Geometry Description for MCFast

Abstract

This document contains a description of MCFast geometry structures and geometry files.

MCFast v4_3 and above
February 8, 2000

Contents

1 MCFast Geometry Description **2**

1.1 Changing a MCFast Geometry Template 3

1.2 Units 4

2 MCFast Geometry Input **5**

2.1 Absorber 5

2.2 Beampipe 7

2.3 Detector 7

2.4 Dipole 8

2.5 Drift Chambers 9

2.6 Electromagnetic Calorimeters 12

2.7 Forward tracking 14

2.8 Hadron calorimeter 23

2.9 Hits on track 23

2.10 Materials 24

2.11 Mixtures 25

2.12 Muon 26

2.13 Silicon barrel detectors 27

2.14 Silicon forward disks 34

2.15 Silicon planes (xy) 39

2.16 Silicon Rectangular Z-Planes (SiZRect) 41

2.16.1 Stereo angles for SiZRecT Devices 48

2.17 Solenoid magnet 49

3 The Include files **50**

3.1 Silicon barrel detectors 50

1 MCFast Geometry Description

The basic geometry shapes recognized by MCFast are planes, cylinders and cones. They can be combined to form detectors elements such as silicon trackers, drift chambers, and calorimeters. Detectors other than silicon devices are generally described by the outer box and the detector volume inside. Materials are included as either thin scatterers or as thick absorbers. Beam pipes and magnets are included as separate elements.

The detectors are assumed to be in regions of constant field. Magnets can be solenoids and dipoles. Detector elements and magnets are described in simple ASCII geometry files. The expected formats of the files and a brief description of the input is given in this document.

MCFast tracking currently converts detector layers and scatterers into infinitely thin planes or cylinders. They are put into two ordered lists or radial planes and Z-planes. (See the routine $MCFAST_DIR/src/geom/geo_trkorder.F). The calorimetry is the only part of MCFast that is volume based. MCFast can handle two tracking elements that are at the same radius or at the same

z position. MCFast allows does not check for overlapping volumes or planes, however, so be careful when defining your detector.

Geometry files in the DBin format are identified by the file extension *.db*. Examples of MCFast geometry files can be found in $MCFAST_DIR/examples/mcfast. (To define $MCFAST_DIR *setup mcfast*.)

The geometry input variables are defined by templates. The template definitions for the various geometry geometry can be found in $MCFAST_DIR/db. A link to $MCFAST_DIR/db must be made inside the local directory where you plan to run MCFast. (ln -s $MCFAST_DIR/db db)

Some geometry structures are nested which is defined in the templates by a parent/child relationship. The silicon barrel detector, for example, has several levels: barrel detector elements, layers and wafers. Because we are working in fortran, the maximum number of elements is defined inside the template. If you exceed this number, strange things may happen.

This can be a confusing point. The maximum number in the template is the total number of that element that can be defined in the detector. There is also a maximum number that is allowed per layer or per device that is defined internally to MCFast. For ordinary non-nested structures, these numbers should be the same. For nested structures, these numbers are different and have been set to reasonable values as determined by the authors. These numbers as they are implemented in the v2.6 version are given below.

1.1 Changing a MCFast Geometry Template

If you would like to increase this number then you will need to contact one of the authors or you can make your own version of MCFast libraries and change the template files yourself. Remember that the geometry structures are defined in the template files and separately in the include files. The geometry input is read in through condensed structures defined in the templates, and MCFast itself uses similar but less condensed structures which can be found in $MCFAST_DIR/inc/geom. The translation between the two

structure formats is done in the $MCFAST_DIR/src/geom load_*.F routines. Since we are currently using Fortran, the maximum size of the arrays must be set for both structures. We have had to do this in two separate places which can cause problems when something needs to be changed.

For experts and the very brave To change a template yourself, you will need to make your own copy of the mcfast files and have set the environment variable MCFAST_DIR to your new directory. In $MCFAST_DIR/src/dbin_mcfast you will find the main template file (template.db) and the code that is auto-generated by dbin. Create a link in this directory to the directory containing your new template: ln -s $MCFAST_DIR/db db. Once you have changed the template in $MCFAST_DIR/db and checked it against the corresponding fortran include file, run ~bphyslib/util/dev/dbin template.db. You should now rebuild all MCFast libraries and object files to be certain that there is no mismatch of version.

1.2 Units

Mcfast uses standard units of cm, ns, grams, GeV and Tesla. Angles are in radians.

2 MCFast Geometry Input

2.1 Absorber

The Absorber geometry structure describes cylindrical and conical inactive material. Absorbers are treated in two ways inside MCFast. Thin absorbers (type 41,42) are treated as scattering sites and NO showers will be initiated inside them. In v2_6 thick absorbers are treated as calorimeters without readout and showering can occur. (See EMCAL structures.) In v2_5_1 they are included in the graphics and the muon_offline code.

template **Absorber** (Max=25) thick absorbers or thin scatterers(cylindrical/conic)

char **name** name

char **shape** "TUBE" or "CONE"

int **type** 1=barrel, 2=forward, 41= barrel scatterer, 42= forward scatterer

real **rmin(2)** minimum radius at low z, high z

real **rmax(2)** maximum radius at low z, high z

real **z0** z at center

real **zlen** length in z

material **material** from material list

end template

include file $MCFAST_DIR/inc/geom/absorber.inc, absorber_struct.inc

The AbsorberBox (new in v2_6) geometry structure describes rectangular inactive material. Absorbers are treated in two ways inside MCFast. Thin absorbers (type 41,42) are treated as scattering sites and NO showers will be initiated inside them. Thick box absorbers are treated as calorimeters without readout and showering can occur. (See EMCALBox structures.)

template **AbsorberBox** (Max=25) thick absorbers or thin scatterers(cylindrical/conic)

char **name** name

char **shape** "BOX" only

int **type** 2=forward, 42= forward scatterer

real **xlimit(2)** xmin, xmax

real **ylimit(2)** ymin, ymax

real **xlimit_gap(2)** inner gap xmin, xmax

real **ylimit_gap(2)** inner gap ymin, ymax

real **z0** z at center

real **zlen** length in z

material **material** from material list

end template

include file $MCFAST_DIR/inc/geom/absorber.inc, absorber_struct.inc

2.2 Beampipe

The beampipe is treated as a scatterer.

template **BPipe** (Max=2) beam pipe (cylindrical)

char **name** name

real **rmin** minimum radius (usually = 0)

real **rmax** radius of inner wall

real **z0** z at center

real **zlen** length in z

material **mat_fill** fill material ("VACU" for example)

real **bndrthk(4)** wall thickness at rmin, rmax, front, back

material **matrbnd(4)** at rmin, rmax, front, back

end template

include file $MCFAST_DIR/inc/geom/beampipe.inc

2.3 Detector

This is where you will enter a general description of the detector including its name.

template **detector** Defines detector type for tracing

char **name** Detector Name

char **geom_id** Detector type
"CENTRAL" cylindrical geometry (solenoids)
"FORWARD" z planes (dipoles)
"COMBINED" both In early version of MCFast, this was used

to describe the geometry, now it simply serves as a switch for trk_offline. Combined (wtk) fitting is much slower and is not very reliable.

end detector

include file $MCFAST_DIR/inc/geom/detector_general.inc

2.4 Dipole

template **Dipole** (Max=20) Dipole magnets

char **name** name

real **bfield** magnitude of B field

real **dircos(3)** direction of b-field (current version assumes fields along x or y axis)

real **xmin** minimum x for field region

real **xmax** maximum x for field region

real **ymin** minimum y for field region

real **ymax** maximum z for field region

real **z0** central value for z

real **zlen** length of field region in z

material **mat_fill** fill material usually "AIR"

real **thick_boun(6)** Yoke thickness at: xmin, xmax, ymin, ymax, zmin, zmax

material **mat_boun(6)** Yoke material at: xmin, xmax, ymin, ymax, zmin, zmax

end template

include file $MCFAST_DIR/inc/geom/dipole.inc, dipole_struct.inc

2.5 Drift Chambers

Central Drift Chamber (cylindrical)

template **Drift** (Max=10)

int **num** Chamber number
char **name** Name
int **num_anode** Number of anode layers
child **LayerDRFAno** Link to daughter structure : no entry in .db file
child **OffsetDRFAno** Link to daughter structure: no entry in .db file
int **num_cathode** number of cathode layers (can be zero)
child **LayerDRFCatho** Link to cathode structure
real **rmin** minimum radius of drift volume
real **rmax** maximum radius of drift volume
real **z0** central Z position
real **zlen** length of chambers
material **material** fill material
real **thick_boun(4)** boundary thickness at rmin, rmax, zmin, zmax
material **mat_boun(4)** boundary material rmin, rmax, zmin, zmax

end template

include file $MCFAST_DIR/inc/geom/drift.inc, drift_struct.inc

Drift chamber substructure:

- Drift_cathode

  template **LayerDRFCatho** (Max=2 total, max= 2 per device) Drift Chamber Cathode (not used currently)

int **det** Pointer to parent Drift chamber
parent **Drift** parent structure: no entry in .db file
int **lyr** Cathode layer number
real **delta_r**
real **zlen** cathode length in z
int **nstrips** number of strips in z
int **n_phi_segm** number of segments in phi
int **ID_anode** points to corresponding anode layer
int **cell_offset** phi offset of strips
real **eff_hit** efficiency
real **resa**
real **resb**
real **resc**

end template

include file $MCFAST_DIR/inc/geom/drf_cathode_struct.inc

- Drift_layer

  template **LayerDRFAno** (Max=100 total, Max= 100 per device)

  int **det** pointer to Drift chamber
  parent **Drift** Parent Structure: No entry in .db file
  int **lyr** Anode layer number
  real **radius** radius of layer
  real **zlen** length of anode
  real **cell_height** height of cell
  int **nwires** number of wires in layer
  int **ID_readout** readout side +1 = zmax, -1 = zmin
  int **ID_cathode** cathode plan corresponding to this anode
  real **phi0** phi offset (phi of wire 0)
  real **stereo_tau**
  real **stereo_offset**
  real **eff_hit** hits efficiency

real **eff_dedx** efficiency for dedx info

real **siga** $\sigma = \sigma_a + \sigma_b \times drift + \sigma_c \times drift^2$ $(drift = (drifttime)/max(drifttime)$

real **sigb**

real **sigc**

end template

include file $MCFAST_DIR/inc/geom/drf_anode_struct.inc

- Drift_offset

template **OffsetDRFAno** (Max=100 total, Max = 100 per device)
Drift Chamber offsets (defaults to zero offset)

int **det** pointer to Drift chamber

parent **Drift** Parent Structure: no entry in structure

int **lyr** anode layer

real **cell_offset**

real **sag**

real **offset(3)**

real **dircos(3)**

end template

include file $MCFAST_DIR/inc/geom/drf_anode_struct.inc

2.6 Electromagnetic Calorimeters

The basic Cylindrical Calorimeter Structure is called EMCal. This will serve as the model a new calorimeter structure that will be developed for future versions.

template **EMCal** (Max=40)

char **name** Name of calorimeter

char **shape** Shape "TUBE" or "CONE"

int **type** central (=1), forward or backward(=2)

real **rmin(2)** minimum radius at lo z, hi z

real **rmax(2)** maximum radius at lo z, hi z

real **z0** central z position

real **zlen** length in z

material **material** average mixture of calorimeter material

material **active** active material

int **nphi** number of segments in phi

int **neta** number of segments in eta

real **siga_em** resolution for electromagnetic showers

real **sigb_em** offset for electromagnetic showers

real **siga_had** resolution for hadronic showers

real **sigb_had** offset for hadronic showers

real **em_had_ratio** e/pi ratio

end template

include file $MCFAST_DIR/inc/geom/emcal.inc, emcal_struct.inc

The basic Boxlike Calorimeter Structure is called CalorBox (New in v2_6).

template **CalorBox** (Max=40)

char **name** Name of calorimeter

char **shape** Shape "BOX" only

int **type** forward or backward(=2)

real **xlimit(2)** xmin, xmax

real **ylimit(2)** ymin, ymax

real **xlimit_gap(2)** inner gap xmin, xmax

real **ylimit_gap(2)** inner gap ymin, ymax

real **z0** central z position

real **zlen** length in z

material **material** average mixture of calorimeter material

material **active** active material

int **ncr1** number of segments in x

int **ncr2** number of segments in y

real **siga_em** resolution for electromagnetic showers

real **sigb_em** offset for electromagnetic showers

real **siga_had** resolution for hadronic showers

real **sigb_had** offset for hadronic showers

real **em_had_ratio** e/pi ratio

end template

include file $MCFAST_DIR/inc/geom/emcal.inc, emcal_struct.inc

2.7 Forward tracking

template **FTrk** (Max=100) Forward tracking devices – Box-like

int **det** Detector number (1...100)

char **name** Detector name

int **nlyr** number of detection planes

child **LayerFTRK** detection plane structure (no entry in .db file)

real **xmin** xmin of the device

real **xmax** xmax of the device

real **ymin** ymin of the device

real **ymax** ymax of the device

real **z0** central z position

real **zlen** length in z

material **mat_fill** fill material or gas mixture

real **thick_boun(6)** thickness of box at: xmin, xmax, ymin, ymax, zmin, zmax

material **mat_boun(6)** material of box at: xmin, xmax, ymin, ymax, zmin, zmax

real **siga** resolution

real **sigb**

real **sigc**

end template

include file $MCFAST_DIR/inc/geom/for_trk.inc, for_trk_struct.inc

Forward tracking layer structure:

- Ftrk_layer
  Detection planes for box-like forward chambers.

  template **LayerFTRK** (Max=500 total, max= 100 per device)

  int **det** pointer to FTrk

  parent **FTrk** FTrk structure (no entry in .db file)

  int **lyr** layer number. Layers must be numbered in increasing order of z.

  real **z_local** local z wrt to center of FTrk Box

  real **thick** thickness of detection layer

  real **xmin** xmin of detection plane

  real **xmax** xmax of detection plane

  real **xmin_gap** xmin of gap in detection plane

  real **xmax_gap** xmax of gap in detection plane

  real **ymin** ymin of detection plane

  real **ymax** ymax of detection plane

  real **ymin_gap** ymin of gap in detection plane

  real **ymax_gap** ymax of gap in detection plane

  real **stereo** stereo angle ($[-\frac{\pi}{2}, +\frac{\pi}{2},]$radians); see text for details.

  int **ncell** number of cells

  real **coord0_x** x coordinate of 1st cell

  real **coord0_y** y coordinate of 1st cell

  int **type** controls interpretation of stereo; see text for details.

  real **eff_hit** hit efficiency

  real **siga** $\sigma = \sigma_a + \sigma_b * (driftdist) + \sigma_c * (driftdist)^2$;
  if $\sigma = 0$, $\sigma = cellsize/\sqrt{12}$

  real **sigb**

  real **sigc**

  end template

  include file $MCFAST_DIR/inc/geom/for_trk_lyr_struct.inc

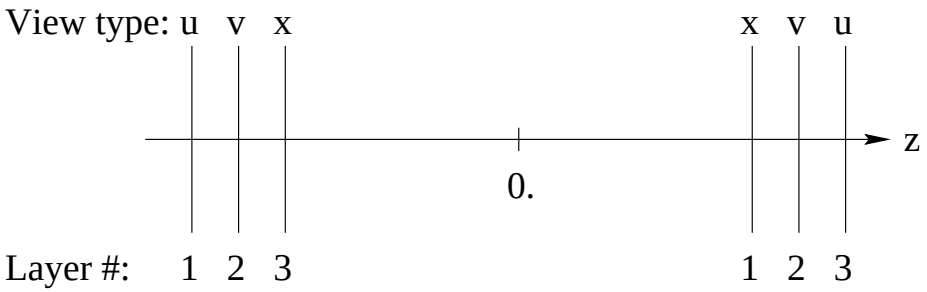


Figure 1: The rule for ordering layers within a forward tracking detector: layer 1 is at the lowest value of z. One consequence of this is that if an xuv multiplet is mirror-symmetric about $z = 0$, the mapping of layer number to view type is not preserved. The labeling of the views in this figure is for illustrative purposes only.

Inside MCFast it is assumed that, within one forward tracking detector, the layer numbers are in increasing order of z. This is illustrated in figure 1. For experts only: this assumption is made geo_trkorder_for_trk.F when the position of the scattering surfaces between layers is computed. The consequences of violating the rule are relatively small: the scattering in the material between layers occurs at the wrong value of z.

Until MCFast v5_0 the value of the `type` element of the ftrk template was undefined and was ignored by the code; there were some comments floating around which suggest that it once had a planned use which was never implemented. In MCFast v5_0 the element `type` was given meaning; the different values of `type` describe different interpretations of the `stereo` element of the ftrk template:

- `type=1` The outer edges of the forward tracking plane and the edges of the central hole are aligned with the world x and y axes; the wires are strung at the specified angle within that frame. See figures 2 and 3.
- `type=2` and `type=3`. The wires are strung parallel to the edges of the frame and the entire frame is rotated by the specified angle. The edges of the gap are parallel to the edges of the frame. See figures 4 and 5.

- All other values of `type` are treated as `type=1`.

Figure 2, which illustrates `type=1`, shows three rectangles. The outermost rectangle represents the outer edge of the material which makes up the forward tracking plane. The innermost rectangle represents the edges of the beam hole. The mid-sized rectangle represents the outer boundary of the active region of the layer. The diagonal lines represent several of the strips which fill the sensitive area; the stereo angle is indicated on the figure and, for this example, it is positive. Notice that the strips are broken by the central hole. The ftrk template quantities `xmin, xmax, ymin, ymax` are defined by large black dots on the outermost rectangle. The template quantities `coord0_x and coord0_y` are defined by the dot in the lower left corner of the sensitive area; this point is labeled (x_0, y_0). The coordinates of the upper right corner of the sensitive area are not specified: to deal with this, the code assumes that the right-hand insensitive margin has the same width as the left-hand one; similarly, the code assumes that the upper insensitive margin has the same height as the lower one; that is, the active area is centered on the full area of the layer. Before MCFast v5_0, there were bugs in the code which positioned the active area on the full area; the bugs were not triggered if the active area was the same size as the full area. The quantities labeled $\vec{\eta}$ and (x_1, y_1) are only of relevance to very expert users and to developers; they will be discussed at the end of this sub-section.

Figure 3 illustrates a configuration which has only one difference from that shown in figure 2: in this case the stereo angle is negative.

Figure 4 illustrates a forward tracking plane of type 2. The left part of the figure shows a plane when the stereo angle, θ, is zero. The quantities which define the size of the plane are defined in this configuration; these quantities are, `xmin, xmax, ymin, ymax, coord0_x` and `coord0_y`. This is indistinguishable from a type 1 detector with $\theta = 90^\circ$. The configuration on the right hand side illustrates the detector when θ is non-zero; in this case it is positive. This differs from a type 1 detector in that the entire detector, frame and all, has been rotated. For convenience, the parameters `xmin, xmax, ymin, ymax, coord0_x` and `coord0_y` are defined in the $\theta = 0$ configuration, even when θ has a non-zero value.

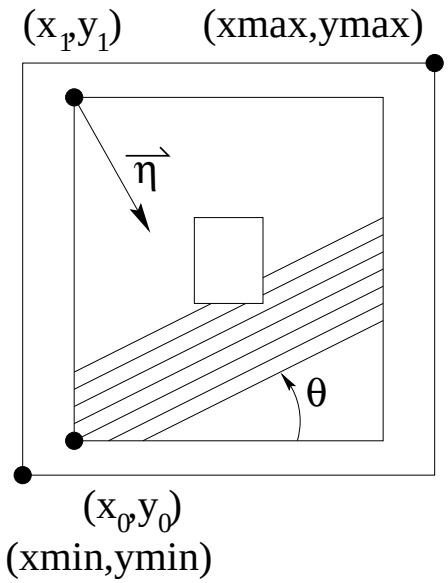


Figure 2: Example of a type 1 silicon strip detector with a positive stereo angle.

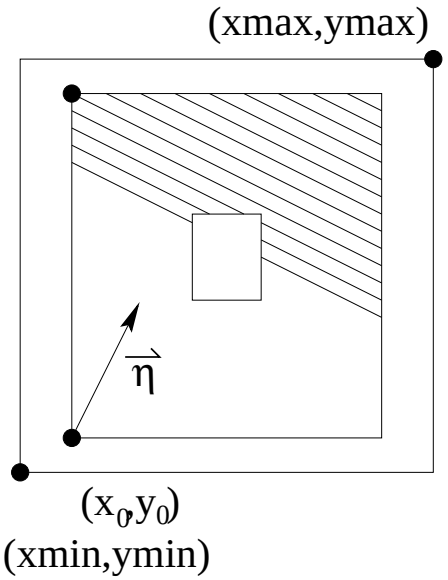


Figure 3: Example of a type 1 silicon strip detector with a negative stereo angle.

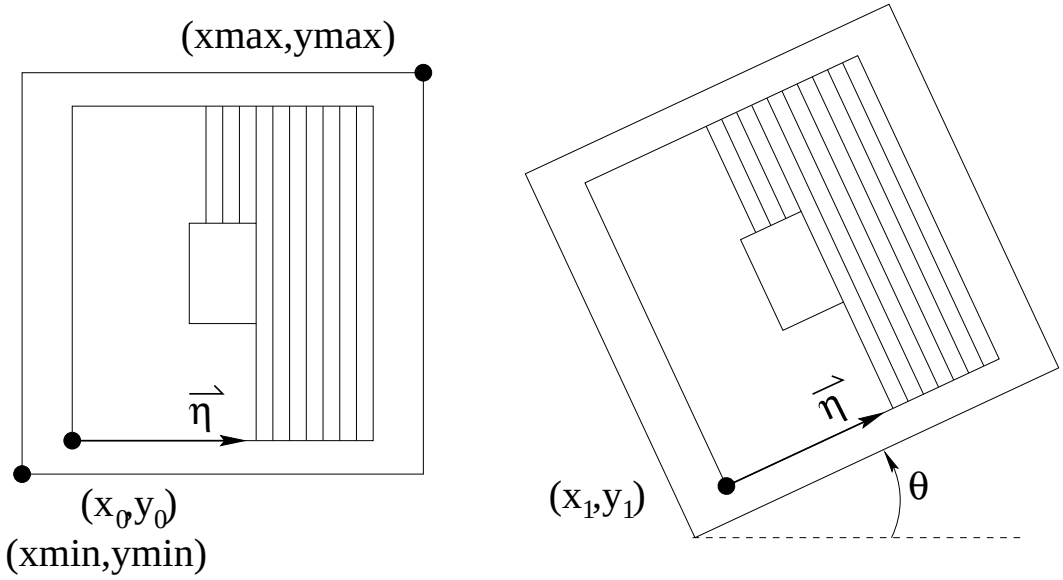


Figure 4: Example of a type 2 silicon strip detector with a positive stereo angle. The plane is oriented as shown in the right hand figure but the size parameters are defined in the $\theta = 0$ configuration shown to the left. The rotation is made about the z axis.

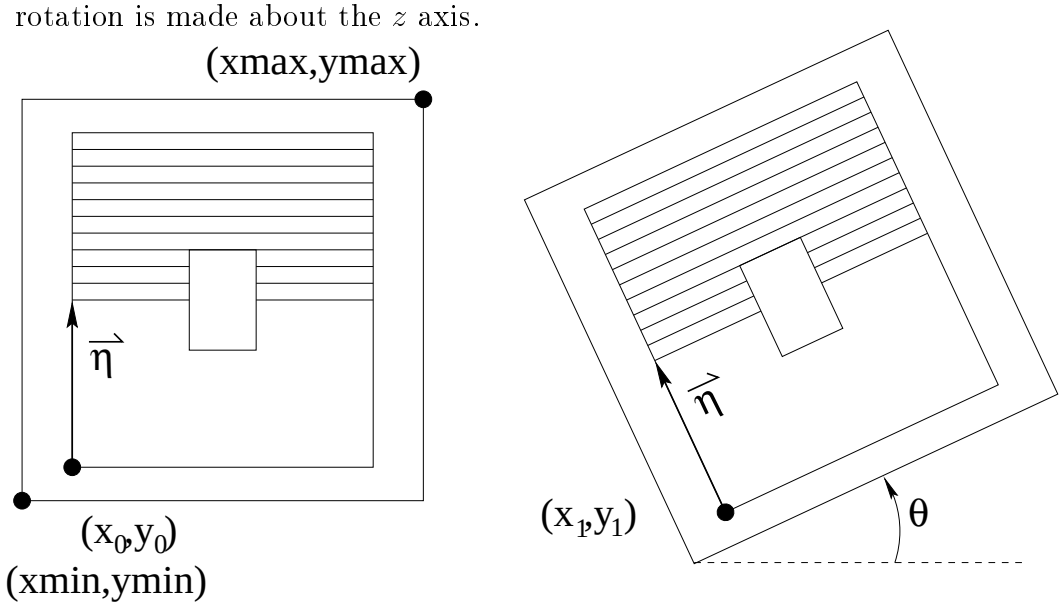


Figure 5: Example of a type 3 silicon strip detector with a positive stereo angle. This is distinguished from a type 2 detector by the orientation of the strips. The plane is oriented as shown in the right hand figure but the size parameters are defined in the $\theta = 0$ configuration shown to the left. The rotation is made about the z axis.

Figure 5 illustrates a forward tracking detector of type 3. This differs from a type 2 detector in the orientation of the wires.

There is no facility within MCFast to specify one rotation angle for the frame and a different one for the wires.

The remaining material in this sub-section is for experts only.

In the last four figures there are items labeled (x_1, y_1) and $\vec{\eta}$. These items are not found in the template and only expert MCFast users need consider them. They are used in the computation of the cell number of the hit cell and in the computations of the parameters which describe a hit. To compute the cell number one needs a reference point, $\vec{x_0}$, and a measurement direction $\vec{\eta}$. This direction lies in the plane of the detector and is perpendicular to the wire direction. Given a point in the plane, $\vec{x}$, the cell which contains the hit, n, is given by,

$$n \quad = \quad \mathrm{int}\left((\vec{x} - \vec{x_0}) \cdot \vec{\eta}\right). \tag{1}$$

With an appropriate choice of $\vec{x_0}$, this gives values of n which lie between 0 and `ncell-1`. Consider figure 3. In this figure the arrow labeled $\vec{\eta}$ shows the sign convention for the measurement direction and the point labeled (x_0, y_0) is the unique point which satisfies all of the requirements for the the reference point. This choice of reference point works for all negative stereo angles and for a stereo angle of zero.

Now consider figure 2, which shows a positive stereo angle. Again the arrow labeled $\vec{\eta}$ shows the sign convention for the measurement direction. If the same reference point were chosen as for the case of a negative stereo angle, then n would take on negative values. In order to avoid this, MCFast internally changes the reference point to the point labeled (x_1, y_1), at the upper left corner of the sensitive area. This is strictly internal to MCFast and is out of control of the user; the geometry file still requires that the point (x_0, y_0) be specified for `coord0_x` and `coord0_y`.

Now consider figures 4 and 5. In these cases the measurement direction is parallel to one of the edges of the detector and it rotates along with the plane.

The reference point is the point (x_1, y_1); these are the world coordinates of the position to which the point (x_0, y_0) is moved by the rotation.

Note, however, that the reference point which is stored in the hit structure is neither (x_0, y_0) nor (x_1, y_1) — it happens to be a point on the hit wire. The convention for that reference point will be described in another note.

One should notice that the cell size (or pitch) is not a template parameter. It is derived from the number of cells (`ncell`), which is a template parameter, and the size of the active area, which can be inferred from the template parameters. In all cases except for that of a type 1 layer with an arbitrary stereo angle, the computation of the cell size is straightforward. Figure 6 sets up the algebra for computing the cell size for the non-trivial case. The lower left-hand corner of the active region, point A, is at the lower edge of cell 0 and the upper right corner of the active region, point D is at the upper edge of cell `ncell-1`. The measurement direction is shown twice as two thick, red lines marked $\vec{\eta}$. Some of the strips are drawn perpendicular to $\vec{\eta}$. The points B and C are on a line which is parallel to the strips and which goes through the lower right corner of the active region. The cell size is given by,

$$\begin{aligned} \texttt{cell_size} &= \frac{|AB| + |CD|}{\texttt{ncell}} \\ &= \frac{|xdim \times \sin\theta| + |ydim \times \cos\theta|}{\texttt{ncell}} \end{aligned} \tag{2}$$

The figure shows the case that the stereo angle is less than zero but the final result holds for all values of the stereo angle.

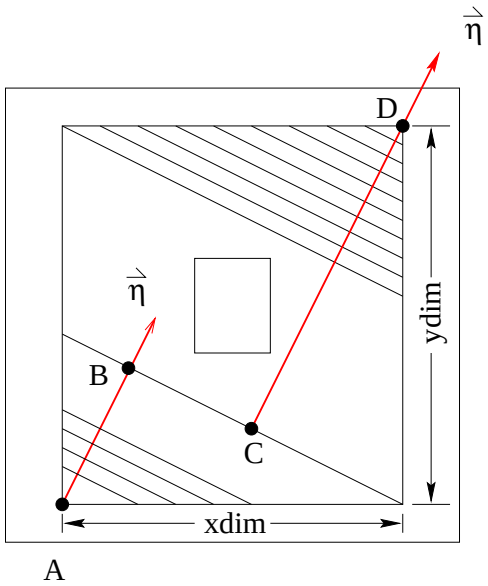


Figure 6: Figure showing how to compute `ncell` for `type=1` and a general stereo angle. The definition of the stereo angle is shown in figures 2 and 3. See the text for details.

2.8 Hadron calorimeter

The Hadron calorimetry structure is now obsolete.

2.9 Hits on track

Sets minimum number of hits required for an "offline" track

template **HitsOnTrack** (Max=2)

int **all** Minimum number of hits required in track fit

int **z** Minimum number of stereo hits – USER cut only

int **svx** Minimum number of silicon hits – USER cut only

end template

include file $MCFAST_DIR/inc/geom/track_minhit.inc

2.10 Materials

List of Materials and Mixtures

template **Material** (Max=50)

char **name** Name
real **a** A
real **z** Z
real **density** density [g/cm^3]
real **radlen** radiation length [cm]
real **abslen** absorption length [cm]
real **collen** collision length [cm]
real **dedx** dedx [GeV/cm]

end Material

include file $MCFAST_DIR/inc/geom/material.inc, material_struct.inc

Example of a material list

	name	A	Z	density	RadLen	AbsLen	C
Material	"VACU"	0.1000E-15	0.1000E-15	0.1000E-15	0.1000E+17	0.1000E+17	0
Material	"AIR"	0.1461E+02	0.7300E+01	0.1250E-02	0.3042E+05	0.6750E+05	0
Material	"ALUM"	0.2698E+02	0.1300E+02	0.2700E+01	0.8900E+01	0.3720E+02	0
Material	"IRON"	0.5585E+02	0.2600E+02	0.7870E+01	0.1760E+01	0.1710E+02	0
Material	"SILI"	0.2809E+02	0.1400E+02	0.2330E+01	0.9360E+01	0.3000E+02	0
Material	"SCIN"	0.1200E+02	0.6000E+01	0.1032E+01	0.4240E+02	0.7940E+02	0
Material	"CU"	0.6354E+02	0.2900E+02	0.8960E+01	0.1430E+01	0.1030E+02	0
Material	"TITA"	0.4790E+02	0.2200E+02	0.4540E+01	0.3560E+01	0.2750E+02	0
Material	"BE"	0.9010E+01	0.4000E+01	0.1848E+01	0.3530E+02	0.3670E+02	0
Material	"CARB"	0.1201E+02	0.6000E+01	0.2265E+01	0.1880E+02	0.3820E+02	0
Material	"CFIB"	0.1201E+02	0.6000E+01	0.2265E+01	0.2140E+02	0.4990E+02	0
Material	"MXSO"	0.5572E+02	0.2550E+02	0.6026E+01	0.2372E+01	0.2481E+02	0
Material	"ARGON"	39.95	18.0	1.78E-3	11706.	5.E+4	1
Material	"C2H6"	30.8	18.0	1.36E-3	34035.	5.E+4	1
Material	"MIX1"	0.5077E+02	0.2368E+02	0.4451E+01	0.3380E+01	0.3181E+02	0

2.11 Mixtures

Mixtures

template **Mixture** (Max=20)

char **name** mixture name

int **nmat** number of materials

material **matnames(5)** material names

real **prop(5)** proportion by volume

end template

include file $MCFAST_DIR/inc/geom/material.inc, material_struct.inc

2.12 Muon

Very Simple Muon Detector In v2_5_1 the muon detector is completely separate from the tracking and the muon_offline code estimates the amount of absorber and calorimeter material the muon sees and does a rough calculation of dedx. In v2_6 this will be more integrated into the tracking and tracing code.

template **MUON** (Max=40) cylindrical geometry

char **name**

char **shape** Shape "TUBE" or "CONE"

int **type** central = 1, forward or backward = 2

real **rmin(2)** minimum radius at lo z, hi z

real **rmax(2)** maximum radius at lo z, hi z

real **z0** central z position

real **zlen** length in Z

material **material** average material (not used)

int **nlayer** number of layers

end template

include file $MCFAST_DIR/inc/geom/muon.inc, muon_struct.inc

2.13 Silicon barrel detectors

Silicon barrel detectors are described as collections of planar wafers that are positioned such that their normal vector is in the xy plane. See figures 7 and 8. For purposes of tracking the wafers are treated as infinitesimally thin objects. However the description of the wafer does include a thickness, which is only used in the computation of the expected multiple scattering and energy loss; these are applied as if they occured at the infinitesimally thin plane. Also, the quantity β in the db file is not the same as the β which appears in the include file; see figure 8 and the discussion about include files, which begins on page 50. Finally, figure 9 gives a graphical representation of the connections between the components of the silicon barrel description.

template **SiBarrel** (Max=15)

int **num**

char **name**

int **nlyr**

child **LayerSiB**

real **z0_global** central z position for detector

end template

include file $MCFAST_DIR/inc/geom/si_barrel.inc, si_barrel_struct.inc

Silicon Barrel layers:

- Siblayers Layers for Silicon Barrel

  template **LayerSiB** (Max=30 total, max 30 per device)

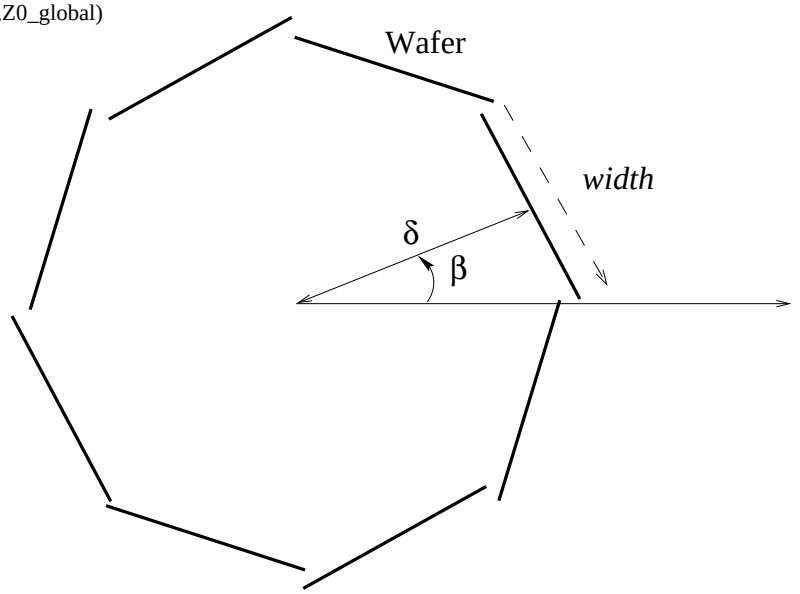
  int **det**

  parent **SiBarrel**

  int **lyr**

Silicon Barrel

Centered on (O,O,Z0_global)



Silicon Barrel input variables:

Beta is the angle in the x-y plane between the x axis and the center of the first wafer

Tilt is the rotatation angle of the wafer about the center point of the wafer

tilt equals zero when beta is perpendicular to the plane

tilt is positive for counterclockwise rotations - see next figure.

Delta is the distance from the origin to the wafer along beta

Width is the full width of the wafer

Z_length is full length of the wafer

Wafers can overlap if they are tilted.

The angle between the centers of adjacent wafers is 2π / nwafer

Figure 7: Silicon Barrel description, part 1 of 3. See the next figure for an important caveat about β.

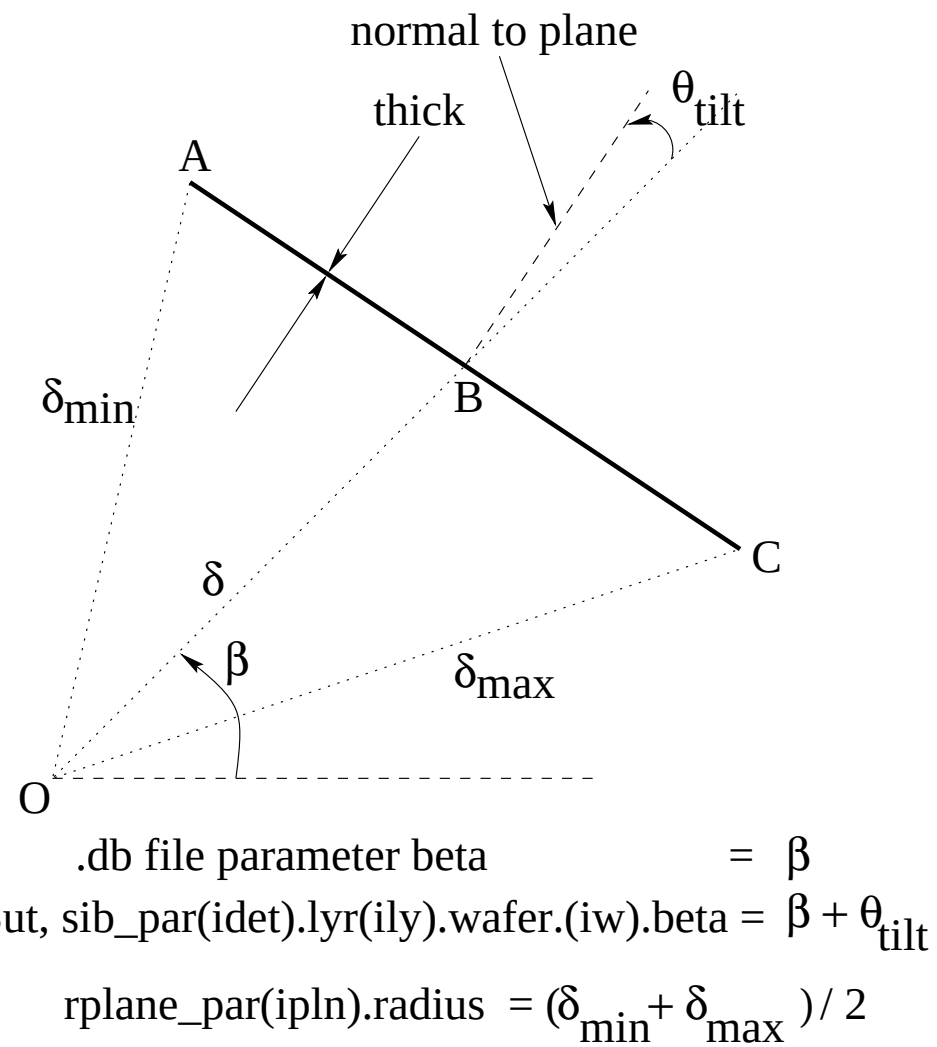


Figure 8: Silicon Barrel description, part 2 of 3. This figure shows additional elements found in the db file. It also shows some of the derived quantities which are stored in the silicon barrel and rplane include files.

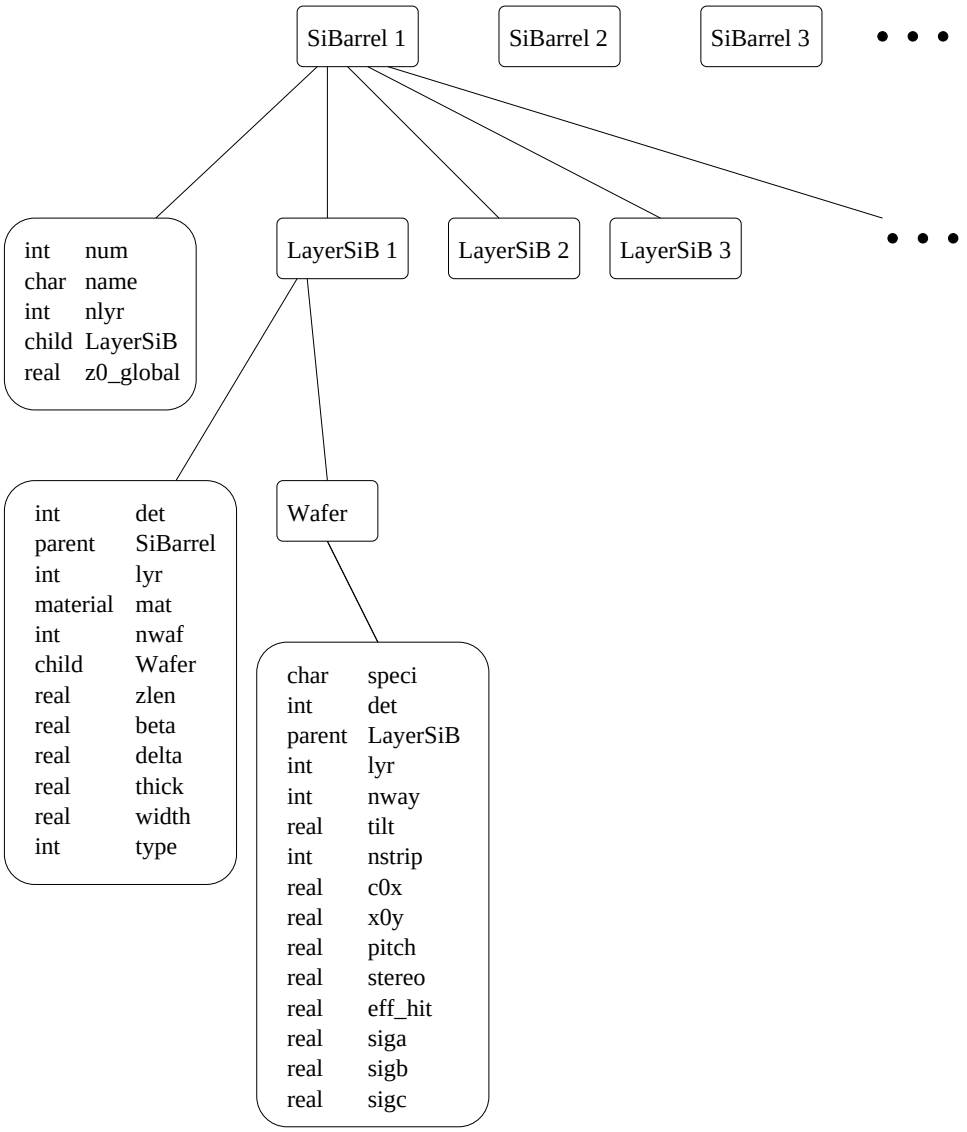


Figure 9: Silicon Barrel description, part 3 of 3. This figure shows a graphical representation of the various elements of the description of a silicon barrel detector.

material **mat** material of layer

int **nwaf** number of wafers in layer

child **Wafer**

real **zlen** full length in Z – all wafers in one layer have the same length

real **beta** phi angle from the x axis to the center of wafer(1) in the xy plane

real **delta** radial distance in x-y plane to the center of each wafer

real **thick** layer thickness

real **width** width of each wafer – wafers can overlap

int **type** type == 1= r/phi or stereo, 2 = z measurement only; you can use STEREO angle instead of type = 2

end template

include file $MCFAST_DIR/inc/geom/sib_lyr_struct.inc

Silicon Barrel Wafers:

– Wafer

Planes (or Wafers) that make up the layers in a silicon barrel detector. Wafers can overlap inside each layer.

template **Wafer** (Max=200 total in db file, Max = 110 per layer)

char **speci**

int **det**

parent **LayerSiB** Parent structure

int **lyr** Parent layer

int **nwaf** used only for speci=SPC, but ALWAYS present.

real **tilt** positive tilt is counterclockwise

int **nstrip** number of strips in layer

real **c0x** distance from midpoint of wafer to strip number 0 (measured quantity corresponding to strip 0 in local coordinates)

real **x0y** needed for z measurement– type = 2

real **pitch** pitch of strips

real **stereo** stereo angle in radians of strips – relative to ϕ measurement

real **eff_hit**

real **siga** resolution in cm – see below

real **sigb**

real **sigc**

$\sigma = \sigma_a + \sigma_b \times |\cos\theta| + \sigma_c \times cos^2\theta$

$\cos\theta = pz/p$

end template

include file $MCFAST_DIR/inc/geom/sib_plane_struct.inc

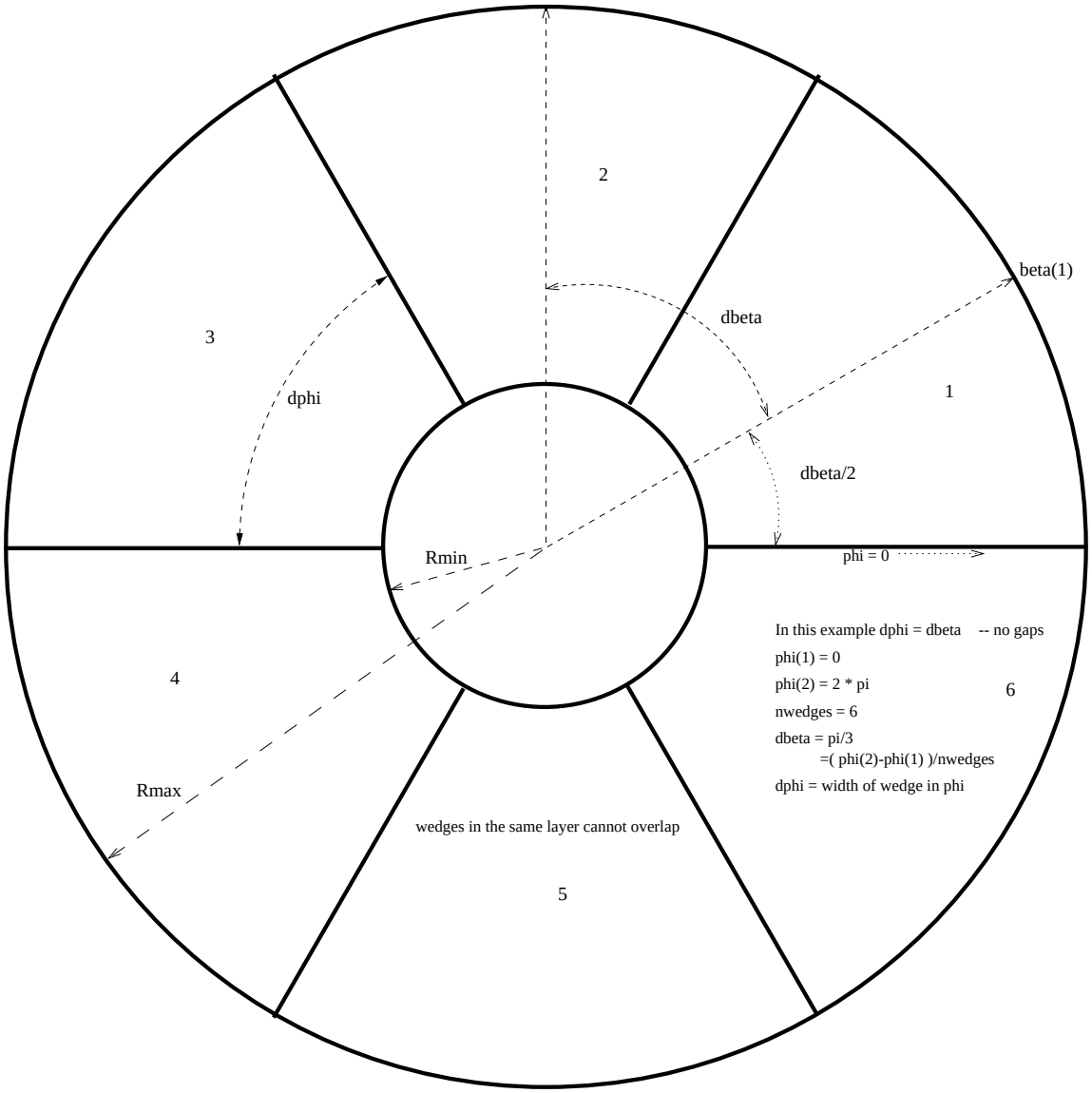


Figure 10: Silicon Disk description

2.14 Silicon forward disks

Silicon forward disks are described by layers of non-overlapping wedges. (See figures 10, 11 and 12.) The disks are perpendicular to the z axis.

template **SiDisk** (Max=32)

int **num**

char **name**

int **nlyr**

real **zpos** Z position of detector

end template

include file $MCFAST_DIR/inc/geom/sicir_disk.inc, sicir_disk_struct.inc

Layers in Silicon Disks Detectors:

- Sidisk_layer
  Layers that make up the forward disk detectors

  template **LayerSiDi** (Max=80 , max = 80 per device)

  int **det** pointer to SiDisk device number
  parent **SiDisk** Parent structure
  int **lyr** layer number
  material **mat** material ("SILI" for example)
  int **nwed** number of wedges in this layer
  child **Wedge** Wedge structure
  real **z_local** z of layer relative to detector Z position
  real **thick** thickness of silicon in layer
  real **rmin** minimum radius (for type 1-3; distance to edge for type 4)

real **rmax** maximum radius (for type 1-3: distance to edge for type 4)

real **phi(2)** ϕ coverage of layer (0,2π) for full coverage

real **dphi** ϕ coverage of one wedge: ($\phi(2) - \phi(1)$)/nwed for full coverage with no gaps. First wedge is centered at $\phi(1) + (\phi(2) - \phi(1))/2$nwed

int **type** measurement type
1 r measurement – code under development
2 phi measurement
3 U/V measurements– radial edges at rmin/rmax
4 U/V measurements– straight edges at rmin/rmax

end template

include file $MCFAST_DIR/inc/geom/sicir_lyr_struct.inc

Wedges of silicon disk layers:

– Sidisk_wedge
Wedges in one layer cannot overlap. Strip number is calculated relative to an outer corner of the disk and should be positive. For positive stereo ($0 <$ stereo $< \pi/2$) nstrip counts from the outer corner at low phi.

template **Wedge** (Max=80 total in db file, Max= 80 per layer)

char **speci** "SPC" do define each one or "ALL" for all

int **det** SiDisk number

parent **LayerSiDi** Parent structure

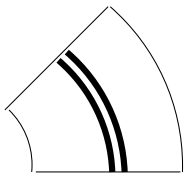
int **lyr** layer number

int **nwed** wedge number: only used for speci="SPC", but ALWAYS present

int **nstrip** number of strips on wedge

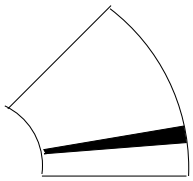
real **c0_r** – width of dead region in R
– will make symmetric dead regions at inner and outer edge

real **c0_phi** – width of dead region in phi
– will make symmetric dead regions at phi edges



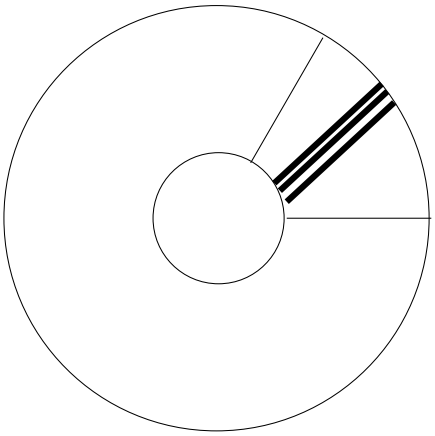
R measurement

Type = 1

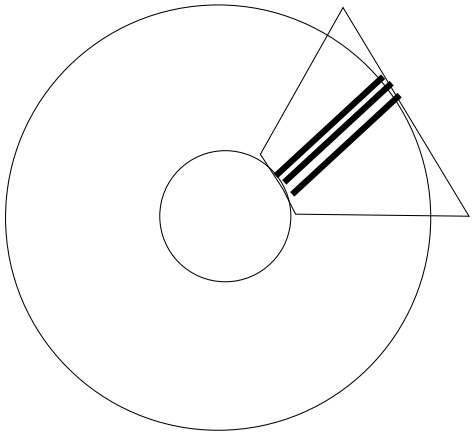


ϕ measurementf

Type = 2



Type = 3



Type = 4

U/V measrements

Figure 11: Silicon Disk Types

real **pitch** strip pitch
real **stereo** stereo angle in radians– for type 3 and 4 only
real **eff_hit** hit efficiency
real **siga** – type 2 in radians
real **sigb**
real **sigc** $\sigma = \sigma_a$ in cm for type 3-4, in radians for type = 2
end template
include file $MCFAST_DIR/inc/geom/sicir_wedge_struct.inc

At present only siga is used in trk_make_sidisk_hit.F.

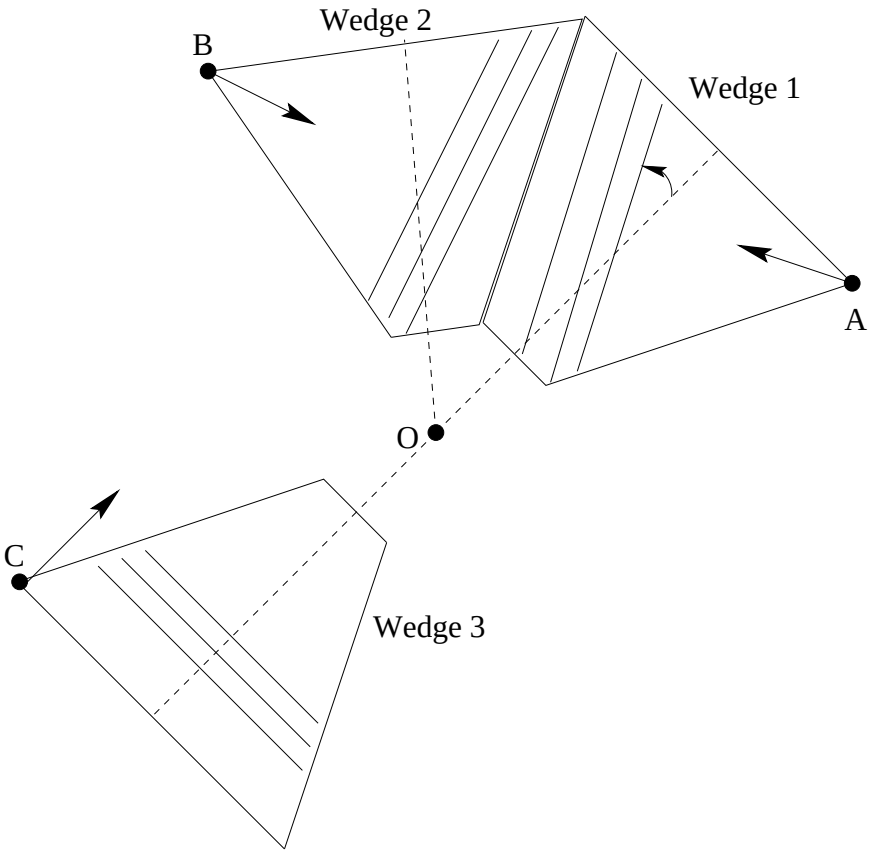


Figure 12: Coordinate systems for forward silicon detectors of wedge type 4. The figure shows cartoons of three wedges in the xy plane. The point labeled O is the beam spot. The dashed lines show the center lines of the wafers. The solid lines show several of the sense strips on the wafers. The curved arrow on wedge 1 shows the sense of a positive stereo angle, clockwise wrt the center line; wedge 1 shows an example of a positive stereo angle and wedge 2 shows an example of a negative stereo angle. Wedge 3 shows a stereo angle of $+90°$. The point A and the arrow attached to it show the origin of the coordinate system and the positive sense of measurements for wedge 1. Similarly the points B and C show, respectively, the origins and positive sense for wafers 2 and 3.

2.15 Silicon planes (xy)

Planar Silicon detectors are described by rectangular surfaces perpendicular to the z axis. These detectors can be strips or pixels. MCFast provides two device types to describe such planes, `SiXY` devices and `SiZRect` devices. The former device type is the simpler of the two and is described in this subsection. The latter is more general and is described in the next subsection.

template **SiXY** (Max=100)

int **num** station number

char **name** station name

int **nly** number of layers in each station

child **LayerSiXY**

real **zpos** z position of station

end template

include file $MCFAST_DIR/inc/geom/sixy_disk.inc, sixy_disk_struct.inc

Silicon xy plane layers:

- Sixy_layer

  template **LayerSiXY** (Max = 500 total, max = 100 per device)

  int **det** station id
  parent **SiXY**
  int **lyr** layer id
  material **mat** material
  real **z_local** local z position relative to station position
  real **thick** layer thickness
  real **stereo** stereo

int **type** type
real **eff_hit** efficiency
real **xmin** x min of layer
real **xmax** x max of layer
real **xmin_gap** x min of gap
real **xmax_gap** x max of gap
int **nsegm_x** number of strips or pixels in x
real **pitch_x** pitch of pixels in x
real **coord0_x** local coordinate 0 for x
real **sigma_x** resolution in x (see below for negative values)
real **ymin** y min of layer
real **ymax** y max of layer
real **ymin_gap** u min of gap
real **ymax_gap** y max of gap
int **nsegm_y** number of segments in y
real **pitch_y** pitch in y
real **coord0_y** local coordinate 0 for y coordinate
real **sigma_y** resolution in y (see below for negative values)

end template

include file $MCFAST_DIR/inc/geom/sixy_disk_layer_struct.inc

If either sigma_x or sigma_y is negative, then the user routine usr_siz_rect_res.F will be called to compute the measurement and the error on the measurement. This hook can be used, for example, to generate clusters of hit strips rather than just throwing a gaussian resolution.

It is not possible to declare rotated sixy layers; this functionality, however, is provided for SizRect devices, which are described in the next subsection.

2.16 Silicon Rectangular Z-Planes (SiZRect)

Planar Silicon detectors can also be described by rectangular surfaces (silicon wafers) perpendicular to z axis but not necessarily centered about z axis. One more wafers forms a layer and one or more layers forms a station. For the purpose of tracking, wafers are treated as infinitesimally thin objects. However the description of the wafer does include a thickness, which is only used in the computation of the expected multiple scattering and energy loss; these are applied as if they occured at the infinitesimally thin plane.

template **SizRect** (Max=100)

int **num** station number

char **name** station name

int **nlayers** number of layers in the station

child **LayerSizRect**

real **xpos** x-position of station's center (in global coordinate system)

real **ypos** y-position of station's center (in global coordinate system)

real **zpos** z-position of station's center (in global coordinate system)

end template

include file $MCFAST_DIR/inc/geom/siz_rect.inc, siz_rect_struct.inc

Silicon rectangular layers:
Layers that make up the rectangular silicon detectors. A layer may have a rectangular inner gap. The center of the gap must coincide with the center of layer. Three types of gaps are supported, X, Y, and window; these are illustrated in figure 13. A layer may be rotated about the z axis. For a layer with a window-type gap, wafers are not permitted to overlap each other. For all other gap types, including no gap, wafers within a layer may overlap in x and/or y. The MCFast wafer numbering convention is that the wafer 1 is in the upper left corner of the layer. Wafer numbers then increase across the layer in x and then down in y. Wafers may be offset from the average z-position of the layer; see figure 14 and the description of wafers, below.

- Siz_rect_layer

  template **LayerSizRect** (Max=500, max=20 per device)

  int **det** station id

  parent **SizRect**

  int **lyr** layer id

  int **nWaferX** number of segments along x-direction

  int **nWaferY** number of segments along y-direction

  real **zLayerLocal** local z-position of layer relative to station position

  real **rotation** rotation of layer about z axis; see text for details.

  real **Xlen** layer full length along x

  real **Ylen** layer full length along y

  real **GapXlen** full width along x of gap in layer (see note below)

  real **GapYlen** full width along y of gap in layer (see note below)

  char **StaggeringPattern** X, Y, CH, or NONE. See below.

  end template

  include file $MCFAST_DIR/inc/geom/siz_rect_layer_struct.inc

A full description of the staggering pattern is given after the wafer template has been described.

To implement a gap one should specify:

X: GapXlen>0 && GapYlen=Ylen

Y: GapXlen=Xlen && GapXlen=Xlen

Window: 0<GapXlen<Xlen && 0<GapYlen<Ylen.

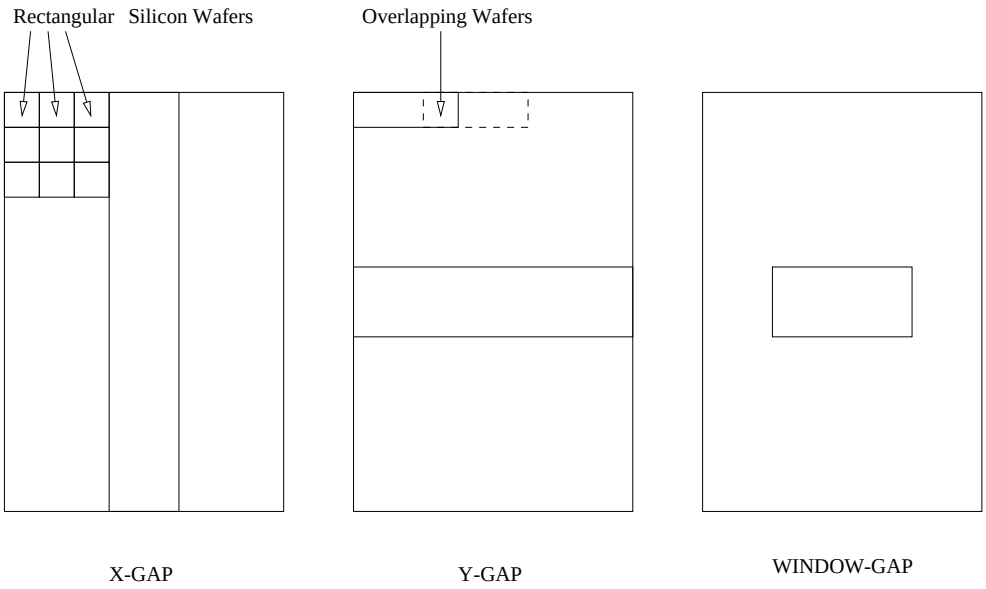


Figure 13: Rectangular Silicon Layers with Gaps (front view)

Silicon rectangular wafers:

- Siz_rect_wafer

  template **WfrSizRect** (Max=500, max=200 per layer)

  char **speci** ALL, SPC or ONE; see below.

  int **det** station id

  parent **LayerSizRect**

  int **lyr** layer id

  int **waferX** wafer id along x (in layer coordinate system); ignored for speci=ALL; should be non-zero for speci=SPC.

  int **waferY** wafer id along y (in layer coordinate system); ignored for speci=ALL; should be non-zero for speci=SPC.

  int **type** 1=pixel, 2=strips

  material **mat** material (for example, "SILI")

  real **eff_hit** efficiency

  real **stereo** orientation of strips wrt x axis; see below.

  real **Xlen** length of wafer along x

  real **Ylen** length of wafer along y

  real **Zlen** z-thickness of wafer

  real **zOffset** z offset for staggering pattern. See below.

  int **nsegm_x** number of strips or pixels along x

  real **pitch_x** pitch of strips of pixels in x

  real **coord0_x** local coordinate of first strip of pixel for x

  real **sigma_x** resolution in x (see below for negative values)

  int **nsegm_y** number of strips or pixels along y

  real **pitch_y** pitch of strips of pixels in y

  real **coord0_y** local coordinate of first strip of pixel for y

  real **sigma_y** resolution in y (see below for negative values)

  end template

  include file $MCFAST_DIR/inc/geom/siz_rect_wafer_struct.inc

This template describes wafers which make up layers. If `speci=ALL` then MCFast will fill the layer with copies of this wafer definition; this is done according to nWaferX, nWaferY, the gap size and the staggering pattern, all of which are defined in the layer template. If any part of a wafer intrudes into the gap, it is omitted from the pattern. Also

If `speci=SPC` then this wafer definition is used to define exactly one wafer, the one specified by waferX and waferY. If a layer is composed of some repeated wafers plus a few special ones, the `speci=ALL` wafer must appear first in the `.db` file, followed by the `speci=SPC` wafer(s).

With v5_0 there is a new value, `speci=ONE`. Wafers with this value are treated the same as ALL except that the wafers are permitted to overlap the central gap/hole; this value of speci can be used to clone sixy devices by making layers with one big wafer.

For `speci=ALL` wafers, the z-position of the center of wafer 1, relative to the nominal z of the layer, is specified by the zOffset parameter. See figure 14. The z-positions of the remaining wafers in the layer are calculated using zOffset and the StaggeringPattern specified for that layer. The staggering patterns are:

X The first **column** is offset by `+zOffset`, the second by `-zOffset`, ...; that is, the pattern alternates in X.

Y The first **row** is offset by `+zOffset`, the second by `-zOffset`, ...; that is, the pattern alternates in Y.

CH A checkered board pattern. The wafer in the upper left corner is offset by `+zOffset`, its lower and right neighbours are both offset by `-zOffset`, and so on.

NONE All wafers in the layer are at `+zOffset`.

In case of `speci=SPC` the wafer z-position is defined directly by zOffset.

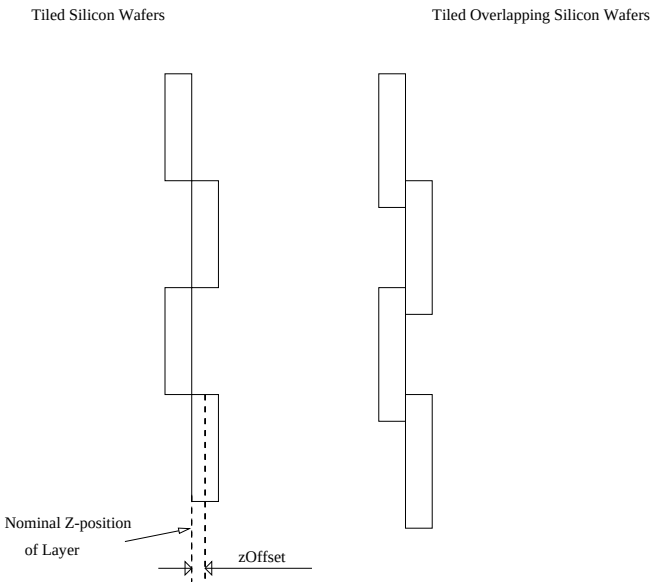


Figure 14: Offset Silicon Wafers (side view)

If either sigma_x or sigma_y is negative, then the user routine usr_siz_rect_res.F will be called to compute the measurement and the error on the measurement. This hook can be used, for example, to generate clusters of hit strips rather than just throwing a gaussian resolution.

Figures 15 and 16 show the meaning of some of the parameters used to position wafers within a layer. In particular, figure 15 makes the distinction between rotation angles and stereo angles. For convenience, all of the positioning parameters, except for the rotation angle itself, are defined for an unrotated layer. In figure 16 the outer rectangle marks the physical size of the wafer while the inner rectangle marks the active area of the wafer; the widths of the horizontal and vertical dead areas are allowed to be different. The active area is centered on the full area of the wafer. The point marked (x_0, y_0) corresponds to `(coord0_x,coord0_y)`, which defines the lower left hand corner of the active area.

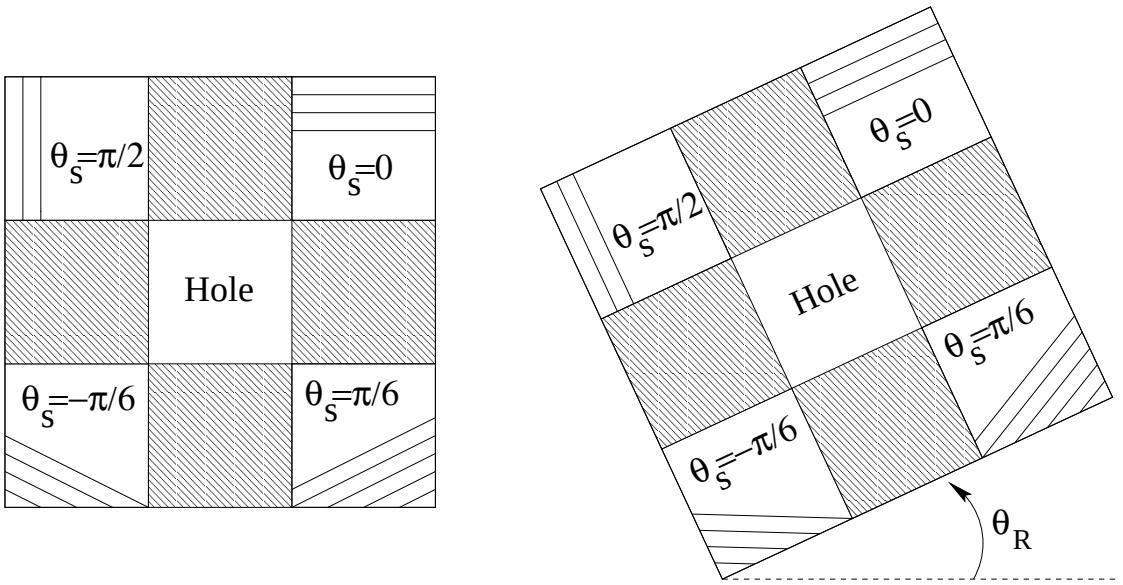


Figure 15: This figure shows the stereo angle, θ_s, which is defined in the wafer template, and the rotation angle, θ_R, which is defined in the layer template.

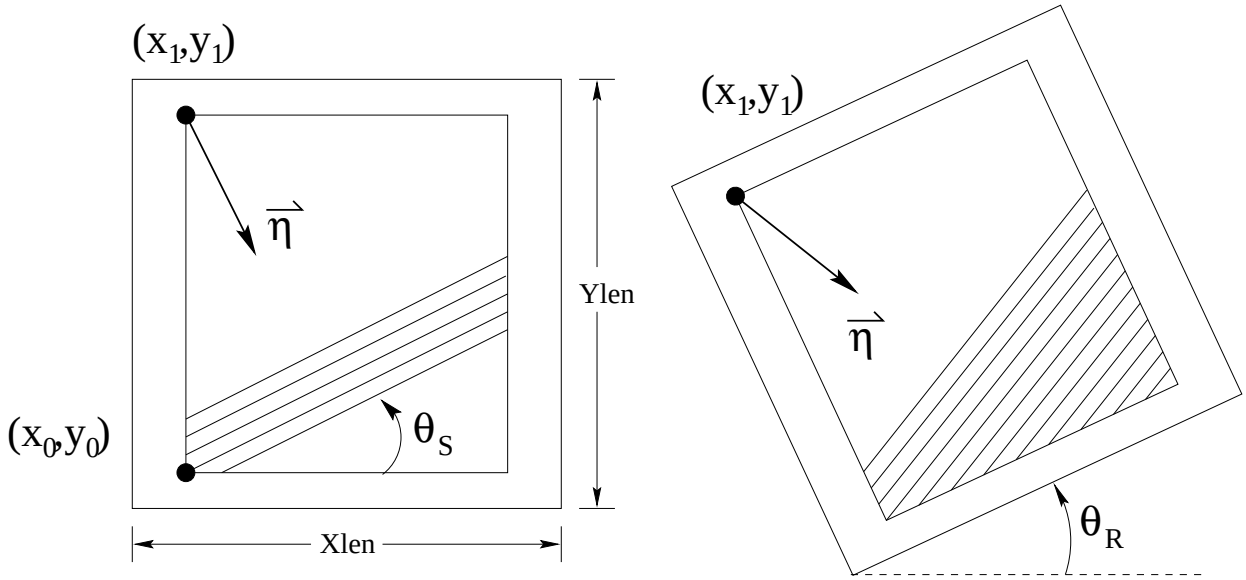


Figure 16: The left hand figure shows a wafer in an unrotated position while the right hand figure shows the same wafer in its rotated position. For convenience, the positioning parameters are defined in the unrotated configuration.

The points marked (x_1, y_1) and the vector $\vec{\eta}$ are of concern only to experts. These are defined similarly to the corresponding quantities for forward tracking detectors.

2.16.1 Stereo angles for SiZRecT Devices

There are some restrictions when specifying stereo angles for SizRect devices.

As of version v5_0, stereo angles and rotation angles only work properly for strip SizRect devices. Pixel devices must have `stereo=0.0` and `rotation=0.0`. This will be fixed in a future release.

It is necessary that the dimensions of the active area, the pitch and the number of pixels in each dimension be self consistent. MCFast does not check for this.

There is a special case which is important. If one specifies, `nsegm_x>1`, `nsegm_y=1` and `stereo=0`, then MCFast will assume that the user really intended to measure the local x coordinate; therefore the routine which makes SizRect hits will use `stereo=-`π`/2`. Without this trick, the user would have to specify a lot of digits in `stereo` in order to get a true x measurement.

For a general stereo angle, it is required that `nsegm_x>1`. If one specifies `nsegm_x=1`, `nsegm_y>1` and `stereo`$\neq$`0.0`, the program will print an error message and stop. See the discussion on stereo angles in subsection 2.7 for for the algebra needed to keep `nsegm_x` and `pitch_x` consistent with the dimensions of the active area. In addition figure 6 may be useful.

2.17 Solenoid magnet

template **Solenoid** (Max=10)

char **name** Name

real **bfield** Magnitude of B -field

real **rmin** minimum radius of solenoidal field (should be 0.0)

real **rmax** maximum radius of solenoidal field

real **z0** Central z position of field region

real **zlen** length in Z

material **mat_fill** material inside solenoid (not used)

real **thick_boun(4)** boundary thickness at rmin=0 , rmax(coil), zmin =0, zmax =0

material **mat_boun(4)** material at boundary rmin=' ', rmax(coil material), zmin=' ', zmax=' '

include file $MCFAST_DIR/inc/geom/solenoid.inc

end template

3 The Include files

3.1 Silicon barrel detectors

The include files listed below show the structures which describe a silicon barrel detector; see figures 7 and 8. Recall that the parameter β means one thing in the .db file and another in the data structures below. (The description of the .db file begins on page 27.) Finally, figure 17 shows the relationships between the structures which describe a complete silicon barrel detector.

include file: $MCFAST_DIR/inc/geom/si_barrel.inc, si_barrel_struct.inc

structure /si_barrel_struct/ (Max=15)

integer **numlyr** Number of layers

integer **nchan** Total number of channels

record /sib_lyr_struct/ **lyr(max_sib_lyr)** Layer parameters

character*40 **name** Name of device

real **z** Global z-position of center of device

end structure si_barrel_struct.inc

template file: $MCFAST_DIR/db/sibarrel.db

include file: $MCFAST_DIR/inc/geom/sib_lyr_struct.inc

structure /sib_lyr_struct/ (Max=30)

integer **numplane** Number of planes in the layer

integer **nchan** Number of channels in this layer

integer **chan0** Channel number of 1st strip

record /sib_plane_struct/ **plane(max_sib_plane)** Parameters of each plane in the layer

end structure

template file: $MCFAST_DIR/db/siblayers.db

include file: $MCFAST_DIR/inc/geom/sib_plane_struct.inc

structure /sib_plane_struct/ (Max=110)

integer **type** 1 = r-phi measurement; 2 = z measurement
real **dmin** dmin of plane (in r-phi direction)
real **dmax** dmax of plane (in r-phi direction)
real **zmin** zmin of plane
real **zmax** zmax of plane
real **beta** Angle between normal to plane and x axis
real **xpos(3)** Center of wafer.
real **eta(3)** Direction cosines of normal to plane
real **delta_min** Minimum distance from origin to edge of plane
real **delta_max** Maximum distance from origin to edge of plane
real **thick** Thickness of plane
real **tilt** Tilt angle for plane
real **eff_hit** Hit efficiency
real **siga** Constant a in resolution formula
real **sigb** Constant b in resolution formula
real **sigc** Constant c in resolution formula
integer **material** Material composing plane
real **coord0_x** Local x-coordinate of strip 0
real **coord0_y** Local y-coordinate of strip 0
real **pitch** Spacing of strips
integer **nstrip** Number of strips in this plane
integer **nchan** Number of channels in this plane

integer **chan0** Channel number of the 1st strip in plane

real **stereo** Stereo angle in rad

real **cos_stereo** cos(stereo)

real **sin_stereo** sin(stereo)

end structure

template file: $MCFAST_DIR/db/wafer.db

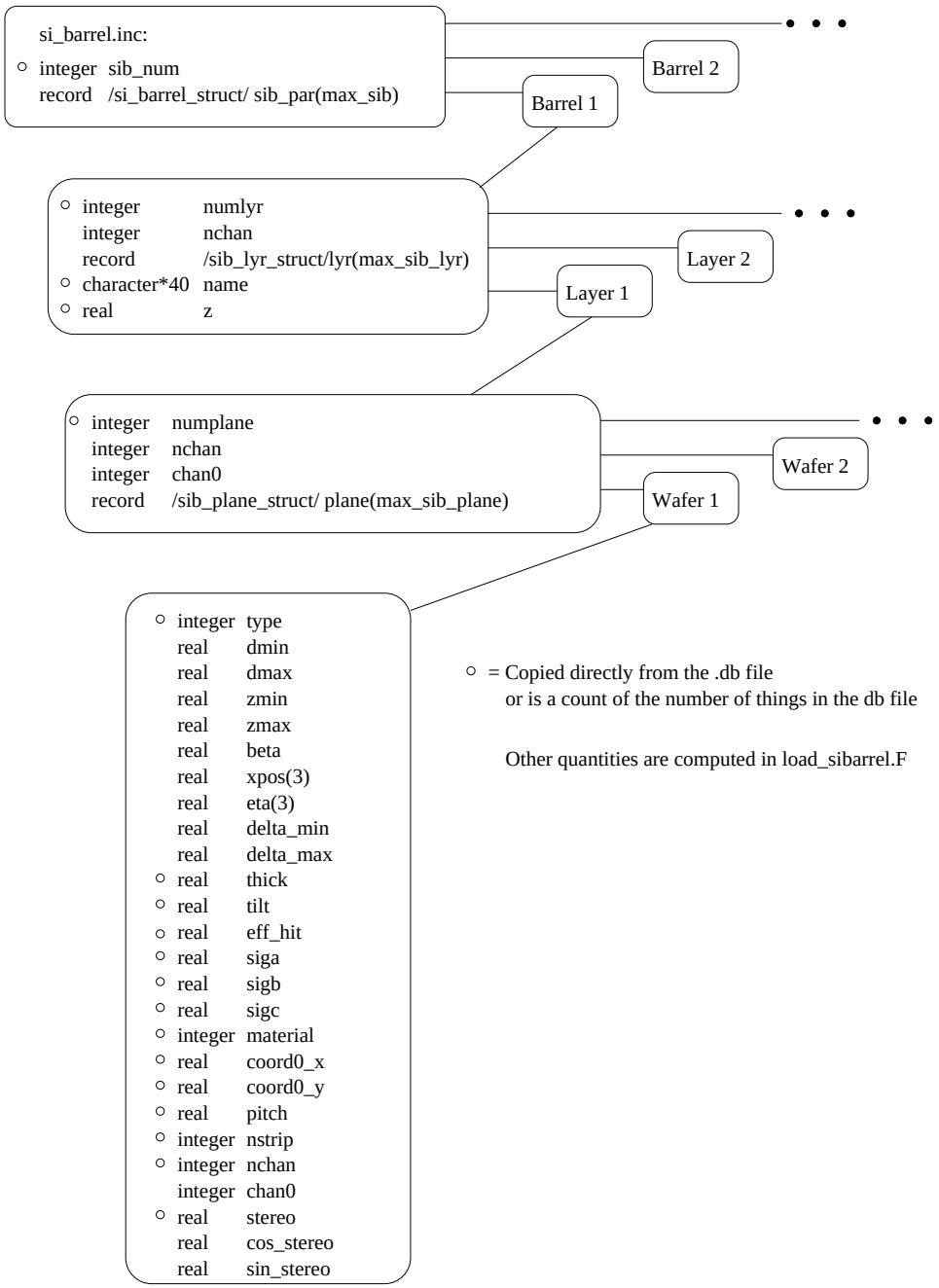


Figure 17: Silicon Barrel description. This figure shows the relationships between the elements which describe a silicon barrel detector.