

BACHELOR THESIS

A SPEED OPTIMIZED  
IMPLEMENTATION OF THE RESPONSE  
SIMULATION FOR MONOLITHIC  
ACTIVE PIXEL SENSORS

JONATHAN ENDERS

INSTITUT FÜR KERNPHYSIK  
GOETHE-UNIVERSITÄT

First examiner: Prof. Dr. Joachim Stroth  
Second examiner: Dr. Michael Deveaux

September 21, 2011

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Prerequisites</b>	<b>4</b>
2.1	CBM . . . . .	4
2.2	MVD . . . . .	6
2.3	CMOSMAPS . . . . .	6
2.4	The software . . . . .	9
2.5	Simulation . . . . .	9
2.6	Digitizer . . . . .	10
<b>3</b>	<b>The model</b>	<b>11</b>
3.1	Segmentation model . . . . .	11
3.1.1	The track length . . . . .	13
3.2	Integration model . . . . .	13
3.2.1	Solving the integral . . . . .	14
3.2.2	Inspecting the solutions . . . . .	17
3.2.3	Applying to model . . . . .	18
<b>4</b>	<b>The code</b>	<b>19</b>
4.1	Relating the variables . . . . .	20
4.1.1	The prefactor . . . . .	21
4.1.2	Fitting the line equation . . . . .	21
4.1.3	Integration variable . . . . .	23
4.2	Deriving the final expression . . . . .	24
4.3	The new code . . . . .	24
4.4	Call tree comparison . . . . .	27
4.5	Physical meaning of the variables . . . . .	28
4.6	Options for further optimization . . . . .	28
<b>5</b>	<b>Validation and benchmarking</b>	<b>29</b>
5.1	Validation . . . . .	29
5.1.1	Simulation setup . . . . .	29
5.1.2	Further implementations . . . . .	31
5.1.3	Test results . . . . .	32
5.1.4	Discussion of the results . . . . .	37
5.1.5	Precision . . . . .	38
5.2	Speed . . . . .	41
5.3	Conclusion . . . . .	42

# Deutsche Zusammenfassung

Das Compressed Baryonic Matter (CBM) Experiment soll in den nächsten fünf bis zehn Jahren an der zukünftigen Facility for Antiproton and Ion Research (FAIR) gestartet werden. Das Experiment soll der Erforschung exotischer Zustände von Materie bei extrem hohen Dichten dienen. Diese Zustände können in stellaren Objekten wie Neutronensternen gefunden werden.

Die Hard- und Software für das Experiment befindet sich momentan in der Entwicklung. Ein komplexes System aus zahlreichen Detektoren wird entworfen und der Beschleuniger des GSI Helmholtz Centre for Heavy Ion Research GmbH wird erweitert, um den Teilchenstrahl bereitzustellen, den sich das CBM Experiment mit den diversen anderen Experimenten teilt.

Für den Micro Vertex Detector (MVD) wird Software für die Analyse der Detektormessdaten, als auch für die Detektorsimulation implementiert.

In dieser Bachelorarbeit wird die Implementierung eines der existierenden Detektorsimulationsmodelle<sup>1</sup> des MVD geändert. Dafür wird eine Summation durch das Ergebnis eines analytisch ausgewerteten Integrals ersetzt. Dies zielt darauf ab, die Ausführungsgeschwindigkeit des Modells zu verbessern. Das ist wünschenswert, da diverse zeitaufwändige Simulationen zur Detektordatenauswertung die Detektorsimulation nutzen.

Schliesslich wurde mit den Änderungen an der Implementierung eine allgemeine Verschnellerung um einen Faktor drei erreicht.

---

<sup>1</sup>Das existierende Modell wird in [5] entwickelt

# Chapter 1

## Introduction

The Compressed Baryonic Matter (CBM) Experiment is set to be conducted at the future Facility for Antiproton and Ion Research (FAIR) in the coming five to ten years. The intent of the experiment is the study of exotic forms of matter formed in extremely high density conditions found in stellar objects like neutron stars.

The design of the hardware and software for this experiment is currently taking place. A complex system of different detectors is being developed and the accelerator of the GSI Helmholtz Centre for Heavy Ion Research GmbH is extended to deliver the beam which the CBM experiment will share with various other experiments.

For the Micro Vertex Detector (MVD), software for analyzing the detector output as well as a simulation of the detector response is being implemented.

In this Bachelor thesis the implementation of one of the existing detector response models<sup>1</sup> for the MVD is altered. For that a summation is replaced by the result of an analytically evaluated integral. This aims to improve the execution speed of the model which is desirable as various time consuming simulations concerning the analysis of the detector output depend on the detector response simulation.

In the end the alterations yield an overall speed-up of the detector response model by a factor of about three.

---

<sup>1</sup>The existing model is developed in [5]

## Chapter 2

# Prerequisites

### 2.1 CBM

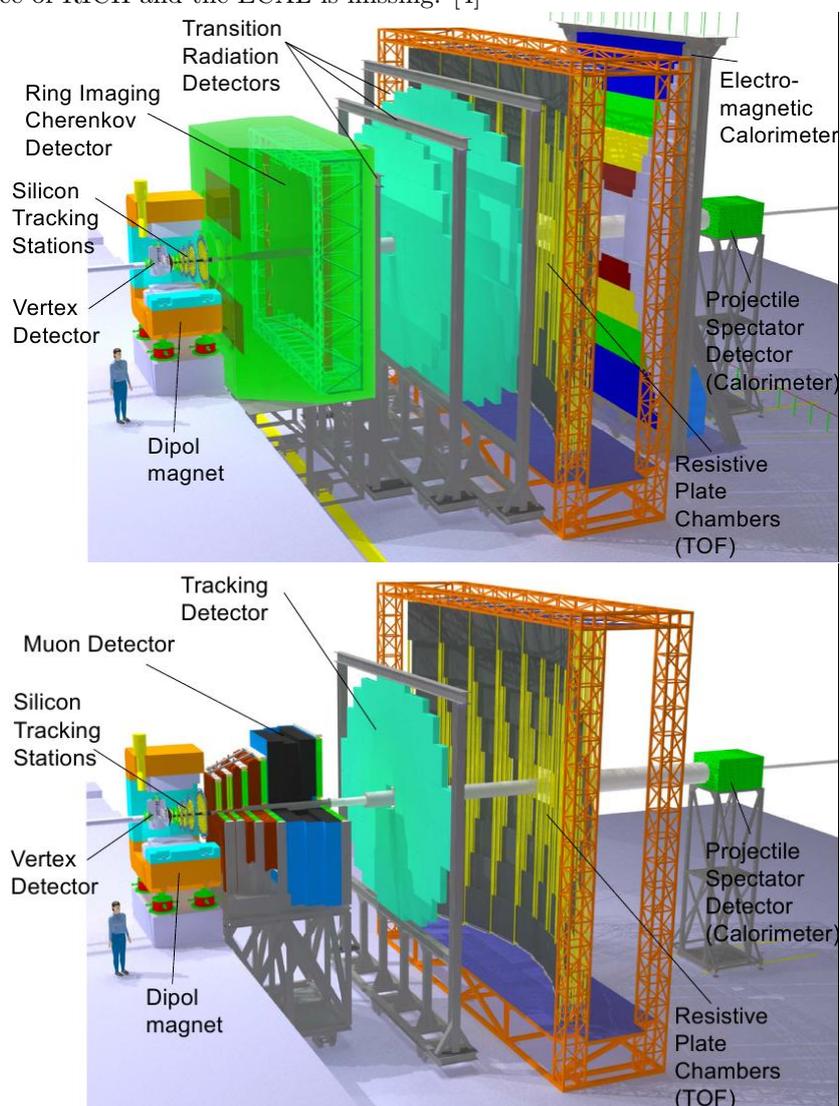
Being a fixed target experiment the CBM station will consist primarily of the target and an entire array of different detectors. As planned each detector will serve a different purpose, will have a different distance from the target and will use a different technology. All detectors together will form a cone-like shape with the target at the apex of the cone. Starting from the target and moving outwards the following detectors will be encountered[6, p. 897]:

- The Micro Vertex Detector (MVD) ought to be able to distinguish between vertices created at the location of the target and vertices created later by decaying reaction products. It is planned to contain 2 to 3 detector stations 5 to 20 cm off the target.
- The Silicon Tracking System (STS) aims to find the track and momentum coordinates of charged particles. It is planned to contain about 8 microstrip detector stations placed 30 to 100 cm from the target.
- The Ring Imaging Cherenkov detector (RICH) should detect electrons while suppressing pions. The radiator material which will be placed behind the STS is contained in a 2.9 m long gas vessel. The Cherenkov radiation that is produced by the electrons passing through the gas vessel is detected using photo multipliers.
- The Muon Chamber (MuCh) system shall be used for low-momentum muon identification. For this a cascade of six hadron absorbers with six tracking chambers ordered intermittently will be placed behind the STS taking the space of the RICH detector.
- The Transition Radiation Detector (TRD) shall track ultrarelativistic electrons and positrons. It's planned to consist of three to four detector layers with a distance of 5 to 10 meters past the target. The technology will be wire chamber (MWPC) or gas electron multiplier (GEM).
- The timing Resistive Plate Chambers (RPC) ought to measure the time of flight (TOF) which is used for the identification of hadrons 10 meters down the target.

- The Electromagnetic Calorimeter (ECAL) will be used to detect photons and mesons. It is planned to consist of alternating layers of lead and scintillator material. The distance from the target will be variable.
- The Projectile Spectator Detector (PSD) will be the farthest from the target. It shall measure the number of non-interacting nucleons from a projectile allowing for the reconstruction of the reaction plane. Similar to the ECAL it will be a lead-scintillator calorimeter.

CBM cannot be equipped to run with all the detectors at the same time. As seen in Figure 2.1 there are two setups each using a different set of detectors.

Figure 2.1: Two different CBM detector setups with the target on the left side. In the upper setup the RICH is being used. In the lower setup MuCh takes the place of RICH and the ECAL is missing. [4]



## 2.2 MVD

The Micro Vertex Detector is the first detector that the particles leaving the target encounter. Of all detectors it is the closest to the target. It is made up of presumably three adjacent detector planes placed 5 cm apart each being able to detect the spot on the plane at which the particle crossed it. For that purpose the planes are split up in pixels. The specific aim of the MVD is to reconstruct the spacial coordinates where a particle decay (a vertex) took place. This is done via an extrapolation of the tracks of the decay products. A backdraw of this method is that small uncertainties in the track coordinates result in big uncertainties in the vertex coordinates. That's why exceptionally high spatial resolution is expected from the MVD. For the measurement the MVD contains the following core components:

- sensors: get hit by particles and output digital data stream
- data acquisition (DAQ): uses hardware to preprocess the data and to store it
- data analysis: mainly software designed to look for particles/phenomena

## 2.3 CMOSMAPS

Due to the very specific and high demands that the MVD detector planes have to fulfill there aren't a lot of technologies available that fit those needs. The technology that does seem to make the best compromise among spatial resolution, radiation hardness, detection efficiency and price are Complementary Metal Oxide Semiconductor Monolithic Active Pixel Sensors (CMOSMAPS short MAPS). A photo of one such sensor can be seen in Figure 2.2

Figure 2.2: Photo of a MAPS of type MIMOSA26[1]

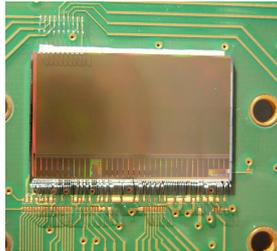
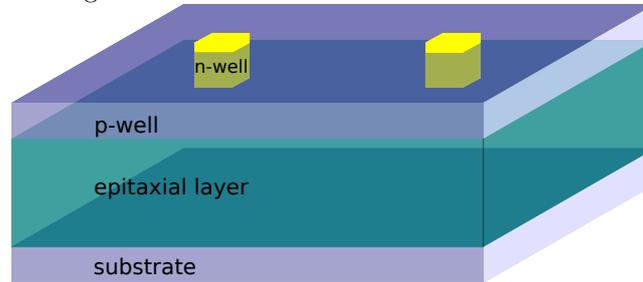


Figure 2.3: Schematic cross section of a MAPS

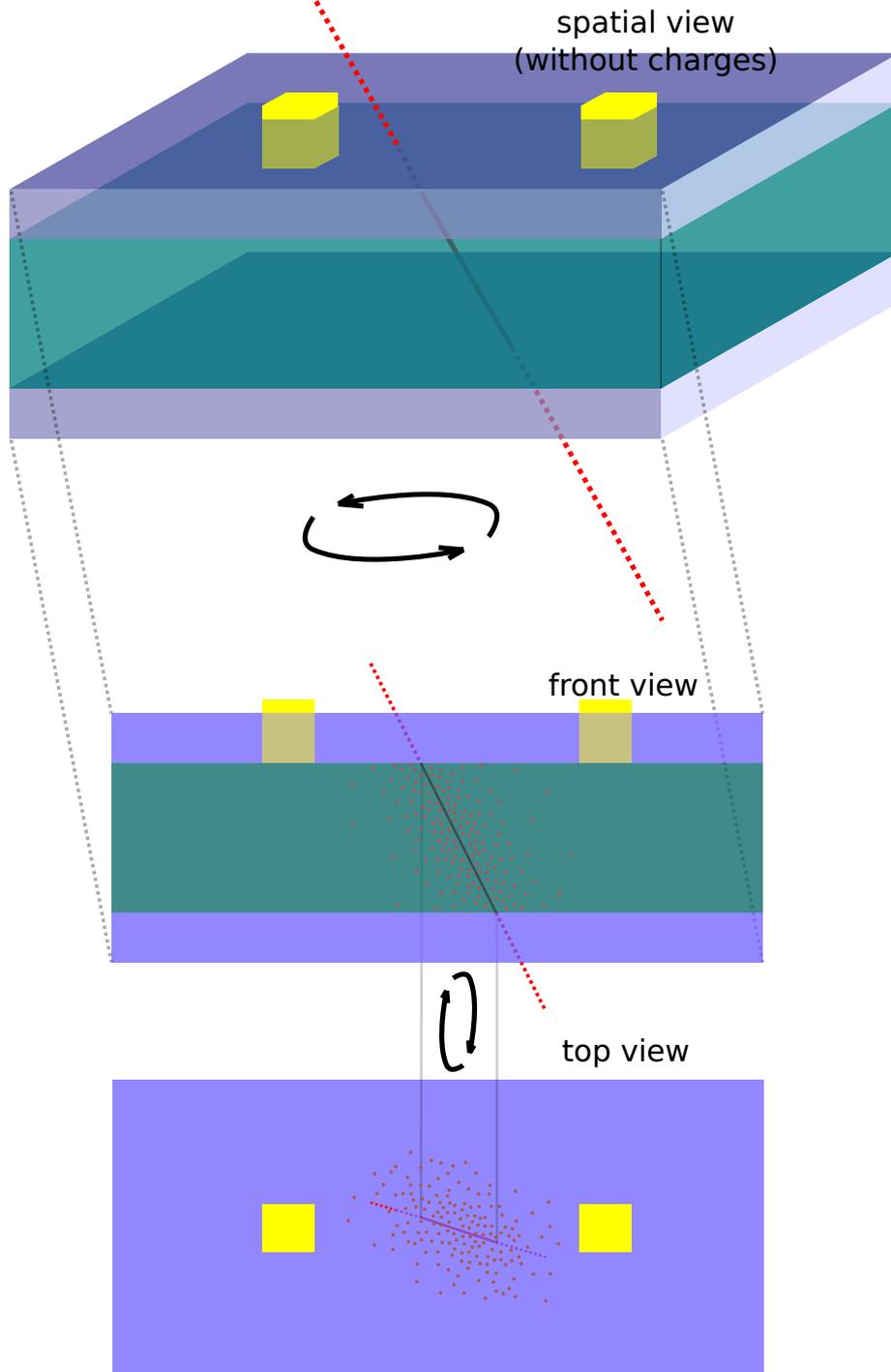


MAPS are silicon pixel detectors. As such they consist of several differently doped silicon layers shown schematically in Figure 2.3. The top (p-well) and the bottom (substrate) layer are highly p-doped. The so-called epitaxial layer in the middle is weakly p-doped. Between the differently p-doped layers a small voltage builds due to the doping gradient. When a particle passes through the epitaxial layer it generates electron/hole pairs. These diffuse through the epitaxial layer. Only the electrons are reflected on the potential gradient they encounter on the intersection to the substrate and p-well. Thus the electrons can practically not leave the epitaxial layer.

The p-well layer is penetrated by n-doped n-wells periodically. Here the pairs are collected to cause an electric signal which is amplified, discriminated and encoded on the chip resulting in a digital data stream. One n-well in conjunction with the sensitive volume around it forms one pixel. [2]

In Figure 2.4 a MAPS being penetrated by a particle is sketched.

Figure 2.4: Particle crossing MAPS



## 2.4 The software

With the hardware that makes up the MVD comes a need for software. Specifically there are two main functions that need to be handled by this software. Those are:

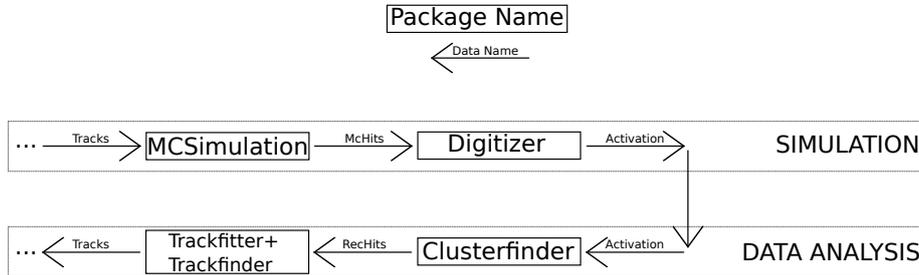
- the track extraction based on the detector response
- the detector response simulation based on the incoming track

Note that the second is essentially the reverse of the first. The first function is the part that is actually needed for the detector operation. For all the particles the tracks need to be reconstructed using the detector response. The second function is merely a tool designed to test the first without the need for actual detector generated data. It simulates the detector.

## 2.5 Simulation

Experiments that use the MVD can also be simulated in software. For that purpose several different code packages<sup>1</sup> have to work together. As every package relies only on the output of the former package one can visualize the packages as a chain that the data passes through[3, p. 178]. This is shown in Figure 2.5.

Figure 2.5: Excerpt of the simulation and data analysis chain



- Tracks: locations of particle tracks
- McHits, RecHits: coordinates where particles penetrated the detector
- Activation: list of pixels that are activated by a particle passing nearby

At the end of the simulation chain the data analysis chain that is used for the real detector data can be attached. In this way the data analysis chain can be tested. It ought to be able to reconstruct parameters that were used as input for the simulation.

<sup>1</sup>The code packages are a part of the scientific libraries CBMROOT and GEANT.

## 2.6 Digitizer

This thesis concerns itself with the digitizer package. Given the entry and exit coordinates (McHits) of particles of the epitaxial layer the digitizer has to predict what the real MVD response (Activation) would be. In order to do so models of how charge is generated and distributed among the pixels were developed and calibrated with actual experimental data. The digitizer outputs the result based on the models in form of a list of activated pixels. The digitizer is part of the CBMROOT library.

ROOT is a collection of scientifically related programming libraries most of which originated at CERN<sup>2</sup>. CBMROOT is an extension of ROOT that adds functionality specific to the CBM experiment. Aside from being a library ROOT can also be used like an application. As such it is accessible to the user by a C++ equivalent scripting language<sup>3</sup>.

For a common task like simulating a fixed target experiment the typical approach would be to write and run a high-level and thus relatively short script also called a macro. Those macros are able to call the libraries' functions which are precompiled thus offering good performance in terms of speed.

---

<sup>2</sup>Organisation Européenne pour la Recherche Nucléaire

<sup>3</sup>The scripts are interpreted using the C++ interpreter CINT

# Chapter 3

## The model

In all the models that predict the pixel activation for the MVD the assumption is made that the particle passing through the MVD follows a straight line. This assumption is justified because of the thin active volume ( $\mathcal{O}(10\mu m)$ ) of the MAPS. The thickness of this volume is small as compared to the radius of particle tracks reaching the MVD.

### 3.1 Segmentation model

Through experimental data it is known<sup>1</sup> that the charge that is collected by a whole pixel caused by a particle crossing the detector perpendicularly nearby can be well approximated by a reshaped Lorentz function<sup>2</sup>:

$$Q_{perp} = \frac{A}{\bar{a} * r^2 + 1}$$

$r$ : distance from the particle's crossing point to the center of the pixel of interest (in the top view)

$A$ : variable that scales the height of the Lorentz function

$\bar{a}$ : variable that scales the width of the Lorentz function

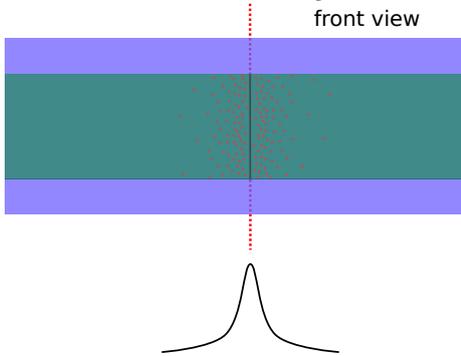
A more thorough description about how these parameters are defined can be found in section 4.1. In Figure 3.1 this distribution of the charge is visualized.

---

<sup>1</sup>The motivation for the description using a Lorentz distribution is given by [5, p. 108]

<sup>2</sup>The mathematical form of a Lorentz function is  $\mathcal{L}(x) = \frac{\frac{1}{2\pi} * \Gamma}{(\frac{1}{2}\Gamma)^2 + (x-x_0)^2}$

Figure 3.1: Charge distribution  
front view



To gain a good description for particles impinging at arbitrary angles the following approach is taken:

The ionizing track that runs through the epitaxial layer of the detector is split into multiple segments. For a nearby pixel the charge caused by every segment  $q_i$  is calculated individually using a Lorentz function each centered at the center of every segment. The sum of all the segment charge contributions is the total charge that is actually recorded by the pixel:

$$Q_{arb} = \sum_i q_i = \sum_i \frac{\bar{A}}{\bar{a} * r_i^2 + 1}$$

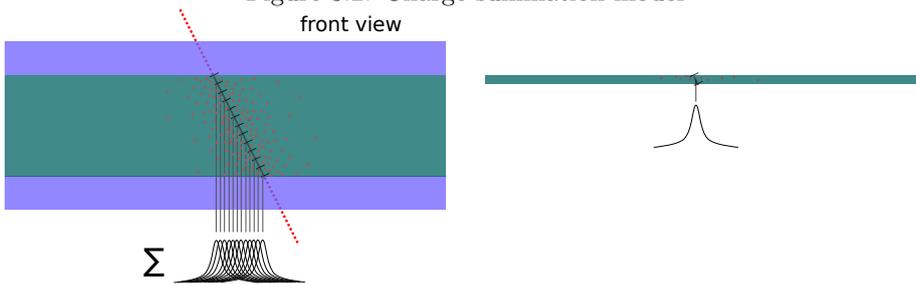
The sum goes over all segments each having an index.

$i$ : segment index

$r_i$ : distance from the center of the  $i$ -th segment to the center of the pixel of interest (top view)

The summation is visualized in Figure 3.2.

Figure 3.2: Charge summation model  
front view



When looking at one segment of infinitesimal length from above (top view) it becomes obvious that the charge generated by this segment which is of pointlike shape ought to be radial symmetric. The particle impinging perfectly perpendicular has the same pointlike shape. Thus the approach with the sum over the radial symmetric Lorentz functions  $q_i$  seems reasonable.  $Q_{arb}$  itself is not radial symmetric.

This model adequately describes the detector response for the arbitrary case. From now on it will be referred to as segmentation model (SM).

### 3.1.1 The track length

It is known that the generated charge is proportional to the length of the track in the epitaxial layer. For one segment which is just a small piece of the whole track, the generated charge is reduced accordingly:  $\bar{A} = \frac{l}{L}A$

$$\mathcal{Q}_{arb} = \sum_n \frac{l}{L} \frac{A}{\bar{a} * r_i^2 + 1} = \sum_n \frac{1}{N} \frac{A}{\bar{a} * r_i^2 + 1}$$

$l$ : length of one segment

$L$ : length of whole track inside the epitaxial layer

$N$ : number of segments

## 3.2 Integration model

The topic of this thesis is the derivation of an expression that describes  $\mathcal{Q}_{arb}$  directly without the need of summing over multiple radial symmetric functions. An expression like that can be derived by reducing the segments to infinitesimal length replacing the sum with an integral:

$$\mathcal{Q}_{arb} = \int_{l_{st}}^{l_{end}} \frac{dl'}{L} \frac{A}{\bar{a} * r(l')^2 + 1}$$

Recall that all the models assume a straight line for the particle track going through the sensitive layer of the detector. Taking cartesian coordinates  $x, y$  with  $r^2 = x^2 + y^2$  a straight line is represented by a linear relation between  $x$  and  $l$  (also  $y$  and  $l$ ):

$$\begin{aligned} x(l') &= m_x l' + b_x \\ y(l') &= m_y l' + b_y \end{aligned}$$

The integral becomes:

$$\begin{aligned} \mathcal{Q}_{arb} &= \int_{l_{st}}^{l_{end}} \frac{dl'}{L} \frac{A}{\bar{a} * [(m_x l' + b_x)^2 + (m_y l' + b_y)^2] + 1} \\ &= \int_{l_{st}}^{l_{end}} \frac{dl'}{L} \frac{A}{\bar{a} * [(m_x^2 + m_y^2)l'^2 + 2(m_x b_x + m_y b_y)l' + (b_x^2 + b_y^2)] + 1} \\ &= \frac{A}{L\bar{a}} \int_{l_{st}}^{l_{end}} \frac{dl'}{(m_x^2 + m_y^2)l'^2 + 2(m_x b_x + m_y b_y)l' + (b_x^2 + b_y^2 + 1/\bar{a})} \end{aligned}$$

This is an integral of a rational function. Solving it results in the desired closed form for  $\mathcal{Q}_{arb}$ .

### 3.2.1 Solving the integral

A closed form solution of the following expression is desired:

$$\int_{x_{st}}^{x_{end}} \frac{1}{ax^2 + bx + c} dx$$

The standard approach to solve such an integral is to first do a partial fraction expansion of the integrand yielding the following:

$$\frac{1}{ax^2 + bx + c} = \frac{1}{a} \left( \frac{A}{x - x_+} + \frac{B}{x - x_-} \right)$$

Calculating the roots  $x_+$  and  $x_-$  of the denominator:

$$\begin{aligned} ax_{\pm}^2 + bx_{\pm} + c &= 0 & \boxed{a \neq 0} \\ x_{\pm}^2 + \frac{b}{a}x_{\pm} + \frac{c}{a} &= 0 \\ \left(x_{\pm} + \frac{b}{2a}\right)^2 + \frac{c}{a} - \left(\frac{b}{2a}\right)^2 &= 0 \\ \left(x_{\pm} + \frac{b}{2a}\right)^2 &= \left(\frac{b}{2a}\right)^2 - \frac{c}{a} \\ x_{\pm} &= \pm \sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}} - \frac{b}{2a} \end{aligned}$$

And for  $A$  and  $B$ :

$$\begin{aligned} \frac{1}{a} \frac{1}{(x - x_+)(x - x_-)} &= \frac{1}{a} \left( \frac{A}{x - x_+} + \frac{B}{x - x_-} \right) \\ 1 &= A(x - x_-) + B(x - x_+) \quad \forall x \\ \rightarrow 0 &= A + B \quad 1 + Ax_- + Bx_+ = 0 \\ B &= -A \quad 1 + A(x_- - x_+) = 0 \\ A &= \frac{1}{x_+ - x_-} = \frac{1}{2\sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}}} \\ B &= \frac{-1}{2\sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}}} \end{aligned}$$

Thus:

$$\frac{1}{ax^2 + bx + c} = \frac{1}{a} \frac{1}{2\sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}}} \left( \frac{1}{x - \sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}} + \frac{b}{2a}} - \frac{1}{x + \sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}} + \frac{b}{2a}} \right)$$

If  $\sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}}$  real i.e.  $b^2 > 4ac$  integration is possible using the relation:

$$\int \frac{1}{x - \alpha} dx = \ln|x - \alpha| + C \quad \text{assuming } \alpha \in \mathbb{R}$$

$$\int_{x_{st}}^{x_{end}} \frac{dx}{ax^2 + bx + c} = \frac{1}{a} \frac{1}{2\sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}}} \left[ \ln \left| x - \sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}} + \frac{b}{2a} \right| \right. \\ \left. - \ln \left| x + \sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}} + \frac{b}{2a} \right| \right]_{x_{st}}^{x_{end}}$$

because  $\ln |a| - \ln |b| = \ln \left| \frac{a}{b} \right| = \ln \left| \frac{a}{b} \right|$

$$\int_{x_{st}}^{x_{end}} \frac{dx}{ax^2 + bx + c} = \\ \frac{1}{a} \frac{1}{2\sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}}} \ln \left| \frac{(x_{end} - \sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}} + \frac{b}{2a})(x_{st} + \sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}} + \frac{b}{2a})}{(x_{st} - \sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}} + \frac{b}{2a})(x_{end} + \sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}} + \frac{b}{2a})} \right| = \\ \frac{1}{\sqrt{b^2 - 4ac}} \ln \left| \frac{(2ax_{end} - \sqrt{b^2 - 4ac} + b)(2ax_{st} + \sqrt{b^2 - 4ac} + b)}{(2ax_{st} - \sqrt{b^2 - 4ac} + b)(2ax_{end} + \sqrt{b^2 - 4ac} + b)} \right|$$

This result is true for  $a \neq 0$  and  $4ac - b^2 < 0$ .

What needs to be done to get the other cases?

Looking at a table of basic integrals the following is found:

$$\int \frac{1}{1+x^2} dx = \arctan(x) + C \quad x \in \mathbb{R}$$

The function  $\frac{1}{ax^2+bx+c}$  can be reshaped in order to match this form  $\frac{1}{1+x^2}$  with a prefactor and x being rescaled:

$$\frac{1}{ax^2 + bx + c} = \frac{1}{a} \frac{1}{(x + b/2a)^2 - (b/2a)^2 + c/a} \quad \boxed{a \neq 0} \\ = \frac{1}{a} \frac{1}{c/a - (b/2a)^2} \frac{1}{\frac{(x+b/2a)^2}{c/a - (b/2a)^2} + 1} \\ = \frac{1}{a} \frac{1}{c/a - (b/2a)^2} \frac{1}{\left(\frac{x+b/2a}{\sqrt{c/a - (b/2a)^2}}\right)^2 + 1}$$

Using the former expression then yields:

$$\int \frac{1}{ax^2 + bx + c} dx = \frac{1}{a} \frac{1}{c/a - (b/2a)^2} \int \frac{1}{\left(\frac{x+b/2a}{\sqrt{c/a - (b/2a)^2}}\right)^2 + 1} dx \\ = \frac{1}{a} \frac{\sqrt{c/a - (b/2a)^2}}{c/a - (b/2a)^2} \int \frac{1}{z^2 + 1} dz \quad \boxed{\text{if } c/a - (b/2a)^2 > 0} \\ = \frac{1}{a} \frac{1}{\sqrt{c/a - (b/2a)^2}} \left[ \arctan \left( \frac{x + b/2a}{\sqrt{c/a - (b/2a)^2}} \right) + C \right]$$

$$\int_{x_{st}}^{x_{end}} \frac{1}{ax^2 + bx + c} dx = \frac{2}{\sqrt{4ac - b^2}} \left[ \arctan \left( \frac{2ax_{end} + b}{\sqrt{4ac - b^2}} \right) - \arctan \left( \frac{2ax_{st} + b}{\sqrt{4ac - b^2}} \right) \right]$$

This result is true for  $a \neq 0$  and  $4ac - b^2 > 0$ .

What if  $4ac - b^2 = 0$ ?

$$\frac{1}{ax^2 + bx + c} = \frac{1}{a(x + b/2a)^2 - (b/2a)^2 + c/a} \quad \boxed{a \neq 0}$$

$$= \frac{1}{a(x + b/2a)^2}$$

$$\int_{x_{st}}^{x_{end}} \frac{dx}{ax^2 + bx + c} = \frac{1}{a} \int_{x_{st}}^{x_{end}} \frac{dx}{(x + b/2a)^2}$$

$$= \frac{-1}{a} \left[ \frac{1}{x + b/2a} \right]_{x_{st}}^{x_{end}}$$

$$= \left[ \frac{1}{ax_{st} + b/2} - \frac{1}{ax_{end} + b/2} \right]$$

For the case  $a = 0$

$$\int_{x_{st}}^{x_{end}} \frac{dx}{ax^2 + bx + c} = \int_{x_{st}}^{x_{end}} \frac{dx}{bx + c} \quad \boxed{b \neq 0}$$

$$= \frac{1}{b} [\ln |bx + c|]_{x_{st}}^{x_{end}}$$

$$= \frac{1}{b} \ln \left| \frac{bx_{end} + c}{bx_{st} + c} \right|$$

For the case  $a = 0$  and  $b = 0$

$$\int_{x_{st}}^{x_{end}} \frac{dx}{ax^2 + bx + c} = \int_{x_{st}}^{x_{end}} \frac{dx}{c} \quad \boxed{c \neq 0}$$

$$= \frac{x_{end} - x_{st}}{c}$$

For the case  $a = 0$  and  $b = 0$  and  $c = 0$  the integrand is undefined.

Summing all this up one gets:

$$\int_{x_{st}}^{x_{end}} \frac{dx}{ax^2 + bx + c} =$$

$\left\{ \begin{array}{l} \text{undefined} \\ \frac{x_{end} - x_{st}}{c} \\ \frac{1}{b} \ln \left  \frac{bx_{end} + c}{bx_{st} + c} \right  \\ \left[ \frac{1}{ax_{st} + b/2} - \frac{1}{ax_{end} + b/2} \right] \\ \frac{2}{\sqrt{4ac - b^2}} \left[ \arctan \left( \frac{2ax_{end} + b}{\sqrt{4ac - b^2}} \right) - \arctan \left( \frac{2ax_{st} + b}{\sqrt{4ac - b^2}} \right) \right] \\ \frac{1}{\sqrt{b^2 - 4ac}} \ln \left  \frac{(2ax_{end} - \sqrt{b^2 - 4ac} + b)(2ax_{st} + \sqrt{b^2 - 4ac} + b)}{(2ax_{st} - \sqrt{b^2 - 4ac} + b)(2ax_{end} + \sqrt{b^2 - 4ac} + b)} \right  \end{array} \right.$	$\begin{array}{l} \text{if } a = 0, b = 0 \text{ and } c = 0 \\ \text{if } a = 0, b = 0 \text{ and } c \neq 0 \\ \text{if } a = 0 \text{ and } b \neq 0 \\ \text{if } a \neq 0 \text{ and } 4ac - b^2 = 0 \\ \text{if } a \neq 0 \text{ and } 4ac - b^2 > 0 \\ \text{if } a \neq 0 \text{ and } 4ac - b^2 < 0 \end{array}$
--	--

### 3.2.2 Inspecting the solutions

Relating the variables of the model with the variables used in the evaluation of the integral:

$$\begin{aligned} \text{prefactor} &: \frac{A}{L\bar{a}} \\ a &= m_x^2 + m_y^2 \\ b &= 2(m_x b_x + m_y b_y) \\ c &= b_x^2 + b_y^2 + 1/\bar{a} \end{aligned}$$

The condition  $4ac \begin{cases} < \\ = \\ > \end{cases} b^2$  becomes:

$$\begin{aligned} &4 * (m_x^2 + m_y^2) * (b_x^2 + b_y^2 + 1/\bar{a}) \begin{cases} < \\ = \\ > \end{cases} 4 * (m_x b_x + m_y b_y)^2 \\ &m_x^2 b_x^2 + m_y^2 b_y^2 + m_x^2 b_y^2 + m_y^2 b_x^2 + \frac{m_x^2 + m_y^2}{\bar{a}} \begin{cases} < \\ = \\ > \end{cases} m_x^2 b_x^2 + 2m_x m_y b_x b_y + m_y^2 b_y^2 \\ &m_x^2 b_y^2 + m_y^2 b_x^2 - 2m_x m_y b_x b_y + \frac{m_x^2 + m_y^2}{\bar{a}} \begin{cases} < \\ = \\ > \end{cases} 0 \\ &(m_x b_y - m_y b_x)^2 \begin{cases} < \\ = \\ > \end{cases} - \frac{m_x^2 + m_y^2}{\bar{a}} \end{aligned}$$

Obviously  $(m_x b_y - m_y b_x)^2 \geq 0$ . Also  $m_x^2 \geq 0$  and  $m_y^2 \geq 0$ . Assuming  $\bar{a} > 0$  which will be seen later, it follows that  $\frac{m_x^2 + m_y^2}{\bar{a}} \geq 0$ .

Apparently some of the solutions of the integral aren't relevant for the specific case in the model:

- Looking at the above relations it follows that  $4ac < b^2$  is never true.
- If  $4ac = b^2$  then  $m_x = 0$  and  $m_y = 0$  thus also  $a = 0$ . The case  $4ac = b^2$  with  $a \neq 0$  doesn't occur.
- If  $a = 0$  then  $m_x^2 + m_y^2 = 0$ . This can only be true if  $m_x = 0$  and  $m_y = 0$ . In that case  $b = 2(m_x b_x + m_y b_y) = 0$ . The case with  $a = 0$  and  $b \neq 0$  doesn't occur.
- Because  $\bar{a} > 0$  and  $b_x^2 + b_y^2 \geq 0$  it follows that  $c \neq 0$ .

There are two cases left that actually do occur in the calculations for the model:

- In the case  $a = 0$  and  $b = 0$  it follows that  $m_x = 0$  and  $m_y = 0$ .
- The case  $4ac > b^2$  is true in every other scenario.

### 3.2.3 Applying to model

Applying the solution of the integral in the case  $m_x \neq 0$  or  $m_y \neq 0$  gives<sup>3</sup>:

$$\mathcal{Q} = \frac{A}{L\bar{a}} \frac{2}{\sqrt{4(m_x^2 + m_y^2)(b_x^2 + b_y^2 + 1/\bar{a}) - 4(m_x b_x + m_y b_y)^2}} * \left[ \arctan \left( \frac{2(m_x^2 + m_y^2)l' + 2(m_x b_x + m_y b_y)}{\sqrt{4(m_x^2 + m_y^2)(b_x^2 + b_y^2 + 1/\bar{a}) - 4(m_x b_x + m_y b_y)^2}} \right) \right]_{l'=l_{st}}^{l_{end}}$$

$$\mathcal{Q} = \frac{A}{L\bar{a}} \frac{1}{\sqrt{(b_x m_y - b_y m_x)^2 + (m_x^2 + m_y^2)/\bar{a}}} * \left[ \arctan \left( \frac{(m_x^2 + m_y^2)l' + m_x b_x + m_y b_y}{\sqrt{(b_x m_y - b_y m_x)^2 + (m_x^2 + m_y^2)/\bar{a}}} \right) \right]_{l'=l_{st}}^{l_{end}}$$

For the case  $m_x = 0$  and  $m_y = 0$ :

$$\mathcal{Q} = \frac{A}{L\bar{a}} \frac{l_{end} - l_{st}}{b_x^2 + b_y^2 + 1/\bar{a}}$$

From now on this model will be referred to as integration model (IM).

<sup>3</sup>From now on the subscript  $arb$  is omitted because  $\mathcal{Q}_{arb}$  describes both the perpendicular and the arbitrary case with the two distinctions for  $m_x^2 + m_y^2 (= \neq) 0$

## Chapter 4

# The code

Having derived a new expression to calculate the generated charge, the moment seems right to have a look at how the SM is implemented in code. It looks as follows:

```

:
for (Int_t i=0; i<fNumberOfSegments; ++i) {
:
  sPoint = &fSignalPoints[i];
  xCentre = sPoint->x;
  yCentre = sPoint->y;
:
  Float_t totCharge = (
    sPoint->charge * fLorentzNorm *
    (0.5*fPar0*fPar1/TMath::Pi())/
    TMath::Max(1.e-10,
      ((xCurrent-xCentre)*(xCurrent-xCentre)+
        (yCurrent-yCentre)*(yCurrent-yCentre))/
      fPixelSize/fPixelSize+0.25*fPar1*fPar1
    );
:
}
:
1
```

This code is executed per pixel.

---

<sup>1</sup>*sPoint* → *charge* has the same value for every segment. From now on this value will be called *charge*. Tests that have been performed prior to this thesis indicated that the fluctuations among the segments of the same track can be neglected.

To enhance the readability the following pseudo-code is used:

```

loop over segments{
  xCentre = xCoordinatesOfThisSegment;
  yCentre = yCoordinatesOfThisSegment;
  pixelCharge+ =  $\frac{1}{2\pi} * charge * fLorentzNorm * fPar0 * fPar1*$ 
  *  $\frac{1}{\frac{(xCurren-xCentre)^2}{fPixelSize^2} + \frac{(yCurrent-yCentre)^2}{fPixelSize^2} + 0.25 * fPar1^2}$ ;
}

```

In the model discussion this is analogous with:

$$Q_{arb} = \sum_n q_i = \sum_n \frac{\bar{A}}{\bar{a} * r_i^2 + 1}$$

In order to apply the model alterations one has to relate the variables used in the model discussion with the variables used in the code and replace the summation with the result of the integral. That is remove the segment loop and replace the addend of *pixelCharge*.

## 4.1 Relating the variables

Relating the variables used in the code to the variables used in the model discussion gives:

$$\bar{A} = \frac{\frac{1}{2\pi} * charge * fLorentzNorm * fPar0 * fPar1}{0.25 * fPar1^2}$$

$$\bar{a} = \frac{1}{0.25 * fPar1^2 * fPixelSize^2}$$

$$r_i^2 = (xCurren - xCentre_i)^2 + (yCurrent - yCentre_i)^2$$

Note that xCentre and yCentre are dependent on the segment index. <sup>2</sup>

Earlier *r* was expressed as a function of *l* whose variables *m<sub>x</sub>*, *m<sub>y</sub>*, *b<sub>x</sub>* and *b<sub>y</sub>* remain to be determined. This is performed in subsection 4.1.2.

The physical meaning of some of the variables found in the code are explained in section 4.5.

---

<sup>2</sup>The values for the *\_Center* variables are calculated in the procedure *CbmMvdDigitizeL :: ProduceIonisationPoints* using one line equation for each direction. The values are stored in an array ready to be used by the charge calculation later.

### 4.1.1 The prefactor

$\bar{A} = \frac{l}{L}A$  is given. For  $\mathcal{Q}$  the term  $\frac{A}{L} = \frac{\bar{A}}{l}$  is needed with  $l$  being the segment length whose name in the code is *fSegmentLength*.

From the code one finds the following identities<sup>3</sup>:

$$\begin{aligned} \text{charge} &= \frac{\text{Landau} * \text{trackLength}}{\text{fEpiTh} * \text{fNumberOfSegments}} \\ \text{trackLength} &= \text{fSegmentLength} * \text{fNumberOfSegments} \end{aligned}$$

Therefore:

$$\begin{aligned} \frac{\text{charge}}{\text{fSegmentLength}} &= \frac{\text{Landau} * \text{trackLength}}{\text{fEpiTh} * \text{fNumberOfSegments} * \text{fSegmentLength}} \\ &= \frac{\text{Landau}}{\text{fEpiTh}} \end{aligned}$$

It follows:

$$\boxed{\frac{A}{L} = \frac{2 * \text{Landau} * \text{fLorentzNorm} * \text{fPar0} * \text{fPar1}}{\pi * \text{fEpiTh} * \text{fPar1}^2}}$$

### 4.1.2 Fitting the line equation

The next step is to determine the values for  $m_x$  and  $b_x$  as well as for  $m_y$  and  $b_y$ . First note the following equations:

$$\begin{aligned} x(l) &= m_x l + b_x \\ y(l) &= m_y l + b_y \\ z(l) &= m_z l + b_z \quad ^4 \end{aligned}$$

Let  $\left\{ \begin{array}{cccc} x_{st}, & y_{st}, & z_{st}, & l_{st} \\ x_{end}, & y_{end}, & z_{end}, & l_{end} \end{array} \right\}$  be the coordinates describing the epitaxial layer  $\left\{ \begin{array}{c} \text{entry} \\ \text{exit} \end{array} \right\}$  point of the particle.

Also:

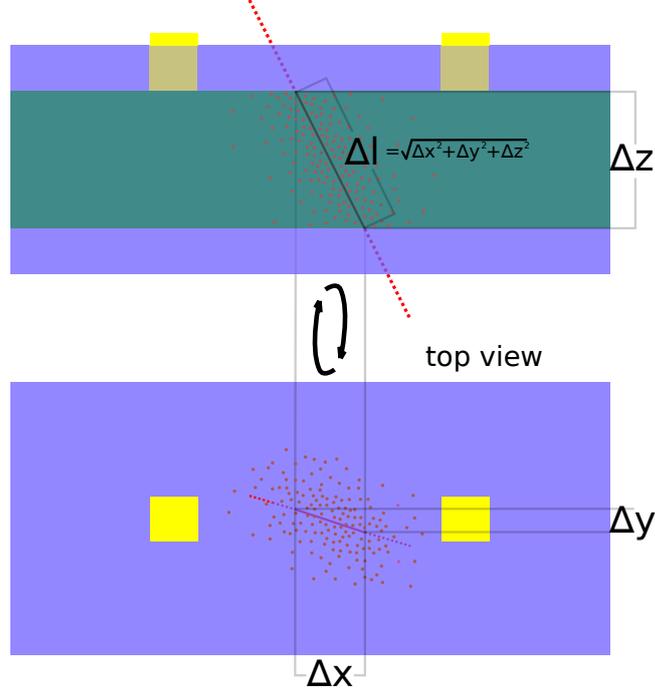
$$\Delta l^2 = (l_{end} - l_{st})^2 = \Delta x^2 + \Delta y^2 + \Delta z^2$$

This visualized in Figure 4.1.

<sup>3</sup>*Landau* is the output of the function `fLandauRandom→Landau(fLandauGain,fLandauSigma/fLandauMPV)`

<sup>4</sup>Note: While the integrand of  $\mathcal{Q}_{arb}$  has no dependence on the direction perpendicular to the chip the integration path does depend on it.

Figure 4.1: Parametrization (note:  $\Delta l$  is the track length in three dimensions)



$$\begin{aligned}
 \Delta x &= x_{end} - x_{st} = x(l_{end}) - x(l_{st}) \\
 &= (m_x l_{end} + b_x) - (m_x l_{st} + b_x) \\
 &= m_x (l_{end} - l_{st}) = m_x \Delta l \quad \rightarrow \quad m_x = \frac{\Delta x}{\Delta l} \\
 x_{st} &= x(l_{st}) = m_x l_{st} + b_x \quad \rightarrow \quad b_x = x_{st} - m_x l_{st}
 \end{aligned}$$

analogously for  $y$  and  $z$ :

$$\begin{aligned}
 m_y &= \frac{\Delta y}{\Delta l} & b_y &= y_{st} - m_y l_{st} \\
 m_z &= \frac{\Delta z}{\Delta l} & b_z &= z_{st} - m_z l_{st}
 \end{aligned}$$

It is convenient to substitute all occurrences of  $l$  by  $z$ . This abolishes the need for  $l$ . Furthermore all tracks enter the detector plane on one side and leave it on the other side. Consequently for  $z$  the integral boundaries are the same for all the tracks and  $\Delta z$  is a constant equal to the epitaxial layer thickness.

$$\begin{aligned}
 z(l) &= m_z l + b_z \quad \rightarrow \quad l(z) = \frac{z - b_z}{m_z} \\
 x(l(z)) &= m_x l(z) + b_x = m_x \frac{z - b_z}{m_z} + b_x = \bar{m}_x z + \bar{b}_x
 \end{aligned}$$

with

$$\bar{m}_x = \frac{m_x}{m_z} = \frac{\frac{\Delta x}{\Delta l}}{\frac{\Delta z}{\Delta l}} = \frac{\Delta x}{\Delta z}$$

and

$$\bar{b}_x = b_x - b_z \frac{m_x}{m_z} = x_{st} - m_x l_{st} - (z_{st} - m_z l_{st}) \frac{m_x}{m_z} = x_{st} - \frac{m_x}{m_z} z_{st} = x_{st} - \bar{m}_x z_{st}$$

The following conditions relate this arbitrary discussion to the code.<sup>5</sup>:

$$\begin{aligned} xCurrent - xCentre_{st} &= x_{st} \\ xCurrent - xCentre_{end} &= x_{end} \end{aligned}$$

Thus:

$$\begin{aligned} \bar{m}_x &= \frac{\Delta x}{\Delta z} = \frac{x_{end} - x_{st}}{\Delta z} \\ &= \frac{(xCurrent - xCentre_{end}) - (xCurrent - xCentre_{st})}{\Delta z} \\ &= \frac{xCentre_{st} - xCentre_{end}}{\Delta z} \end{aligned}$$

$$\bar{b}_x = x_{st} - \bar{m}_x z_{st} = xCurrent - xCentre_{st} - \bar{m}_x z_{st}$$

analogously:

$$\bar{m}_y = \frac{yCentre_{st} - yCentre_{end}}{\Delta z}$$

$$\bar{b}_y = yCurrent - yCentre_{st} - \bar{m}_y z_{st}$$

### 4.1.3 Integration variable

At last the integration variable has to be substituted

$$dz(l) = \frac{\partial z(l)}{\partial l} dl = m_z dl = \frac{\Delta z}{\Delta l} dl \quad \rightarrow \quad dl = \frac{\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}{\Delta z} dz$$

---

<sup>5</sup>In the SM code, values for  $\_Centre_{st/end}$  are only available in the function `CbmMvdDigitizeL::ProduceIonizationPoints`. In the IM code the points are available through the struct `EpiHit`

## 4.2 Deriving the final expression

For  $m_x \neq 0$  or  $m_y \neq 0$ :

$$\begin{aligned} \mathcal{Q} &= \frac{A}{L\bar{a}} \int_{z'=z_{st}}^{z_{end}} \frac{\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}{\Delta z} dz' \\ \mathcal{Q} &= \frac{A}{L\bar{a}} \frac{\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}{\Delta z} \frac{1}{\sqrt{(b_x \bar{m}_y - b_y \bar{m}_x)^2 + (\bar{m}_x^2 + \bar{m}_y^2)/\bar{a}}} * \\ & * \left[ \arctan \left( \frac{(\bar{m}_x^2 + \bar{m}_y^2)z' + \bar{m}_x b_x + \bar{m}_y b_y}{\sqrt{(b_x \bar{m}_y - b_y \bar{m}_x)^2 + (\bar{m}_x^2 + \bar{m}_y^2)/\bar{a}}} \right) \right]_{z'=z_{st}}^{z_{end}} \end{aligned}$$

For  $m_x = 0$  and  $m_y = 0$ :

$$\begin{aligned} \mathcal{Q} &= \frac{A}{L\bar{a}} \int_{z'=z_{st}}^{z_{end}} \frac{\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}{\Delta z} dz' \\ &= \frac{A}{L\bar{a}} \frac{\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}{\Delta z} \frac{z_{end} - z_{st}}{c} \\ &= \frac{A}{L\bar{a}} \frac{\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}{c} \end{aligned}$$

with:

$$\begin{aligned} \frac{A}{L} &= \frac{2 * Landau * fLorentzNorm * fPar0 * fPar1}{\pi * fEpiTh * fPar1^2} \\ \bar{a} &= \frac{1}{0.25 * fPar1^2 * fPixelSize^2} \\ \bar{m}_x &= \frac{xCentre_{st} - xCentre_{end}}{\Delta z} \\ \bar{m}_y &= \frac{yCentre_{st} - yCentre_{end}}{\Delta z} \\ \bar{b}_x &= xCurrent - xCentre_{st} - \bar{m}_x z_{st} \\ \bar{b}_y &= yCurrent - yCentre_{st} - \bar{m}_y z_{st} \\ \Delta x &= xCentre_{st} - xCentre_{end} \\ \Delta y &= yCentre_{st} - yCentre_{end} \\ \Delta z &= zCentre_{st} - zCentre_{end} \end{aligned}$$

## 4.3 The new code

At first a class for integrating over  $\frac{1}{ax^2+bx+c}$  was implemented:

```
class IntegralOfReciprocalOfQuadratic {
public:
    Double_t a, b, c,
             x_st, x_end;
    Double_t operator()();
};
```

At the heart of this class is the function operator():

```

inline Double_t CbmMvdDigitizeL::
IntegralOfReciprocalOfQuadratic::operator()() {
if(a==0.)
  if(b==0.)
    if(c==0.) {
      cout << "ERROR a=b=c=0 IN __FILE__ AT __LINE__";
      cout << endl;
      return((Double_t)0.);
    }
    else // particles impinging perfectly perpendicularly
      return((Double_t)(x_end-x_st )/c);
  else
    return((Double_t)log(fabs((b*x_end+c)/
                              (b*x_st +c)))/b);
else {
  double radicant=4.*a*c-b*b;
  if(radicant==0.)
    return((Double_t)1.0/(a*x_st +b/2.)-
            1.0/(a*x_end+b/2.));
  else if(radicant>0.) { // is valid for all
    double root=sqrt(radicant); // inclined tracks
    return((Double_t)2./root*
            (atan((float)((2.*a*x_end+b)/root))-
             atan((float)((2.*a*x_st +b)/root))));
  } else if(radicant<0.) {
    radicant=-radicant;
    double root=sqrt(radicant);
    return((Double_t)1./root*log(
            fabs(((2*a*x_end+b-root)*(2*a*x_st +b+root))/
                 ((2*a*x_end+b+root)*(2*a*x_st +b-root)))));
  } else {
    cout << "ERROR: UNHANDLED CASE IN __FILE__ AT __LINE__";
    return((Double_t)0.);
  }
}
cout << "ERROR: UNHANDLED CASE IN __FILE__ AT __LINE__";
return((Double_t)0.);
};

```

Once the variables  $a$ ,  $b$ ,  $c$ ,  $x_{st}$ ,  $x_{end}$  are initialized the function operator()() returns the value of the definite integral:

$$\int_{x_{st}}^{x_{end}} \frac{dx}{ax^2 + bx + c}$$

For the application of calculating the generated charge  $Q$  it is arduous to use the parameters provided by the former class (*IntegralOfReciprocalOfQuadratic*). That's why a wrapper class around the former class is provided that takes a more natural set of parameters. These use the same names as the variables that are found in the model discussion:

```
class IntegralOfReciprocalOfQuadraticWrap
  : public IntegralOfReciprocalOfQuadratic {
protected:
  Double_t xmin, ymin, zmin, xmax, ymax, zmax;
  Double_t m_x, b_x_plus_PixelPosX;
  Double_t m_y, b_y_plus_PixelPosY;
  Double_t length, l_over_lz;
public:
  Double_t A_over_L_over_a_bar,
          fPar1_sq_times_fPixelSize_sq_over_4;
  void init(Double_t _xmin, Double_t _ymin,
            Double_t _zmin, Double_t _xmax,
            Double_t _ymax, Double_t _zmax);
  Double_t operator()(Double_t PixelPosX,
                     Double_t PixelPosY);
} LorentzIntegrator;
```

The variables  $A_{over}L_{over}a_{bar}$  and  $fPar1_{sq}times_{f}PixelSize_{sq}over_4$  need to be initialized by hand. Further initialization is done by calling the *init* method passing all the entry and exit coordinates of the particle. Having done that the *operator()* method which takes the pixel coordinates as argument returns the generated charge  $Q$ .

## 4.4 Call tree comparison

Along with the alterations that have been performed to turn the SM into the IM implementation, numerous other changes to the code have been made. Most of them aim for a better readability. Bigger functions were split up into smaller functions with descriptive names summarizing their intent. Where possible global variables have been eliminated in favor of local ones emphasizing transfer of data among the functions. This ought to clarify the flow of information.

The following table shows some of the main alterations by comparing the prior with the altered version of the code. It is structured in a call tree which also resembles the nesting of the functions. Entries that end with brackets () are actual functions that can be found in the corresponding code. Entries without brackets just summarize what is done.

SM Code	IM Code
Exec()	Exec()
BuildEvent()	Build Event()
StationLoop	StationLoop
HitLoop	HitLoop
ProduceIonizationPoints()	CalculateHitCoordinates()
calculate track entry/ exit/length coordinates	calculate track entry/ exit/length coordinates
createSegments	
ProducePixelCharge()	CalculatePixelActivation()
get coordinates of surrounding square	GetSurrounding- SquareCoordinates()
get surrounding square for each segment	
PixelLoop	PixelLoop
SegmentLoop	
calculate charge for this segment	calculate charge using Lorentzintegrator
	AddPixel()
add charge to pixel	create pixel
	call DigestCharge()
call DigestCharge() for new pixels	
fill array	CreateDigisArray()

All code regarding the segments has been removed. In the procedure *ProduceIonizationPoints()* the calculation of the segment positions has been removed. It has been renamed *CalculateHitCoordinates()* as it still calculates the entry and exit points of the epitaxial layer.

## 4.5 Physical meaning of the variables

Some of the variables that are encountered in the code are explained here from a physical point of view:

- *Landau = LandauCh*: This variable is randomly drawn from a Landau distribution for every track. It represents the charge that the particle separates when passing through silicon on a length of *fEpiTh*.
- *fEpiTh*: The thickness of the epitaxial layer
- *fPixelSize<sup>2</sup>*: The area of one pixel
- *fPar0*: One fit parameter obtained through experimental data
- *fPar1*: Second fit parameter obtained through experimental data

## 4.6 Options for further optimization

The following bullets give some ideas about where further optimization is possible in the IM implementation.

- A tradeoff needs to be made between the speed and the amount of pixels recorded. The threshold that determines whether a pixel is recorded or not is called *negligibleChargePerPixel*. It is currently set to 15e. This results in about the same overall number of recorded pixels when comparing the SM with the IM implementation. Note that when lowering this threshold one may also consider increasing the area around the track that is checked for activation. This is done in the function called *GetSurroundingSquareCoordinates*.
- Another spot where speed may be gained at the cost of accuracy is the condition that decides whether a track is treated as being perpendicular or not<sup>6</sup>. One can see in Table 5.5 that for angles below 15° the difference between using the arbitrary case or the perpendicular case formula only makes a difference of less than 10 percent.

---

<sup>6</sup>This condition is marked in the code with the comment "// OPTIMIZATION POINT"

## Chapter 5

# Validation and benchmarking

The testing of the IM implementation is split in two main sections:

- In a validation phase it was confirmed that the SM and the IM yield the same result.
- A benchmark was carried out to determine the speed-up of the altered implementation with respect to the original code.

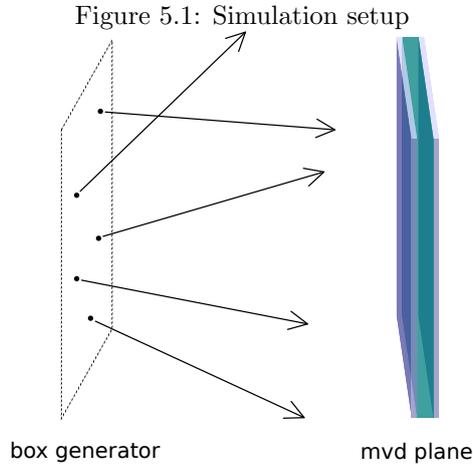
### 5.1 Validation

In order to verify that the IM implementation is equivalent to the SM one, some simulations were performed. During the simulations the charge  $Q$  that is calculated using the previously discussed models was recorded. This was done with both implementations using the same input data.

#### 5.1.1 Simulation setup

A special experimental setup was used in the validation simulations. A description of this setup is given to the simulation software i.e. CBMROOT in the form of a macro. With this description the simulation software can predict for instance a detector response.

The setup for the simulation consists of a so-called "box generator" which is explained further down and one mvd detector plane. This configuration is shown in Figure 5.1.



The "box generator" is a software component that acts as a rectangle in space which emits particles. In the simulation it is created in a macro that also sets several parameters for it. Among these parameters are:

- the size of the rectangle
- the location of the rectangle
- the angle of inclination for the emitted particles

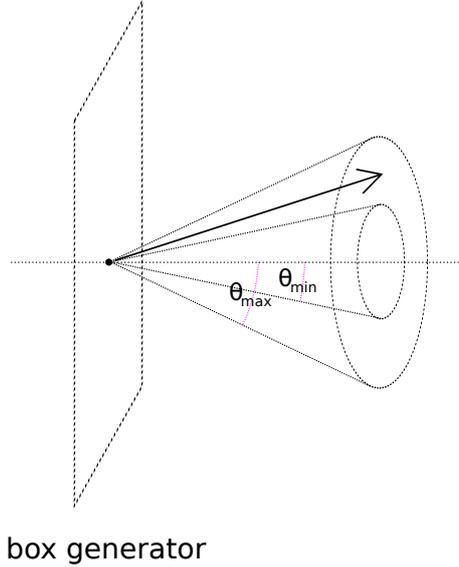
The size and location of the rectangle is of importance because the starting point of the particles is selected randomly from the surface of the rectangle. The inclination of the particles is of special interest for this thesis because the IM needs to replicate the information of the SM for all angles. It is selected randomly to be within the given range. This range of the angle is visualized in Figure 5.2. The outer cone carved by the inner cone represents the volume where particles may travel through.

The momentum is constant for all the generated particles. The type of the particles isn't specified. All particles arriving at the mvd are assumed to be singly charged. Other particles are not accounted for.

The second component in the simulation is the mvd detector plane. It is placed so that it is hit by the particles coming from the "box generator" for all inclinations used during the simulations.

Given the placement and the parameters of the two components i.e. the "box generator" and the mvd detector plane the software is able to conduct the simulation. It does so by generating particles using the "box generator" then calculating the points where those particles penetrate the detector plane. Finally it passes the former points to the digitizer which calculates the pixel activation. In this simulation both tasks, the particle/track generation and the digitizing, are handled separately by two different macros. The output of the "box generator" macro is stored in a file which is then fed into the macro starting the digitizing.

Figure 5.2: Box generator range of angle



### 5.1.2 Further implementations

Apart from the implementations of the SM and IM two other implementations have been tested. They have been called slow segmentation model (SSM) and perpendicular model (PM).

The implementation called SSM is to show that the deviations between the SM and the IM are actually caused by an inaccuracy of the SM.

First of all in the SM, charge contributions of less than 1e per pixel per segment are neglected.

Another difference to the IM implementation is how the pixels around the track are selected. In the SM implementation every single segment has an area around it for whose pixels the charge calculation is performed. If a pixel is outside this area the calculation is skipped. This causes deviations in highly inclined tracks.

At the cost of speed the upper two differences have been removed in the SSM implementation.

The implementation called PM is to show that further optimization is possible. In it all the tracks are treated as being perpendicular to test an option for further optimization. This cuts down on the needed calculation effort as the formula for a perpendicular track is much shorter and doesn't require the calculation of the arctan function. It will be seen later that this yields acceptable results for low inclinations.

### 5.1.3 Test results

In the following tables the results i.e. the calculated charge values of the simulation runs are shown. The unit of these values is the elementary charge (e)<sup>1</sup>. Simulations have been performed for the inclinations of 0°, 15°, 30°, 45°, 60°, 75°. Additionally one simulation contains tracks whose inclinations were randomly chosen to be between 0° and 75° for every track. The range of inclination in degrees is given in the top row of the table for every simulation in the form  $\Theta_{min} - \Theta_{max}$ .

All the different implementations have been tested with all of the upper inclinations. Which implementation was used is indicated in the second table row with the model names SM, IM, SSM and PM.

Only for Table 5.1 some data about the regarded pixel is also shown. It shows:

- an identification number that indicates which particle track caused the activation.
- the pixel index that says in which row and column the pixel is in.

Using this information a visual representation is given in the column called "shape". It shows the arrangement of the pixels that have been activated by one particle.

---

<sup>1</sup>In an experiment the recorded charge values are multiples of 1e. The decimal places in the table can be interpreted as a propability that one further charge is recorded.

Table 5.1: Results of the different response models

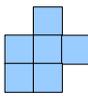
00-75							
trackID	row	column	SM	IM	SSM	PM	shape
2	494	151	12.97	15.90	15.90	15.91	
2	494	152	37.13	37.14	37.13	36.50	
4	1159	172	19.02	19.02	19.02	18.98	
4	1159	173	26.00	26.00	26.00	26.02	
6	1077	1214	17.41	17.41	17.41	17.41	
6	1078	1213	33.88	33.88	33.88	33.86	
6	1078	1214	327.28	327.28	327.28	326.21	
6	1078	1215	15.65	15.65	15.65	15.64	
6	1079	1213	16.01	16.01	16.01	16.00	
6	1079	1214	27.76	27.76	27.76	27.76	
8	974	273	2.07		13.53		
8	975	273	128.26	128.23	128.26	138.60	
9	1589	969	31.72	31.72	31.72	31.46	
9	1590	969	13.28	16.20	16.20	16.15	
11	387	387	58.90	58.90	58.90	57.06	
11	387	388	101.35	101.35	101.35	102.69	
11	388	387	37.45	37.45	37.45	37.65	
11	388	388	54.31	54.32	54.31	53.26	
13	1342	836	57.41	57.42	57.41	54.85	
13	1342	837	35.13	35.14	35.13	34.12	
15	801	1242	18.02	27.55	27.54	22.91	
15	802	1242	374.53	374.38	374.53	269.23	
1	1449	2453	20.33	20.33	20.33	20.32	
1	1450	2453	82.33	82.33	82.33	82.40	
3	1647	1324	566.60	566.46	566.60	608.14	
5	1850	840	80.03	80.03	80.03	80.18	
5	1850	841	16.29	16.29	16.29	16.27	
7	1872	2179	26.25	26.25	26.25	26.25	
10	2847	1885	42.22	42.22	42.22	41.86	
10	2847	1886	18.26	18.26	18.26	18.28	
10	2848	1885	25.92	25.92	25.92	25.83	
10	2848	1886	4.14		14.41		
12	1681	2041	62.32	62.34	62.32	57.29	
12	1681	2042	6.46	16.05	16.05	15.95	
12	1682	2041				15.10	
14	2222	940	40.29	40.29	40.29	39.45	
16	3083	1218	10.23	16.49	16.49	16.27	
16	3084	1218	67.96	67.97	67.96	65.67	
0	448	340	18.23	18.23	18.23	18.52	
0	448	341	67.30	67.30	67.30	67.57	
2	1503	1278	12.94	21.77	21.77	21.75	
2	1503	1279	29.47	35.26	35.25	29.81	
2	1504	1278	4.59	16.24	16.24	15.03	

Table 5.2: Results of the different response models

00-00				15-15			
SM	IM	SSM	PM	SM	IM	SSM	PM
28.49	28.49	28.49	28.49	57.52	57.53	57.52	56.56
30.86	30.86	30.86	30.86	3.04		14.05	
59.64	59.64	59.64	59.64	7.66	15.06	15.06	
71.11	71.11	71.11	71.11	29.62	29.62	29.62	29.75
56.03	56.03	56.03	56.03	31.76	31.76	31.76	31.75
54.49	54.49	54.49	54.49	52.10	52.10	52.10	51.92
19.10	19.10	19.10	19.10	37.93	37.93	37.93	37.73
16.67	16.67	16.67	16.67	69.94	69.94	69.94	70.06
27.88	27.88	27.88	27.88	6.12		14.83	
38.32	38.32	38.32	38.32	38.76	38.77	38.76	38.18
500.70	500.70	500.70	500.70	38.69	38.69	38.69	38.92
18.58	18.58	18.58	18.58	24.80	24.80	24.80	24.86
14.67		14.67		41.37	41.37	41.37	41.22
22.70	22.70	22.70	22.70	31.67	31.67	31.67	31.71
24.20	24.20	24.20	24.20	101.03	101.03	101.03	101.23
54.79	54.79	54.79	54.79	58.78	58.78	58.78	58.31
20.96	20.96	20.96	20.96	8.61	15.20	15.20	15.14
32.73	32.73	32.73	32.73	74.13	74.13	74.13	73.81
121.22	121.22	121.22	121.22	18.85	18.85	18.85	18.89
17.56	17.56	17.56	17.56	34.01	34.01	34.01	34.01
126.96	126.96	126.96	126.96	23.14	23.13	23.14	23.19
18.87	18.87	18.87	18.87	2.03		14.13	
57.75	57.75	57.75	57.75	213.41	213.45	213.41	204.59
23.51	23.51	23.51	23.51	205.39	205.38	205.39	205.06
122.24	122.24	122.24	122.24	15.54	16.53	16.53	16.49
69.44	69.44	69.44	69.44	17.62	17.62	17.62	17.66
28.81	28.81	28.81	28.81	17.83	17.83	17.83	17.86
17.84	17.84	17.84	17.84	33.10	33.10	33.10	32.69
61.72	61.72	61.72	61.72	36.67	36.67	36.67	36.85
62.01	62.01	62.01	62.01	22.70	22.70	22.70	22.75
15.92	15.92	15.92	15.92	59.56	59.56	59.56	59.37
59.99	59.99	59.99	59.99	35.05	35.05	35.05	35.02
88.70	88.70	88.70	88.70	37.87	37.87	37.87	37.70
25.55	25.55	25.55	25.55	18.98	18.98	18.98	19.02
25.88	25.88	25.88	25.88	131.53	131.55	131.53	126.46
49.32	49.32	49.32	49.32	69.64	69.64	69.64	69.19
15.66	15.66	15.66	15.66	30.94	30.94	30.94	30.52
37.80	37.80	37.80	37.80	27.94	27.94	27.94	27.63
23.76	23.76	23.76	23.76	145.40	145.39	145.40	148.60
23.04	23.04	23.04	23.04	4.15		13.84	
14.72		14.72		28.75	28.75	28.75	28.68
23.02	23.02	23.02	23.02	19.98	19.98	19.98	20.06
50.56	50.56	50.56	50.56	26.60	26.60	26.60	26.66

Table 5.3: Results of the different response models

30-30				45-45			
SM	IM	SSM	PM	SM	IM	SSM	PM
28.42	28.42	28.42	27.67	31.26	32.24	32.24	31.92
29.95	29.95	29.95	29.80	40.96	41.95	41.94	37.42
245.68	245.63	245.68	240.89	9.01	18.18	18.17	16.63
47.18	47.18	47.18	47.53	20.33	21.31	21.32	22.12
35.87	35.87	35.87	35.30		17.08		17.55
82.90	82.91	82.90	79.91	21.29	26.89	26.88	23.78
50.84	50.84	50.84	50.50	76.99	76.98	76.99	78.67
48.84	48.85	48.84	45.46	55.15	55.16	55.15	51.50
6.22	16.39	16.39	16.51	1.00		14.95	
57.05	57.05	57.05	56.57	79.93	79.94	79.93	75.14
19.76	19.76	19.76	20.01	50.13	50.13	50.13	49.96
38.06	38.06	38.06	37.59	13.41	20.20	20.20	20.95
29.59	29.59	29.59	29.75	22.23	27.99	27.98	24.67
2.05	15.29	15.29	15.22	7.24	16.67	16.67	15.34
144.07	144.07	144.07	143.76		16.86		17.25
74.43	74.45	74.43	71.59	2.11		14.54	
18.79	18.79	18.79	18.85	21.74	21.74	21.74	22.48
7.57	16.74	16.74	16.65	83.93	83.92	83.93	83.21
30.97	30.97	30.97	31.34	28.64	29.64	29.64	28.44
51.58	51.58	51.58	51.90	98.10	98.10	98.10	99.56
2.07		14.26		25.90	25.90	25.90	25.90
43.88	43.88	43.88	44.25	33.65	33.65	33.65	33.40
21.64	21.63	21.64	21.71	206.67	206.66	206.67	207.48
1.03		13.38		31.26	31.26	31.26	29.96
6.57	16.14	16.14	15.98	6.28	18.77	18.77	18.77
266.74	266.67	266.74	290.37	99.14	99.12	99.14	111.01
3.14		14.89		3.30	15.27	15.26	
370.22	370.10	370.22	431.10	30.85	30.85	30.85	31.93
26.22	26.22	26.22	26.05	297.70	297.56	297.70	237.76
72.88	72.89	72.88	71.01	10.14	18.77	18.76	17.34
97.73	97.71	97.73	103.03	1.02		14.90	
11.62	17.43	17.43	17.53	392.50	392.38	392.50	510.42
13.59	19.00	19.00	18.54	8.12	18.56	18.56	18.12
35.96	35.97	35.96	34.70	12.47	20.55	20.55	20.69
38.25	38.24	38.25	38.90	4.31	16.34	16.34	15.92
160.33	160.29	160.33	173.51	322.95	322.84	322.95	347.68
49.11	49.11	49.11	47.54	1.01		13.77	
42.17	42.18	42.17	39.77	67.07	67.11	67.07	55.01
23.54	24.54	24.54	23.74	8.40	18.90	18.90	19.15
202.49	202.44	202.49	221.04	27.68	27.67	27.68	28.41
24.86	24.86	24.86	24.60	4.44	15.33	15.32	
9.43	16.94	16.94	16.49	163.68	163.71	163.68	117.37
21.83	21.83	21.83	22.16	7.24	17.32	17.32	16.30

Table 5.4: Results of the different response models

60-60				75-75			
SM	IM	SSM	PM	SM	IM	SSM	PM
49.06	53.42	53.43	53.66		22.79		24.78
48.40	53.88	53.88	46.68	85.59	98.94	98.95	38.78
9.34	20.85	20.84	17.25	293.31	301.09	301.10	494.46
	17.74		18.85	15.43	30.89	30.88	18.17
29.92	35.90	35.90	38.54		19.32		15.57
76.64	82.81	82.72	47.56		20.59		24.71
	21.66		22.13		19.23		22.66
	16.66		16.28	91.96	104.28	104.29	67.67
23.66	32.69	32.69	31.00	8.46	25.51	25.51	17.54
175.76	175.73	175.76	213.90	24.40	40.72	40.72	28.32
54.15	56.93	56.91	47.84	109.09	118.21	118.21	167.94
20.33	31.05	31.04	28.12		18.48		20.76
120.74	120.73	120.74	125.30		16.61		
39.48	42.33	42.33	41.31		15.49		
	17.96		18.15		25.63		23.28
	23.95		25.52		26.49		27.48
15.70	25.90	25.89	20.22		18.12		19.04
54.53	61.60	61.54	38.44	6.86	28.64	28.64	23.09
	25.09		27.57	122.50	136.58	136.53	71.14
31.95	40.82	40.80	30.35	103.49	105.38	105.39	133.50
24.08	31.08	31.08	33.69		41.05		42.37
17.79	27.78	27.78	29.66		16.50		16.03
35.66	44.12	44.09	32.83	3.08	30.14	30.14	26.97
	25.52		26.10	130.75	138.75	138.76	127.71
	22.77		22.84	874.64	874.62	874.64	790.80
17.04	28.40	28.40	26.44	74.86	91.55	91.53	57.55
196.03	196.02	196.03	188.61		20.13		17.80
113.35	113.38	113.35	92.77		18.40		18.68
3.16	19.50	19.50	18.43		37.19		41.18
10.89	24.80	24.80	23.36	40.38	58.65	58.65	56.44
97.01	97.01	97.01	97.25	16.49	37.83	37.82	29.56
66.02	66.02	66.02	63.45		15.61		
	17.58		16.88		15.46		
	18.52		18.78		36.05		45.90
	16.84		17.03	9.60	30.68	30.69	25.83
9.13	27.00	27.00	28.20	11.52	26.66	26.65	15.83
12.75	26.21	26.20	21.79	113.84	124.05	124.05	155.59
413.11	413.08	413.11	878.27	62.77	76.49	76.50	42.83
14.99	27.66	27.65	23.90				15.44
	18.71		19.12	72.82	87.15	87.16	40.55
7.67	22.24	22.24	21.83	6.45	28.53	28.53	23.34
35.70	43.70	43.69	34.61		18.46		20.83
18.60	28.99	28.99	28.26	192.00	199.00	199.00	367.24

#### 5.1.4 Discussion of the results

Looking only at the output of the SM and the IM implementation one observes that both implementations generate very similar results. Only in some lightly activated pixels or with highly inclined tracks there are deviations of the order of a few charges. They are caused by an inaccuracy in the SM implementation which firstly ignores contributions of less than  $1e$  per segment per pixel. Secondly every segment has its own limited area around it where charge is being generated which causes deviations for highly inclined tracks.

To proof that the deviations are caused by an inaccuracy of the SM code, the SSM implementation has also been tested. The SSM implementation is explained in subsection 5.1.2. Comparing the SSM to the IM the results match with a precision of a fraction of  $1e$ .

Note that some pixels are missing in one or the other implementation. This is caused by the condition that determines whether or not a pixel or segment per pixel can be neglected. In the SM implementation a charge contribution is neglected if the charge per pixel per segment is less than  $1e$ . In the IM implementation an activated pixel is neglected if the overall charge for that pixel is less than  $15e$ .

For the case of a small inclination this causes some pixels to appear only in the SM implementation. These are pixels that have less than  $15e$  overall charge but single segments that deliver more than  $1e$  of charge for this pixel. For high inclinations the IM records more activated pixels. This is because the IM doesn't neglect contributions where the segment and pixel would be too far apart in the SM implementation. Further explanation about that can be found in subsection 5.1.2.

Looking at the output of the PM implementation which is explained in subsection 5.1.2 one observes that for inclinations of less than  $15^\circ$  the deviations stay below 10 percent. Higher inclinations yield bigger deviations. In Table 5.1.5 the deviations are further quantified.

### 5.1.5 Precision

A more thorough comparison has been conducted between the IM and the SSM implementation. In the following histograms the relative deviation  $\frac{Q_{IM} - Q_{SSM}}{Q_{SSM}} * 100\%$  has been plotted.

For every histogram the first row gives the regarded range of angles. In the second row the number of total entries i.e. the number of compared calculated charge values is given. The mean and RMS follow. The bin size is chosen to be 0.02% for every histogram. Note again that the deviation is given in percent and not as a fraction (0.02% = 0.0002). The height of a bin gives the percentage of the total entries that resides within that bin. Using the percentage instead of the number of entries as the height allows one to easily compare the different histograms with each other.

Figure 5.3: Relative deviation between the results of the IM and the SSM

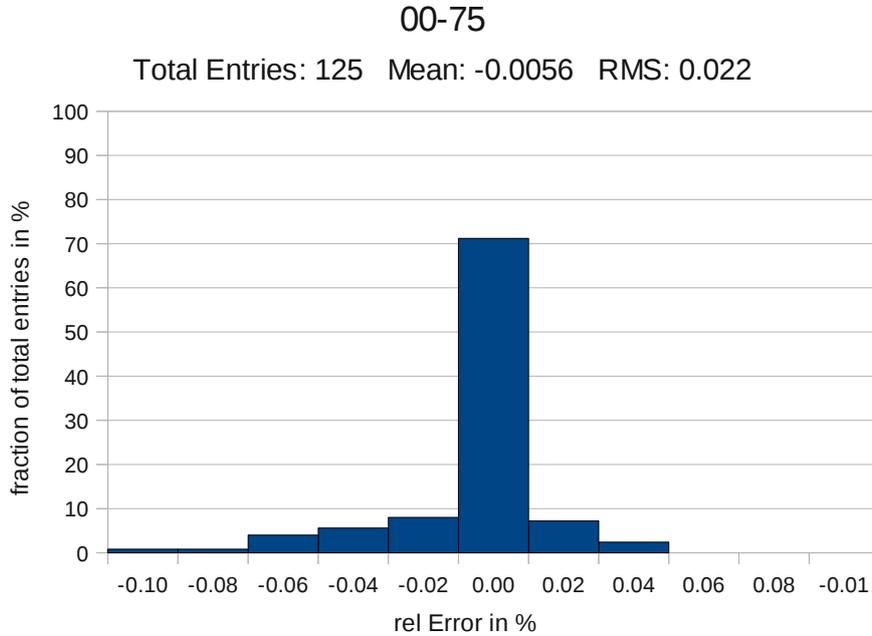


Figure 5.4: Relative deviation between the results of the IM and the SSM

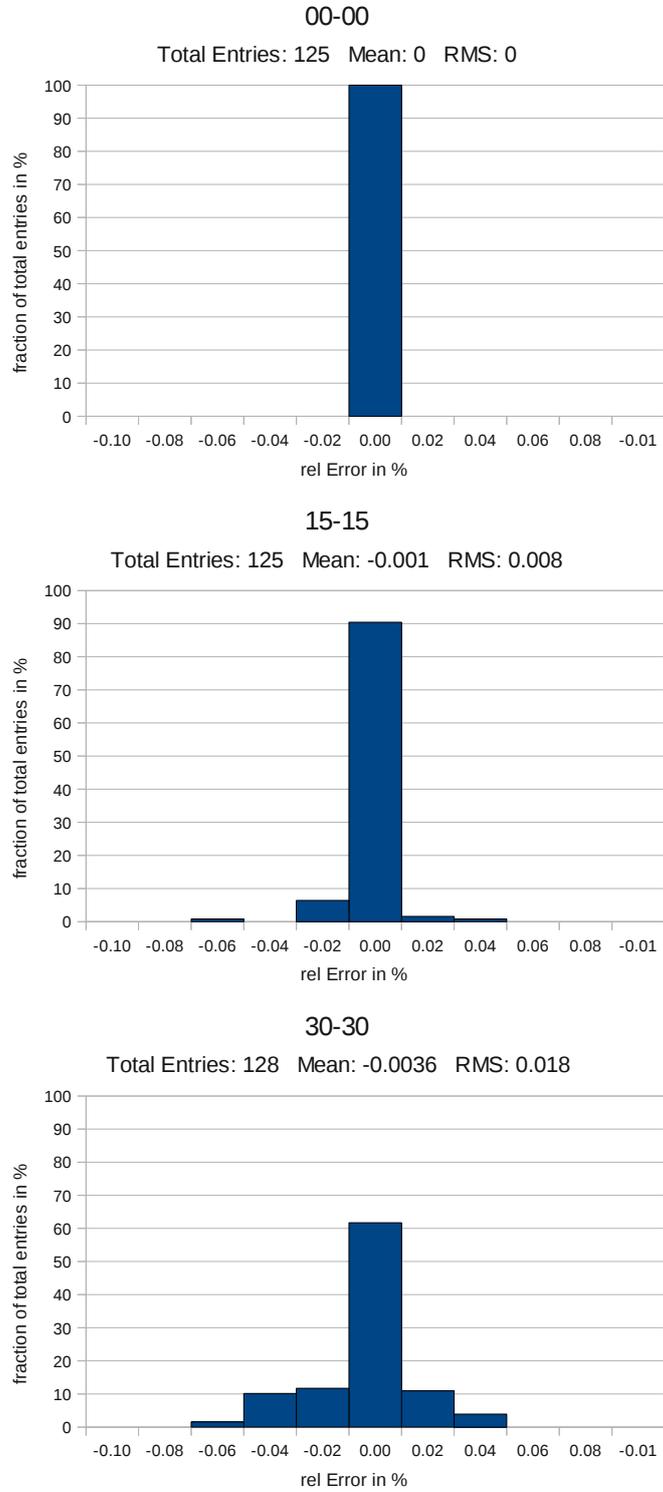
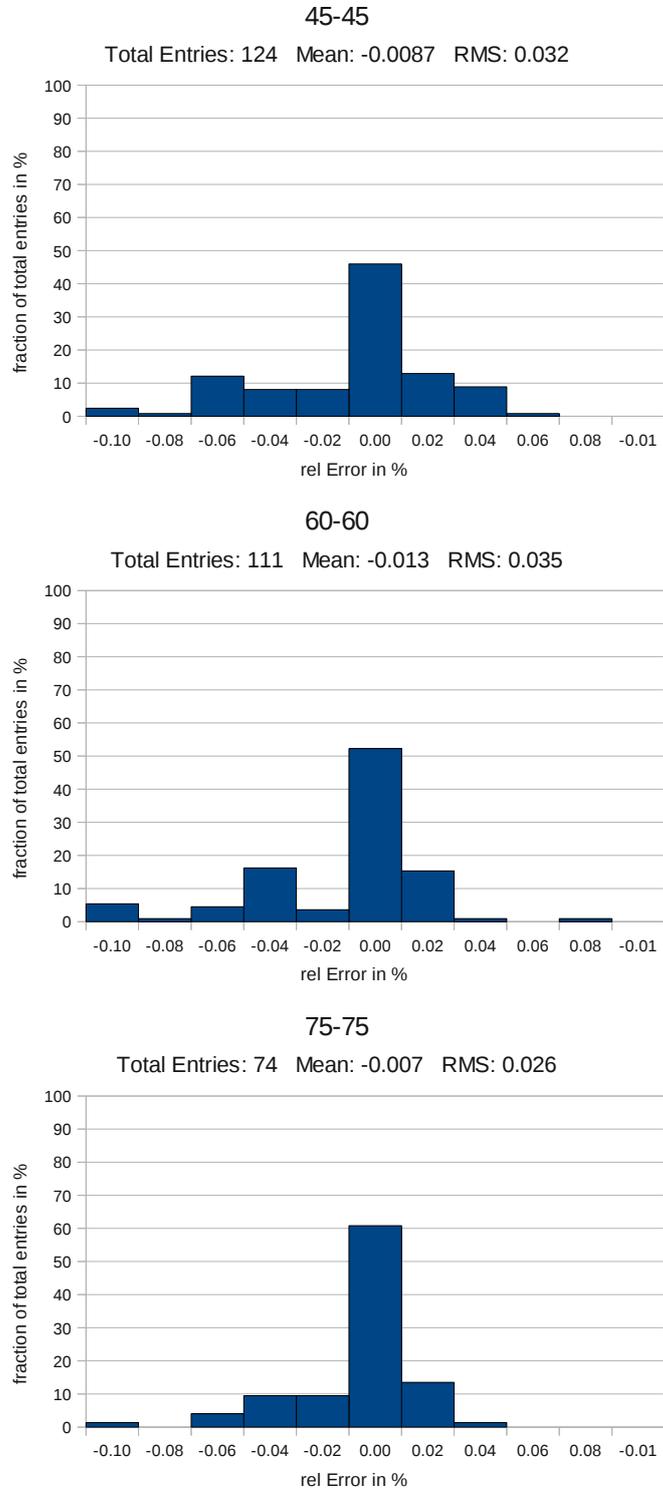


Figure 5.5: Relative deviation between the results of the IM and the SSM



In the following table the statistical quantities for the above histograms are summarized. The RMS of the relative deviation between the IM and SSM increases for low angles but decreases for angles above  $45^\circ$ .

Table 5.5: Relative deviation

angle range	rel Deviation IM to SSM (in %)			rel Deviation PM to SSM (in %)		
	Total	Mean	RMS	Total	Mean	RMS
00-75	125	-0.0056	0.0223	123	0.15	28.47
00-00	125	0.0000	0.0000	125	0.00	0.00
15-15	125	-0.0010	0.0083	124	0.29	1.26
30-30	128	-0.0036	0.0184	125	1.50	4.46
45-45	124	-0.0087	0.0315	118	3.28	12.02
60-60	111	-0.0130	0.0346	110	6.88	21.04
75-75	74	-0.0070	0.0261	71	12.28	47.18

In Table 5.5 some information regarding the PM is shown as well. It has been compared to the SSM.

To get a rough idea whether the above deviations are acceptable one can have a look at how well the SM can reproduce actual experimental data. In [5, p. 150] this analysis is performed. There, one aims for a relative deviation of less than 10 percent.

## 5.2 Speed

In order to test the speed of the different implementations a profiling tool named Callgrind has been used. Callgrind records the amount of instructions executed in every function of the simulation while executing it. Nine runs have been performed with Callgrind using once 10, 1 000 and 100 000 events with each the SM, IM and PM implementation in the simulation.

To get an estimate about the speed difference of the implementations one has to separate the optimized code from the non-optimized overhead of the simulation software. One way to accomplish this is to look for a function that preferably contains all the changes of the code alteration but nothing else. *Exec*<sup>2</sup> is one function that contains all the changes but has quite some non-optimized overhead. The table below lists the amount of instructions that have been executed within the function *Exec* during the corresponding run:

---

<sup>2</sup>Like all the functions quoted in this thesis this one is part of the class `CbmMvdDigitizeL`. The full prototype is `CbmMvdDigitizeL::Exec(Option_t*)`

Figure 5.6: Amount of Instructions used / 1,000,000

Function	Exec			PIP+PPC / CHC+CPA		
EventCount	10	1000	100000	10	1000	100000
SM	3.1	293.6	29142.7	2.7	260.6	25966.3
IM	1.0	95.5	9396.9	0.6	57.2	5687.4
PM	0.7	63.8	6326.2	0.3	30.4	3003.2
ratio SM/IM	3.055	3.073	3.101	4.606	4.558	4.566
ratio SM/PM	4.304	4.600	4.607	8.037	8.583	8.646

The table shows that due to the alterations the function *Exec* has become faster by a factor of about three in the IM. Using the PM for all the tracks which means accepting high errors for highly inclined tracks gives a speed-up of about 4.5. When ascending even deeper into the call tree one finds that the functions *ProduceIonisationPoints* and *ProducePixelCharge* (PIP+PPC) from the SM code taken together do about the same thing as the functions *CalculateHitCoordinates* and *CalculatePixelActivation* (CHC+CPA) from the IM and PM code. Looking at those functions one gets less non-optimized overhead. The approximate amount of instructions used for both these functions in each implementation respectively is also found in the table. There the speed-up is a factor of about 4.5 for the IM because less non-optimized code is taken into account. For the PM the speed-up is about 8.

### 5.3 Conclusion

A new implementation of the response simulation was developed and tested. It turned out to replicate the information obtained by a more accurate version of the former implementation with an error of less than 1 per mil for all tracks with an inclination between  $0^\circ$  and  $75^\circ$ . At the same time the execution of the new implementation took only about a third of the time that was needed before.

It remains to be tested how the new implementation influences the hit reconstruction.

# Bibliography

- [1] Photo of MIMOSA26.
- [2] M. Deveaux. CMOS monolithic active pixel sensors. Technical report, Goethe University Frankfurt.
- [3] M. Deveaux. *Development of fast and radiation hard Monolithic Active Pixel Sensors (MAPS) optimized for open charm meson detection with the CBM - vertex detector*. PhD thesis, Johann Wolfgang Goethe - Universitaet in Frankfurt am Main, 2007.
- [4] M. Deveaux. Status des CBM experimentes. Technical report, 2011.
- [5] C. Dritsa. *Conception of the Micro Vertex Detector of the CBM experiment: Development of a detector response model and feasibility studies of open charm measurement*. PhD thesis, Johann Wolfgang Goethe - Universitaet in Frankfurt am Main, 2011.
- [6] B. Friman, C. Höhne, J. Knoll, S. Leupold, J. Randrup, R. Rapp, and P. Senger. *The CBM Physics Book*. Springer, 2011.