Report

# Simulation of a Silicon-Strip Detector

*Author:*
Bernadette Kolbinger
a0600121

*Supervisor:*
Markus Friedl

November 29, 2012

# Contents

# 1 Introduction

## 1.1 Purpose

Based on an existing C-program which calculates electric fields in a Silicon-Strip-Detector, the algorithm is going to be improved by additional components (e.g. magnetic field, diffusion). The charge collection is going to be simulated on the scale of single charge carriers to get results which can be compared to real measurements. The model should contribute to optimize the geometric setup of Silicon-Strip-Detectors.

This internship contributes to several ongoing research projects which develop silicon detectors for experiments in particle physics.

## 1.2 Project Description

For more than a decade, the Institute of High Energy Physics has been researching in the area of semiconductor detectors. Contributing to several experiments at CERN and other research centers, we have acquired an excellent reputation. Currently we are working on silicon detectors for the setup of the Belle-II-Experiment in Japan and for an Upgrade of CERN's CMS-Experiment. For both, we have developed silicon detectors.

Now we want to strengthen our expertise in the experimental field by simulations of such detectors.

# 2 Background

## 2.1 Silicon Strip Detector - Principle of Operation

Basically, semiconductor detectors work like a p-n junction operated at reverse bias.

Without an external applied voltage, electrons near the p-n interface are likely to diffuse into the p-region leaving positively charged ions in the n-region behind and vice versa. As a result in the area around the p-n interface, a space charge region forms, called the depletion zone.

If the p-type region is now connected to the negative and the n-type region to the postive terminal (reverse bias), the holes and the electrons will be pulled to the electrodes and the depletion region will broaden. Because of this, the diode forms a high resistance to current flow.

The width of the depletion zone is given by:

$$w_{depl} = \sqrt{2\varepsilon\mu r V_{bias}} \tag{1}$$

where $\mu$ is the mobility of the majority carriers (electrons in case of n-type bulk), $r$ the resistivity of the bulk material and $V_{bias}$ the bias voltage.

Increasing $V_{bias}$ will increase the depleted volume in the bulk and therefore the region in which a particle might be detected.

However, increasing $V_{bias}$ also means increasing the electric field in the detector. If the field strength exeeds a critical level, the depletion zone will break down and a high current will flow.

The principle of operation of a silicon strip detector is shown in Figure 1. The Silicon bulk is lightly n-doped and the junction is formed by highly doped p+ strips. Each strip forms a pn-diode. The backplane is highly doped in the same manner as the bulk (n+ type in this case). Usually, detector diodes are doped asymmetrically but not necessarily. In the bulk between the two electrodes an electric field is present due t othe applied voltage.

Because the detector is operated at reverse bias, no current flows (except for a very low leakage current). However, if an ionizing particle traverses the detector, an equal number of free electrons and holes is produced. The electric field causes the charges to travel to the electrodes. It is important to stress that a signal is induced in the electrodes as soon as the charges start to move and not only at the time when the first charge hits the electrodes.

The number of electron-hole-pairs depends on the energy of the particle and on the material it passes through. We will only consider MIPs (*Minimum Ionizing Particles*) which have the lowest possible energy loss (see Bethe-Bloch-Formula). In High Energy Physics, essentially all measured particles are minimum ionizing.

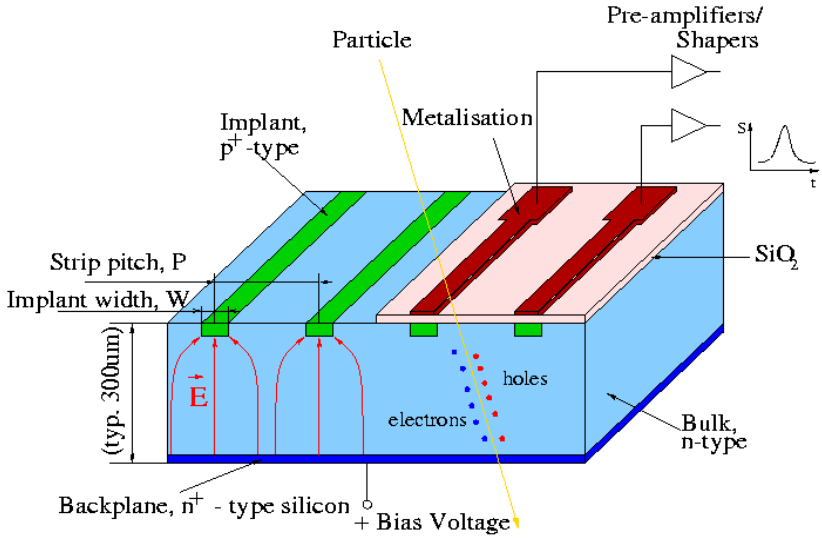



Figure 1: Principle of operation of a silicon strip detector [1].

## 2.2 Potentials inside the Detector

In oder to determine the potential in the whole detector, one needs to solve Poisson's equation:

$$\Delta\phi = -\frac{\rho}{\varepsilon} \tag{2}$$

with the boundary conditions of strips and backplane. $\Delta = \nabla^2$ represents the Laplace Operator, $\phi$ the potential, $\rho = Nq_e$ the charge density with concentration of doping $N$ and $\varepsilon$ the dielectric constant. The resulting potential is called the drift potential and the corresponding electric field is calculated by $\mathbf{E} = -\nabla\phi$.

The weighting potential is obtained by solving Laplace's equation:

$$\Delta\phi = 0 \tag{3}$$

Here, only the read-out electrode strip ist set to unity potential and all the other electrodes are set to zero potential. In the program, these two differential equations will be solved numerically, see chapter 3.1.

## 2.3 Charge Collection

In the presence of an **E**-field, the electrons and holes will drift along the field with an average velocity of [2]:

$$\mathbf{v} = \mu\mathbf{E} \tag{4}$$

where $\mu$ denotes the mobility of the charge carrier which is different for electrons ($1350\,\mathrm{cm}^2$ / Vs) and holes ($450\,\mathrm{cm}^2$ / Vs) [2]. Note that the drift velocity only depends on the electric field and not on the time during acceleration because the carriers interact with the lattice of the silicon crystal.

Carrier transport can also occur due to diffusion. So the motion of the carrier is a superposition of the drift because of the electric field and a random walk due to thermal motion.

We will use a simple model of charge collection [3] to describe the currents of electrons and holes in the detector. We will assume that the charges will be equally distributed along the particle's path through the detector. Because eq. 4 only holds for small electric fields, we will use an empirical relation for the velocities of electrons and holes are given by [3]:

$$\begin{aligned} v_e &= \frac{\mu_e E}{\sqrt{1 + \left(\frac{\mu_e E}{v_{e,sat}}\right)^2}} \\ v_h &= \frac{\mu_h E}{1 + \frac{\mu_h E}{v_{h,sat}}} \end{aligned} \tag{5}$$

where $v_{e,sat}$ and $v_{h,sat}$ are electron and hole saturation velocities, respectively (Figure 2).

The depletion voltage $V_{depl}$ is the voltage resulting in $w_{depl} = d$ with the detector thickness $d$. In this case, the electric field is zero at the backplane and increasing linearly to its maximum at the junction [3]:

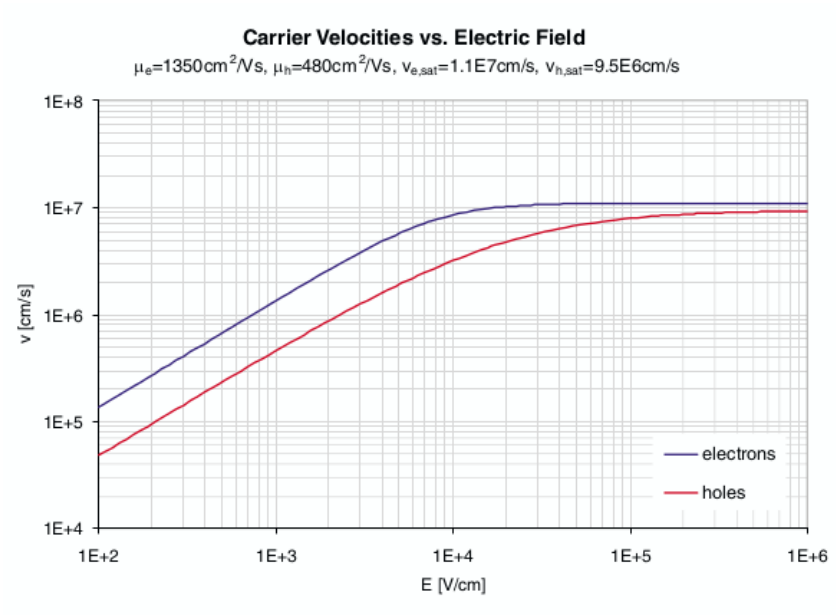



Figure 2: Empirical relation between carrier velocities and the electric field in silicon [4].

$$E_{max} = \frac{eNd}{\varepsilon} \tag{6}$$

For the distribution of the electric field of a reverse-biased diode, see Figure 3. The left side shows the case of partial depletion with $V < V_{depl}$ where the region of space charge does not fill the whole bulk and charge charge collection will not be efficient. The right side shows the electric field for $V > V_{depl}$ (overbias) where a uniform offset, given by $(V - V_{depl})/d$, adds to the triangular electric field.

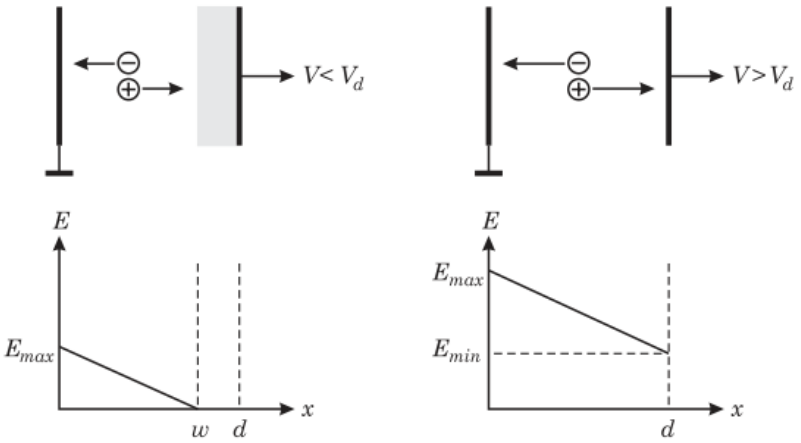
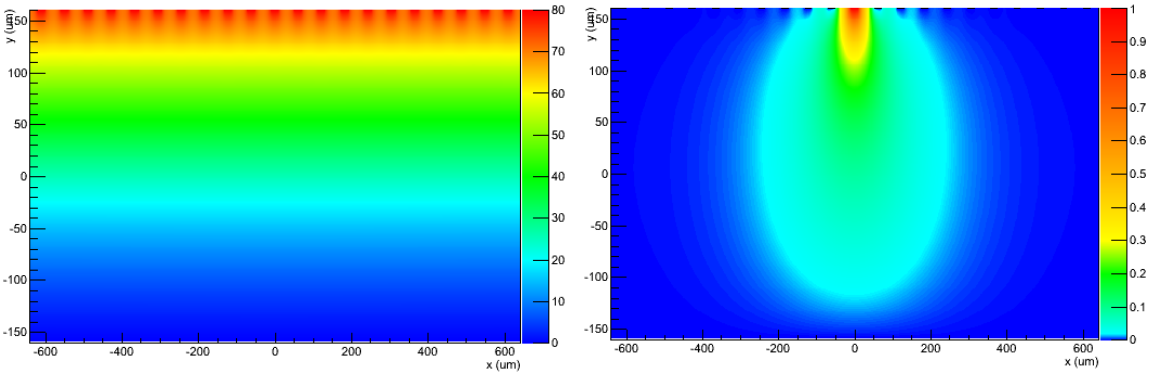



Figure 3: Distribution of the electric field for $V < V_{depl}$ and $V > V_{depl}$ [2].

The current which is induced by a charge $q$ on an electrode $k$ is now given by [2]:

$$i_k = -q\mathbf{v} \cdot \mathbf{E}_w \tag{7}$$

where $\mathbf{E}_w$ denotes the weighting field which is determined by setting the eletrode k to unity potential and all others to zero potential. The weighting field depends on the measuring electrode and determines how charge motion couples to a specific electrode [2]. Figure 4 shows drift and weighting potential of a strip detector. Only in a parallel plate configuration (single strip opposite of the backplane), drift and weighting field are of the same form, such that the latter does not need to be considered explicitly.




(a) Drift potential (b) Weighting potential

Figure 4: Drift and Weighting potentials of a two-dimensional strip detector with: lateral extent=1000 µm, thickness=300 µm, strip pitch=50 µm and strip width=20 µm, p-type bulk and n-type strips.

The current induced by all moving charges is given by [5]:

$$i = -e\mathbf{E}_w \left( \sum_i \mathbf{v}_{i,e} + \sum_i \mathbf{v}_{i,h} \right) \tag{8}$$

where $d$ is again the detector thickness. By integrating the induced current $i$ over time as it is done by a charge-sensitive amplifier), one gets the total collected charge.

## 2.4 Lorentz shift

With a magnetic field orthogonal to the electric field in the detector, the charge carriers will no longer drift straight to the electrodes but will be deflected due to the Lorentz force as shown in Figure 5. Their drift direction will be shifted by the Lorentz angle $\theta_L$. Since the velocities of electrons and holes are different, the Lorentz force affecting them, $\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B})$, will not be the same resulting in different Lorentz angles for electrons and holes [6]:

$$\tan \theta_{L,i} = \frac{\Delta x_i}{d} = \mu_i B \tag{9}$$

with the Hall mobility $\mu$, absolute value of the magnetic field $B$, Lorentz shift $\Delta x$ and $i$ being either $e$ (electrons) or $h$ (holes).

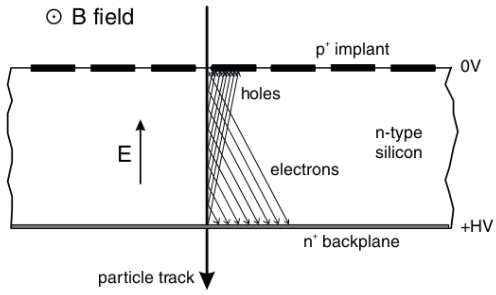



Figure 5: Lorentz shift [7]. Note that electrons and holes travel at a different angle. The sign of the B field results in a drift either to the right or left.

## 2.5 Thermal Diffusion

As already mentioned, carrier movement may also include random thermal diffusion. From the equipartion theorem, we can derive the mean thermal velocity:

$$v = \left(\frac{3k_bT}{m_{\text{eff}}}\right)^{1/2} \tag{10}$$

with Boltzmann's constant $k_b$, the temperature $T$ and the effective mass $m_{\text{eff}}$. We assume that the charge carriers' velocities are Boltzmann distributed [8]:

$$P(v) = \frac{m_{\text{eff}}v}{k_bT}e^{-m_{\text{eff}}v^2/2k_bT} \tag{11}$$

In order to get an absolute value of the carrier velocity of the random walk, we will generate uniformly distributed random numbers and convert them to random numbers $\xi$ distributed according to Equation 11. This is done by [8]:

$$v_{abs} = \left(\frac{2k_bT}{m_{\text{eff}}}\ln\left|\frac{1}{1-\xi}\right|\right)^{1/2} \tag{12}$$

Additionaly, we need a random direction for the movement and this is realized by a vector:

$$\mathbf{d} = \begin{pmatrix}\cos\theta \\ \sin\theta\end{pmatrix}$$

with a uniformly distributed random angle $\theta \in [0, 2\pi]$. Thus, the x and y components of the diffusion velocity become $\mathbf{v}_x = v_{abs}\cos\theta$ and $\mathbf{v}_y = v_{abs}\sin\theta$.

# 3 The Program

## 3.1 Potentials

The program is implemented in the ROOT framework which was developed at CERN.

The implementation of the potential calculation can be found in the files `Potentials.h` and `Potentials.cxx`.

In oder to calculate the drift and the weighting potential, we first have to spatially discretize our two-dimensional detector, as shoen in Figure 6. We choose a grid distance $\Delta h$ of 1 µm for both x and y axes.

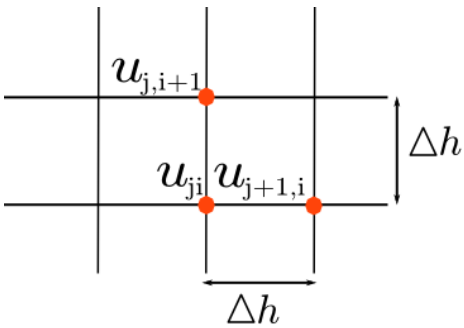



Figure 6:

To discretize Poisson's equation $\Delta u = -f$, the spatial derivatives of the potentials need to be approximated. This is easily done by using the Taylor-approximation for $u(x_{j\pm1}) = u(x_j \pm \Delta h)$ (similar for the y coordinate), resulting in [9]:

$$\Delta u(x_j, y_i) = \frac{1}{\Delta h}[u(x_{j+1}, y_i) + u(x_{j-1}, y_i) + u(x_j, y_{i+1}) + u(x_j, y_{i-1}) - 4u(x_j, y_i)] + \mathcal{O}(\Delta h^2) \tag{13}$$

The discretized Poisson equation now becomes:

$$u_{j+1,i} + u_{j-1,i} + u_{j,i+1} + u_{j,i-1} - 4u_{j,i} = -(\Delta h)^2 f_{j,i} \tag{14}$$

where $u_{j,i}$ denotes $u(x_j, y_i)$. An iteration method is used to solve this equation. Equation 14 is solved for $u_{j,i}$ and then used as an approximation in the next step. So one iteration step ($k \to k+1$) reads:

$$u_{j,i}^{k+1} = \frac{1}{4}\left(u_{j+1,i}^{(k)} + u_{j-1,i}^{(k)} + u_{j,i+1}^{(k)} + u_{j,i-1}^{(k)}\right) + \frac{(\Delta h)^2}{4} f_{j,i} \tag{15}$$

The problem with this iteration method is, that in order to get a good approximation for the potentials, one needs a lot of iteration steps and therefore a lot of time. The computing time increases with $N \times M$ with $N$ and $M$ being the total number of gridpoints. With a typical detector width of 1000 µm and thickness of 300 µm, the calculation can easily take several hours on a modern PC.

A multigrid approach is used which drasticaly reduces the necessary steps to a good approximation. Figure 7 shows how this approach is implemented in the program.

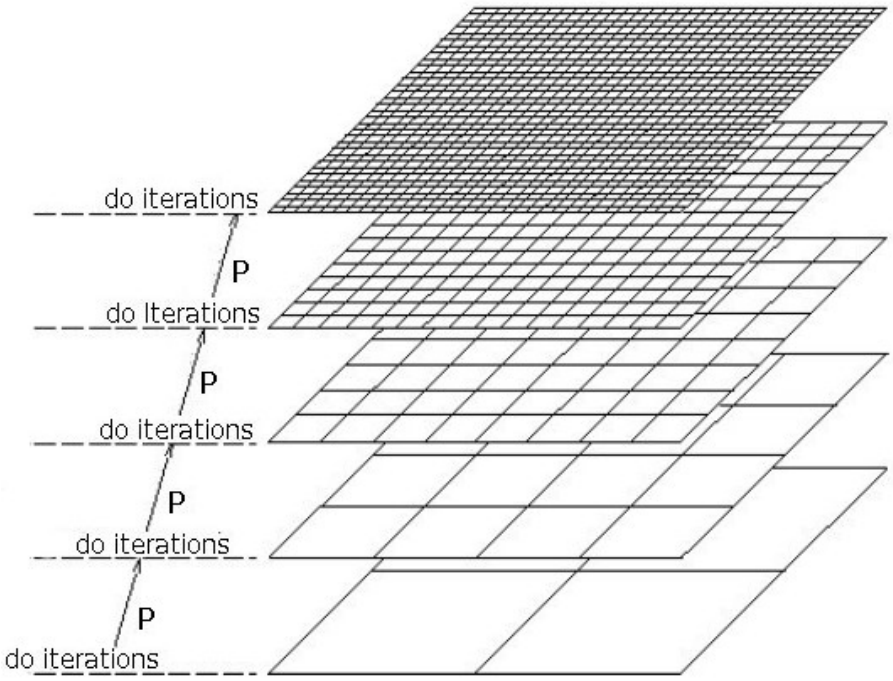



Figure 7: Multigrid method [10].

Instead of doing the calculation only on the $\Delta h = 1\,\mu\text{m}$, the potentials will be calculated on a coarser (or several coarser) grid(s) first with $\Delta h_{coarse} = 2\Delta h_{fine}$ and then transfer the results to shared points of the finer grids and interpolate for intermediate points.

Concretely, the algorithm in the weightfield program operates as follows:

1. The initialization starts on the finest grid. A two dimensional array is created according to user input of detector thickness (`YMAX`), width (`XMAX`), pitch (`pitch` ) and strip width (`width`). Because it is convenient to have one strip (readout electrode for subsequent current calculations) exactly in the middle, `XMAX` is going to adapted if the total number of strips (`count`) in the detector would be an even number according to the user input. All this happens in the constructor of the Potentials class.

2. Then, the electrodes are reset and potentials set according to user input. For the drift potential, the bottom electrode is set to bias voltage (`vbias`) for p-type strips and to 0 otherwise. Likewise, the strips are set to 0 or the bias voltage. For the weighting potential, only the central strip ist set to 1, all the other strips as well as the backplane are set to zero. Furthermore, an additional array (`fix`) of the same size as the potential array is created as a flag where strips and backplane are

positioned thus their potential must not be altered during the iterative calculations. The method `SetBoundaryConditions()` is responsible for setting the boundary conditions of the electrode potentials and `fix` arrays.

3. Afterwards, the grid will be restricted to the coarsest grid used (bottom level in Figure 7). Because detector thickness and width are arbitrary, the number of grids (`multig`) will be adjusted to `XMAX` and `YMAX`. The grid translation is done by the class member function `Restriktor()` which will map the potential array to a coarser grid with (`XMAX`/2+1)×(`YMAX`/2+1) entries. `Restriktor()` is applied several times according to the number of grids used. The total number of grids will be determined depending on detector dimensions.

4. On the coarsest grid, the potentials will be calculated to a certain degree of accuracy. This is done by the class method `Gaussseidl(void*)` which calculates the potentials iteratively as described above in an infinite loop. As a breaking condition, the sum $\sum |\mathrm{pot}^{\mathrm{old}}_{\mathrm{j,i}} - \mathrm{pot}^{\mathrm{new}}_{\mathrm{j,i}}|$ is calculated for $0 < \mathrm{i} <$ `XMAX` and $0 < \mathrm{j} <$ `YMAX` normalized to the dimensions of the detector. As this computation is time-consuming, it is only checked for every hundreth interation whether this value falls below a certain value `err`=0.001.

5. When the calculation on the coarsest grid is finished, the two dimensional potential array will be mapped by `Prolongation()` to the next finer grid with $\mathtt{XMAX}_{\mathrm{new}} = 2\mathtt{XMAX}_{\mathrm{old}} - 1$ and $\mathtt{YMAX}_{\mathrm{new}} = 2\mathtt{YMAX}_{\mathrm{old}} - 1$. Mutual points of these two grids are being copied from the coarser to the finer grid. Additional points on the finer grid are calculated by interpolation, as shown in Figure 8. `Gaussseidl(void*)` is called and the potentials are calculated again with the initial values from the interpolation. This is done iteratively for every grid until the finest one is reached.

## 3.2 Fields

Weighting field $\mathbf{E}_w = -\nabla\phi_w$ and drift field $\mathbf{E}_d = -\nabla\phi_d$ are calculated numerically by:

$$\mathrm{E}_d, x = -\frac{\phi^d_{\mathrm{i,j}} - \phi^d_{\mathrm{i\text{-}1,j}}}{d}$$
$$\mathrm{E}_d, y = -\frac{\phi^d_{\mathrm{i,j}} - \phi^d_{\mathrm{i,j\text{-}1}}}{d}$$

for every point on the grid.

This is done by the method `CalculateFields(Potentials &p, Field** df, Field** wf)`.

If one wants to consider the Lorentz drift i.e. a $\mathbf{B}$ field present in the detector, the $\mathbf{E}_d$ field needs to be rotated by the Lorentz angle $\theta$ (which is differnt for $\mathrm{e}^-$ and h). This is done by a rotation matrix

$$R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{16}$$

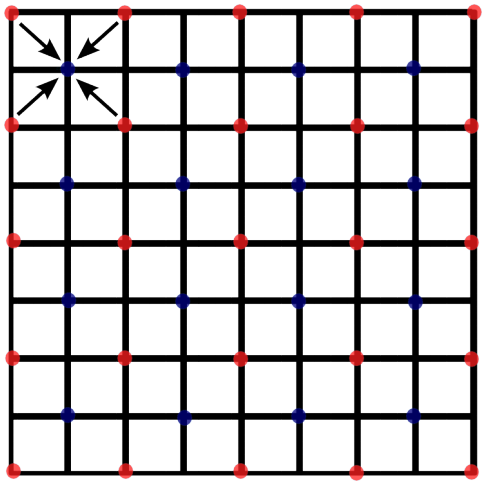


Figure 8: Interpolation during grid mapping from a coarser to a finer grid. Red dots symbolize points which are already part of the finer grid. First, points with four neighbours (blue) are calculated by averaging over their neighbours. Afterwards, the remaining points (black) are calculated in the same way. For boundary points on the left and right of the grid, we use periodic boundary conditions. For points of top/bottom rows, the potential above/below is replaced by cell itself.

and afterwards a projection on the unity vector of the drift direction, because we are only interested in the electric field in the drift direction. Note that whether the Lorentz angle $\theta$ is positive or negative (depending on the sign of the $\mathbf{B}$ field), the rotation is carried out clockwise or anti-clockwise.

## 3.3 Induced Currents

After the calculation of drift, weighting potential and fields, the currents induced in the readout electrode can be calculated. First, the electrons and holes are equally distributed along the path of the incident particle, as shown in Figure 9. A MIP creates $2\times22500$ electrons and holes for a detector with thickness 300 µm. Since $22500/300 = 75$, for an arbitrary thickness `YMAX`, the number of MIPs created is `YMAX`$\times75$.

This is done by `CreateCharges(Potentials&, Carriers*, int)`. Furthermore, a variable `inside` is set to 1 indicating that the charge is inside the detector.

If a $\mathbf{B}$ field is activated, the function `SetLorentz(Carriers*,double)` will set the member variables `etheta` and `htheta` (the Lorentz angles for electrons and holes) according to equation 9.

In the function `CalculateCurrents(Potentials &p,Field **df,Field **wf,Carriers* c,void * wfgui)` the calculation of induced currents is done. Two temporary fields are created and the drift field is rotated by $\theta_e$ and $\theta_h$, respectively.

If one wants to take account of diffusion, a smaller time unit $\Delta t$ and therefore more time steps are needed. Since the computation time is proportional to the number of

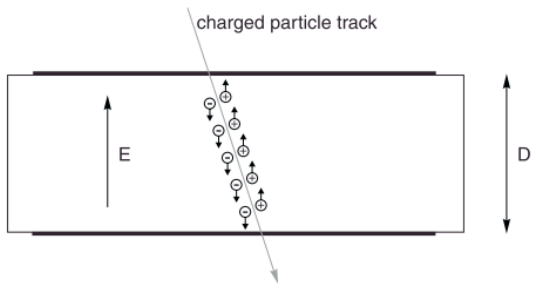



Figure 9: Creation of electrons and holes along the path of an incident particle [3].

steps, `TMAX` will only be increased if $T > 0$ (as there is no diffusion at absolute zero).

The calculation for induced currents is done as follows. For all time steps $0 < t <$`TMAX` and every charge carrier inside the detector (`inside`=1), the following calculation is carried out:

1. x and y componenty of the drift field are determined and the absolute value of the drift field $\sqrt{E_{d,x}^2 + E_{d,y}^2}$ ist calculated.

2. The electron mobility according to equation 4 and the x and y components of the drift velocity $\mathbf{v} = \mu\mathbf{E}$ are calculated.

3. If $T > 0$, the diffusion velocity will be determined according to equation 12.

4. The total velocity, i.e. the sum of drift and diffusion velocities, is calculated.

5. the current $I$ at time $i$ from the charge $q$ is then calculated by:

$$I_{i,q} = q[E_{w,x}(x_q, y_q)v_{q,x} + E_{w,y}(x_q, y_q)v_{q,y}] \tag{17}$$

   with the coordinates $x_q$ and $y_q$, the velocity componentes $v_{q,x}$ and $v_{q,y}$ of charge $q$ and the weighting field components $E_{w,x}$ and $E_{w,y}$. This is done for electrons and holes seperately ($q = \{e, h\}$) because of their different drift velocities.

6. Afterwards, the position of the carrier is updated to: $\mathbf{d}_{\text{new}} = \mathbf{d}_{\text{old}} + \mathbf{v}\Delta t$ and it is checked whether the carrier has left the detector or not.

7. After doing so for every carrier, the electron and hole currents at time $i$ are calculated: $\sum_q I_{i,q}$ with $q = \{e, h\}$. Furthermore, the total current (`itot`) and the collected charges (`qe`, `qh`, `qtot`), for the present time step are determined:

$$I_{i,tot} = I_{i,e} + I_{i,h}$$
$$Q_{i,tot} += I_{i,tot}\Delta t$$

## 3.4 The GUI

## 3.5 How to use it?

The GUI for the *weightfield* program is written based on the ROOT GUI classes. Figure 10 shows screenshots of the GUI.

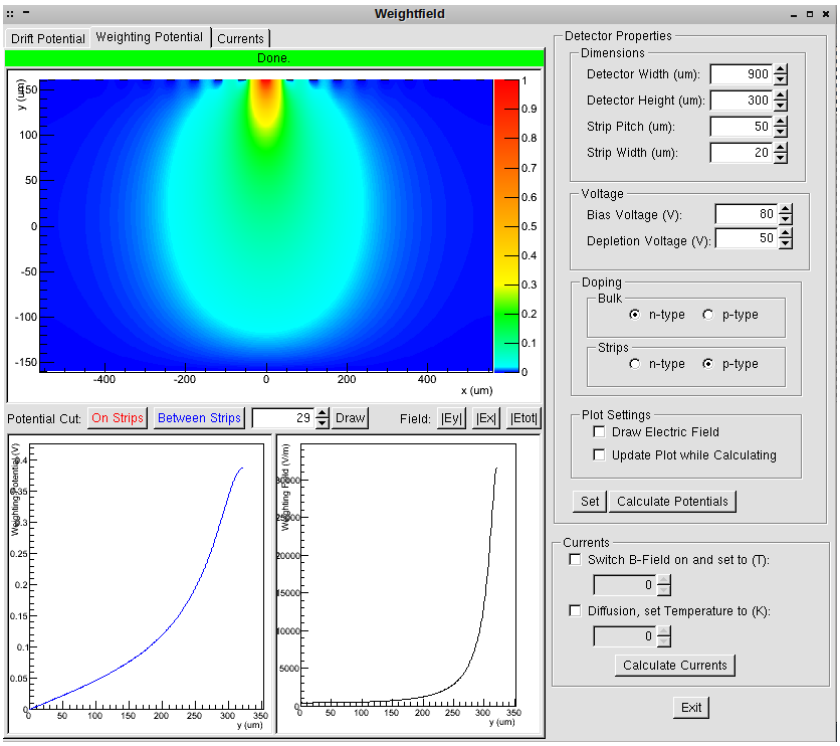



Figure 10: GUI.

There are three tabs which show the calculated results: one for the drift potential, one for the weighting potential and one for the currents.

The frame on the right side contains the controls and entry boxes for user input. Before starting the calculation, one needs to define the detector properties in the frame shown in Figure 11.

The user may enter the following properties:

1. **Detector dimensions:** Width and thickness of the detector and pitch and strip width [1]. By hitting the button *Set*, the cross section of the detector is drawn. The strips are on the top edge with the readout strip being always centered, the backplane is at the bottom. Here, the potential will be 0 inside the detector and *vbias* at strips/backplane depending on the doping.

[1] Be careful not to choose width>pitch. Recall, that strip width can not be larger than strip pitch, see Figure 1.

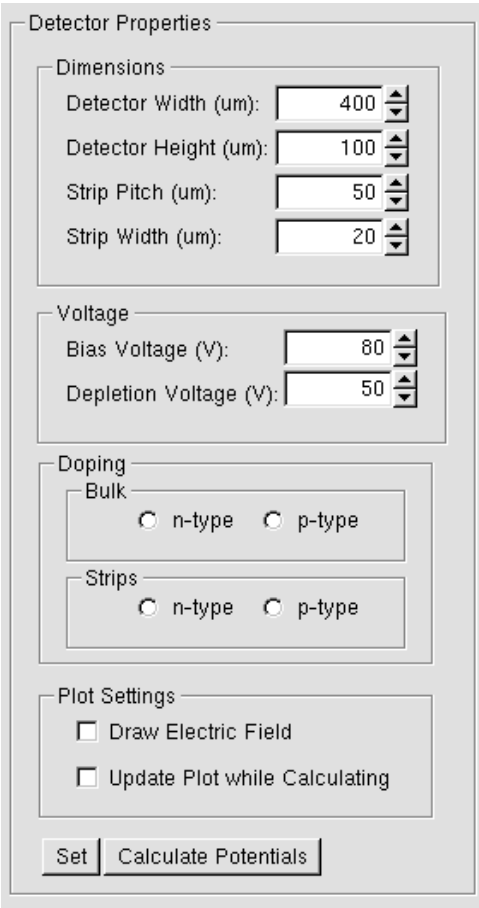



Figure 11: Control frame for detector properties.

2. **Bias and depletion voltage:** Set bias and depletion voltage. Default values are 80 V and 50 V, respectively.

3. **Doping:** One can choose for each, strips and bulk, between n- and p-type doping. Any combination is allowed.

4. **Plot Settings:** By setting a tick in the boxes, the user can select whether the **E**-field should be drawn and whether the potential plots should be updated during calculation.

By hitting the button *Calculate Potentials*, drift and weighting fields are being calculated. If *Update Plot while Calculating* is activated, the user can watch the development of the potentials during the calculation but this will take extra time for repeated plotting. This may take from a few seconds to a few minutes, depending of course on the detector dimensions.

The bar on top of the potential plots (see Figure 12) provides information on the progress of the calculation. When the calculation is done, the progress bar becomes green. Weighting and drift potentials are drawn on the large canvas. Note that the

colour scale of the drift potential is linear, but the scale of the weighting potential is not! Potential cuts and fields are drawn on the two canvases at the bottom. Here, one can choose between on and between strips or enter an arbitrary number between [-XMAX/2,XMAX/2] in the entry box.

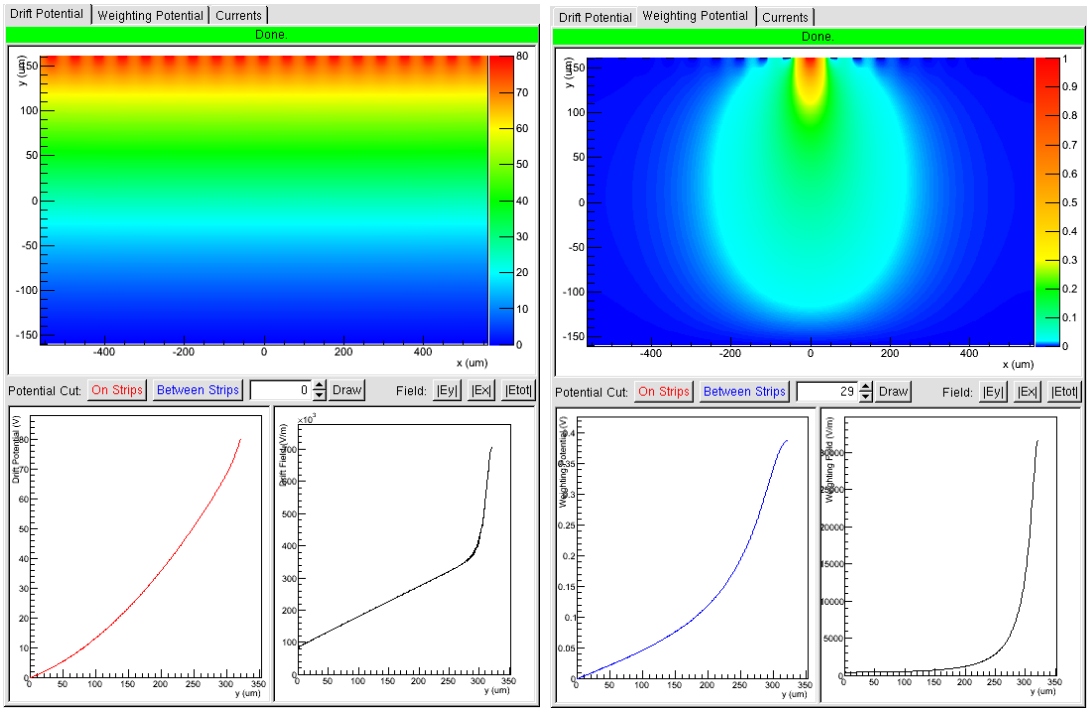



Figure 12: Potential Tabs.

The button *Calculate Currents* will only be enabled when the potential calculation is finished. The user can choose whether a magnetic field **B** should be present and whether diffusion should be considered, see Figure 13. Then one may enter a value for B (T) and for the temperature T (K). Switching to the *Currents* tab, one may also choose where the particle hits the detector.

The results of the current calculation are then shown in the currents tab (see Figure 14) which provides the following:

- Plots of the currents: There will be three curves plotted, one for electrons (blue), one for holes (red), and the total current (green).
- The progress bar below the plot.
- At which point the particle hits the detector may be chosen.
- Collected charges are displayed in the frame *Charge Collection*.
- The Lorentz angle in degrees for electrons and holes is shown in the *Lorentz Drift* frame.

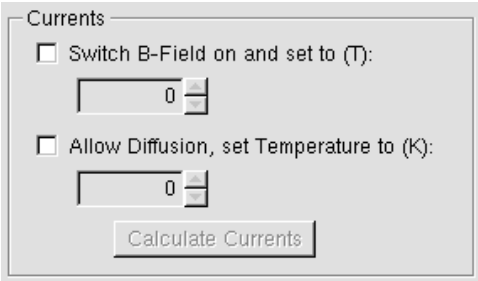



Figure 13: Currents frame.

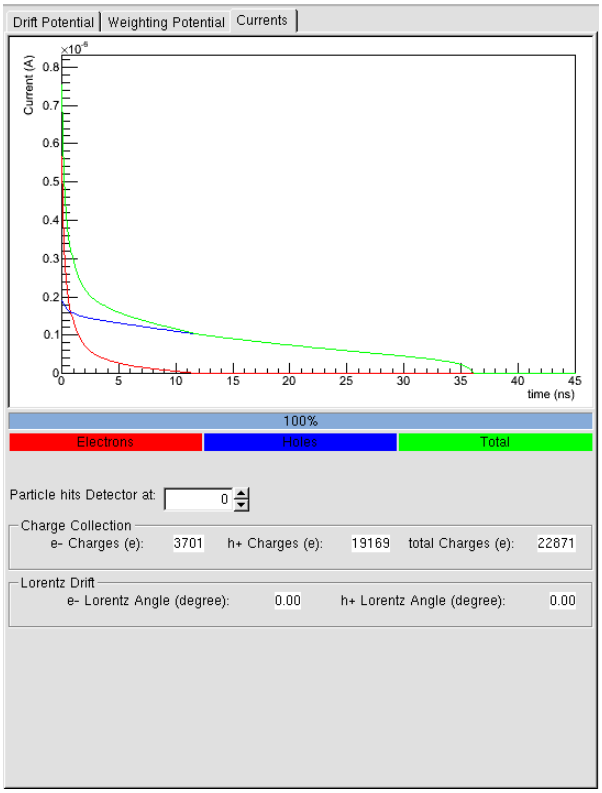



Figure 14: Currents Tab.

## 3.6 How does it work?

As already mentioned, the GUI of *weightfield* is based on the ROOT classes. The class WFGUI is implemented in the files `WFGUI.h` (class declaration) and `WFGUI.cxx` (class method implementation).

WFGUI is derived from the class `TGMainFrame` and therefore inherits from this class.

Because we want the program *weightfield* to be a standalone program (i.e. we want a `main()` to be the starting point of program execution), we create an object of `TApplication` in the `main()` and by calling `theApp->Run();` the event loop will be started (for more information, see Chapter 25 of [11]). Furthermore, a dictionary needs to be created in oder to have the signal and slot mechanism working.

All buttons are connected via this mechanism. The method `Connect` which is a member of the class `TQObject` is used like this:

```
Bool_t Connect(const char* signal, const char* receiver_class, void*
    receiver, const char* slot)
```

For example, the button *Calculate Potentials* is connected like this:

```
CalcPotButton->Connect("Clicked()","WFGUI",this,"ThreadstartPotential()")
```

In oder to still be able to use the GUI while calculation e.g. switch between drift and weighting potential tab, the potential and current calculations are threaded by using ROOTs TThread class. Because ROOT is not yet thread save, the *Stop* button to terminate the calculation before its finished has been removed by default in oder to avoid occasional crashes[2]. For more information on threading, see Chapter 23 in [11].

The two most important mechanisms of the class WFGUI are the potential and the current calculations, all the others are get and set methods or methods to plot histograms, graphs etc. In the following, these two are going to be discussed.

By hitting the button *Calculate Potentials*, the method `ThreadstartPotential()` is called und starts the potential calculation via `void WFGUI::CallCalculatePotentials()` which does the following:

1. All canvases and histograms are being reseted. We want to start fresh and do not want to base the calculation on possiblely wrong remaining values from prior calculations.

2. The progress label above the histogram plots of the potentials is being updated.

3. Some buttons are disabled to prevent a change of parameters (like doping) during the calculation.

4. The user-selected doping is set via the method `Potentials::SetDoping(stripdoping,bulkdoping)`.

[2]Though, if someone wants a stop button and can live with the occasional crashes, the stop button is still implemented but commented in the code.

5. User input of detector width etc. are set by calling `SetPitchWidthXY(YMAXentry->GetNumber(),XMAXentry->GetNumber(),Pitchentry->GetNumber(),Widthentry->GetNumber())`. `GetNumber()` gets the user input of the particular entry field. The same is done for bias and depletion voltages.

6. `Potentials::SetBoundaryConditions()` sets the boundary conditions for weighting and drift potentials.

7. With `dwpot.Multigrid(this)` the actual calculation is started. The object WFGUI (`this`) is passed to this function. This is necessary, because, if *Update Plot while Calculating* is selected, the histograms have to be drawn during the calculation (every hundredth, sevenhundreth or thousandth iteration, depending on the current grid). Everytime a class member of WFGUI (in this case the histograms for weighting and drift potential) is used in another class, we need to pass the GUI itself.

8. Afterwards, the progress label is again updated and all buttons are enabled.

The calculation of currents is also threaded, the button *Calculate Currents* is connected in a similar way:

```
CalculateButton->Connect("Clicked()","WFGUI",this,"ThreadstartCurrents()")
```

The method `ThreadstartCurrents()` calls the method `CallCalculateCurrents()`, where the current calculation is started:

1. The progress bar is reset.

2. B (magnetic field) und T (temperature) are set according to the user's input.

3. The x value where the particle is supposed to hit the detector is set.

4. The charge carriers are created by `CreateCharges(Potentials &p,Carriers *c,int hit)`, Lorentz angles are calculated via `SetLorentz(Carriers *c,double b)` and finally the calculation is done by `CalculateCurrents(Potentials &p,Field** df, Field** dw, Carriers* c,void* wfgui)` which also draws the graphs after calculation and updates the labels in the currents information frame.

# 4 Results

It is important to note that the shape of the current signal from a strip detector will differ from the parallel-plate capacitor configuration. Figure 15 shows the current signals from a double-sided strip detector with p-type bulk and n-type strips and vice versa. One sees that the signals seen on opposite strips of a double-sided detector have different shapes [2]. Though in case of `weightfield`, we do not consider a double-sided strip detector (even though both sides can be simulated separately as demonstrated in Fig. 16), the fact that

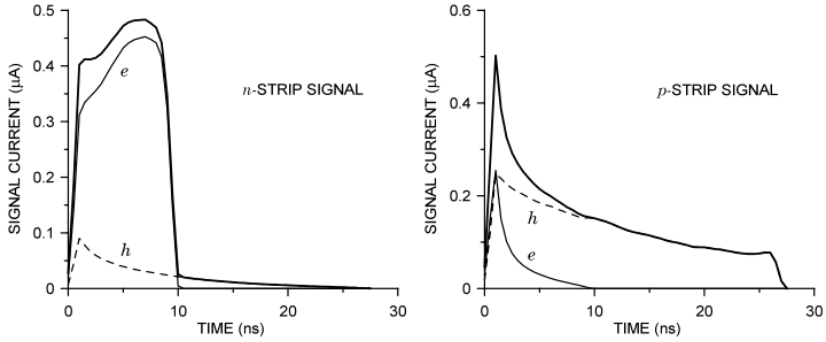



Figure 15: Signal shapes from a double sided strip detector [2].

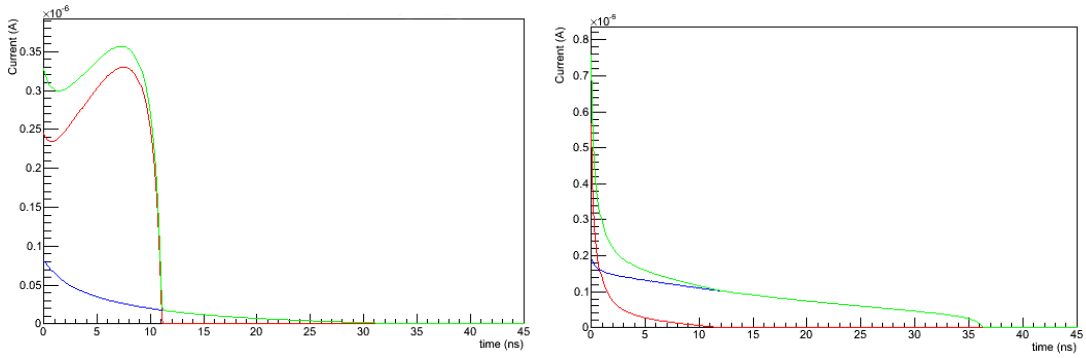



Figure 16: Signals from an n-type bulk, p-type strip (right) and an n-type bulk, n-type strip (left) detector (all the other properties are the same).These two simulations represent the two signal shapes obtained on opposite sides of a double-sided strip detector and thus match the graphs of Fig. 15.

electrons and holes do not induce currents with the same waveform remains. Figure 16 shows the signals calculated with `weightfield` and one easily sees the difference in shape.

Figure 17 shows a sketch of the weighting potential in a 2-dimensional strip detector and the induced currents of two charges. The carrier on the right path (1) experiences a field in one direction resulting in a current as seen below. This case corresponds to the case where the MIP hits the detector at $x = 0$ and the electron-hole pairs are not deflected by a magnetic field. One the other hand, a charge moving along path (2) will move through an electric field with direction changes and therefore a bipolar waveform of the induced current, corresponding to an integral charge of zero.

This is nicely seen in Figure 18 where calculating results from `weightfield` of induced currents are displayed.

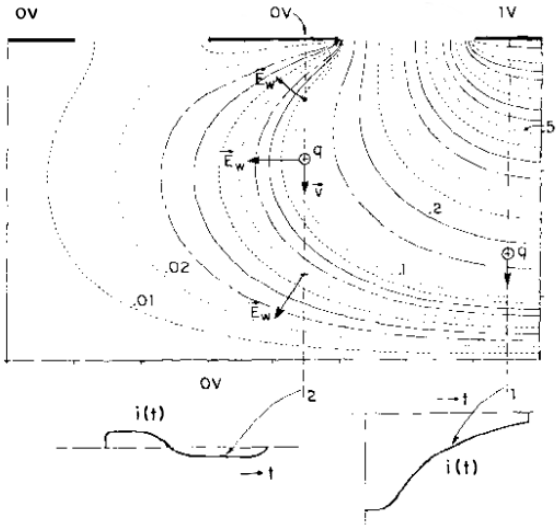



Figure 17: Weighting potential, field and carrier movement [12].

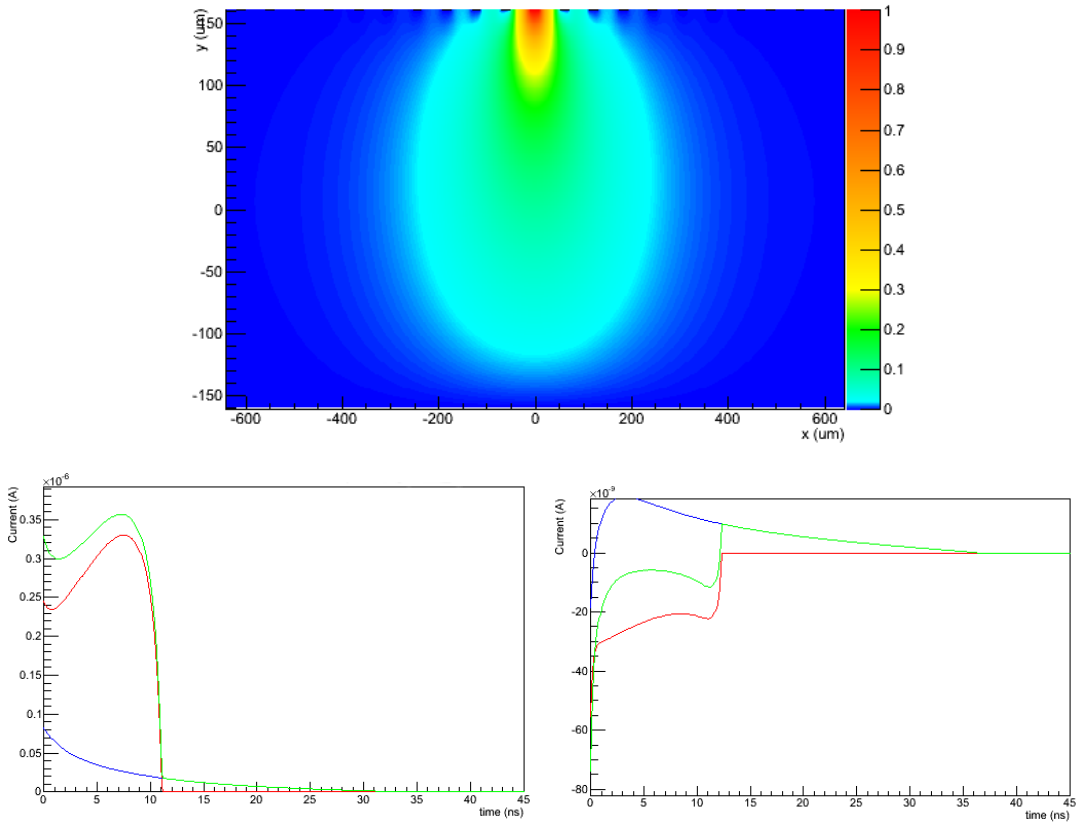



Figure 18: Top: Weighting potential of a detector with: hieght=1000 µm, thickness=300 µm, strip pitch=50 µm and strip width=20 µm. Bottom: induced currents by charge carriers (red = electrons, blue = holes, green = total) induced on the readout strip (left) and on the neighbour strip (right).

The influence of a magnetic field on induced currents is shown in Figure 19. For the calculation a **B**-Field of 4 T [13] was chosen as this is present within the CMS Tracker.

Again, one sees a bipolar current waveform due to the non-parellel carrier movement with respect to the weighting field. In the case of a strip detector with bulk: p-type and strips: n-type, the electrons drift toward the readout strip and the holes toward the backplane, so the waveform of the current of the electrons will be more affected by the magnetic field, as seen in the figure. This effect is furthermore amplified as the Lorentz angle of electrons is significantly larger than that of the holes.

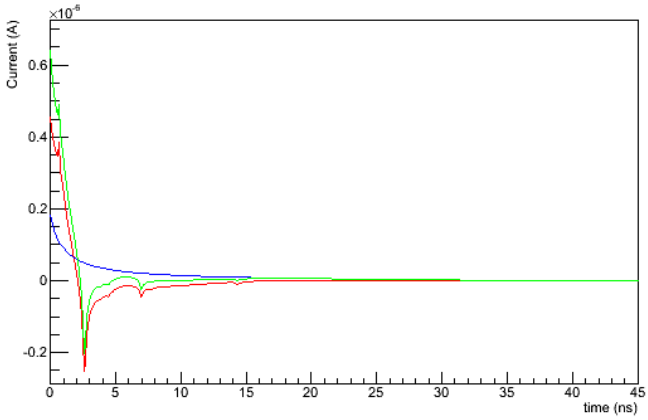



Figure 19: Induced currents with a magnetic field of 4T resulting in Lorentz angles of $\theta_e \approx 28°$ and $\theta_h \approx 11°$ (detector dimensions as in Figure 18).

Figure 20 shows a current signal calculated by `weightfield` with the consideration of thermal diffusion at a temperature of 4°C, the typical operating temperature of the CMS detector [13], which has a noisy and broadening effect on the current shape.

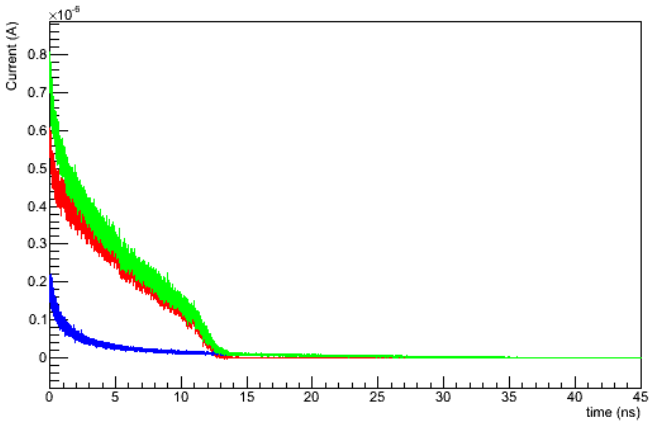



Figure 20: Signal of a strip detector at T ≈ 4°C.

# References


[1] High Energy Physics Department of Firenze. *Silicon_Detector.gif*. URL: http://hep.fi.infn.it/CMS/sensors/Silicon_Detector.gif.

[2] Helmuth Spieler. *Semiconductor detector systems*. Oxford: Oxford University Press, 2005.

[3] M. Friedl, T. Bauer, and M. Krammer. 'A simple model of charge collection in silicon detectors'. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 461.1-3 (Apr. 2001), pp. 192–196. DOI: 10.1016/S0168-9002(00)01204-3.

[4] T. Bauer, M. Friedl, and M. Krammer. 'A Simple Model of Charge Collection in Silicon Detectors'. 2000. URL: http://friedl.hephy.at/sim/poster.pdf.

[5] S. Ramo. 'Currents Induced by Electron Motion'. In: *Proceedings of the IRE* 27.9 (Sept. 1939), pp. 584–585. DOI: 10.1109/JRPROC.1939.228757.

[6] V. Bartsch et al. 'Lorentz angle measurements in silicon detectors'. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 478.1-2 (Feb. 2002), pp. 330–332. DOI: 10.1016/S0168-9002(01)01820-4.

[7] Markus Friedl. 'The CMS Silicon Strip Tracker'. PhD thesis. 2001. URL: http://friedl.hephy.at/diss.

[8] Manhee Jeong and Mark D. Hammig. 'The Atomistic Simulation of Thermal Diffusion and Coulomb Drift in Semiconductor Detectors'. In: *IEEE Transactions on Nuclear Science* 56.3 (June 2009), pp. 1364–1371. DOI: 10.1109/TNS.2009.2021426.

[9] Martin Neumann. 'Skript: Computational Physics I'. URL: http://www.exp.univie.ac.at/cp1/cp1-6/cp1-6.html.

[10] Ulrich Ruede. *multigrid method*. URL: http://www.mgnet.org/mgnet/tutorials/xwb/mg/_12122_tex2html_wrap26.gif.

[11] Cern. *User's Guide: ROOT*. 2007. URL: http://root.cern.ch/drupal/content/users-guide.

[12] V Radeka. 'Low-Noise Techniques In Detectors'. In: *Annual Review of Nuclear and Particle Science* 38.1 (Jan. 1988), pp. 217–277. DOI: 10.1146/annurev.nucl.38.1.217.

[13] V. Khachatryan et al. 'CMS tracking performance results from early LHC operation'. In: 70 (2010), pp. 1165–1192. DOI: 10.1140/epjc/s10052-010-1491-3. URL: http://www.springerlink.com/index/10.1140/epjc/s10052-010-1491-3.