Z-POSITION  303.149994
  Z-POSITION  305.799988
  Z-POSITION  308.449982
  Z-POSITION  311.099976
  Z-POSITION  313.75
  Z-POSITION  316.399994
  Z-POSITION  319.049988
  Z-POSITION  321.699982
  Z-POSITION  324.349976
  Z-POSITION  327.
  Z-POSITION  329.649994
  Z-POSITION  332.299988
  Z-POSITION  334.949982
  Z-POSITION  337.600006
  Z-POSITION  340.25
  Z-POSITION  342.899994
  Z-POSITION  345.549988
  Z-POSITION  348.199982
  Z-POSITION  350.850006
  Z-POSITION  353.5
  Z-POSITION  356.149994
  Z-POSITION  358.799988
  Z-POSITION  361.450012
  Z-POSITION  364.100006
  Z-POSITION  366.75
  Z-POSITION  369.400024
  Z-POSITION  372.050018
  Z-POSITION  374.700012
  Z-POSITION  377.350006
  Z-POSITION  380.
  Z-POSITION  382.650024
  Z-POSITION  385.300018
  Z-POSITION  387.950012
```

```
Z-POSITION  390.600006
Z-POSITION  393.25
Z-POSITION  395.900024
Z-POSITION  398.550018
Z-POSITION  401.200012
Z-POSITION  403.850006
Z-POSITION  406.5
Z-POSITION  409.150024
Z-POSITION  411.800018
Z-POSITION  414.450012
Z-POSITION  417.100006
Z-POSITION  419.75
Z-POSITION  422.400024
```