Development of a Fast Search Algorithm for the MUSiC Framework

von
Jonas Lieb

Bachelorarbeit im Fach Physik

vorgelegt der
Fakultät für Mathematik, Informatik und Naturwissenschaften
der RWTH Aachen

im
September 2015

angefertigt am
III. Physikalischen Institut A

bei
Prof. Dr. Thomas Hebbeker

Ich versichere, dass ich die Arbeit einschließlich beigefügter Darstellungen und Tabellen selbständig angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Jonas Lieb
Aachen, den 9. September 2015

Abstract

The CMS experiment at the LHC produces a vast amount of data: Each second about 20 TB of information is generated by the detector hardware. Accordingly, analysis of the data also requires a lot of computing power. Using a chain of several algorithms, the detector signal is interpreted as physical meaningful data, on which state-of-the-art analyses are performed.

The Model Unspecific Search in CMS (MUSiC) is an analysis carried out on a wide spectrum of final states. Kinematic distributions of these final states are aggregated and compared to the expectation from Standard Model Monte Carlo simulations. By searching for deviations, MUSiC is sensitive to indications of physics beyond the standard model.

This thesis proposes, implements and validates an additional step in the MUSiC analysis, which drastically reduces the runtime.

Kurzdarstellung

Das CMS Experiment am LHC produziert sehr große Datenmengen: Pro Sekunde werden rund 20 TB an Daten von der Detektorhardware generiert. Folglich erfordert auch die Auswertung viel Rechenleistung. Mithilfe mehrerer Algorithmen wird dem Detektorsignal eine physikalische Bedeutung zugewiesen, aufgrund derer modernste Analysen durchgeführt werden.

Die modellunspezifische Suche MUSiC (engl: Model Unspecific Search in CMS) ist eine Analyse, die auf einem breiten Spektrum von Endzuständen arbeitet. Dabei werden kinematische Verteilungen der Endzustände erstellt und mit Erwartungen aus Monte-Carlo-Simulationen des Standardmodelles verglichen. Die Suche nach Abweichungen macht MUSiC sensitiv auf Anzeichen von Physik jenseits des Standardmodelles.

Diese Arbeit erklärt, implementiert und validiert einen zusätzlichen Arbeitsschritt der MUSiC-Analyse, der die Rechenzeit drastisch reduziert.

Contents

1 Introduction

1.1 Units

Throughout this work, a natural unit system will be used, as it is convention in high energy particle physics. The speed of light and the reduced Planck constant are fixed to $c = 1$ and $\hbar = 1$, and as such they are omitted in equations. Additionally, energy is expressed in GeV, where 1 GeV is the energy gain of an electron which is accelerated across a 1 GV potential ($1\,\text{eV} \approx 1.602 \times 10^{-19}\,\text{J}$). These conventions induce a change in units for the other dimensions, most importantly mass and momentum, both of which are notated in GeV. Benefits of this choice of units are simpler equations, the possibility of a direct comparison between energy and masses, and typical quantities of $\mathcal{O}(1\,\text{GeV})$.

1.2 The Standard Model

The *standard model* (SM) is a theory that represents our current knowledge of elementary particles, their forces and interactions, excluding gravity.

Particles in the standard model can be classified into two separate classes: *Fermions*, which make up matter and possess a spin of $1/2$, and *bosons*, which are mediators of forces and possess an integer spin. Three elementary forces arise, each from their separate theory: *electrodynamic* (described by Quantum Electrodynamics, QED), *strong* (described by Quantum Chromo Dynamics, QCD) and *weak* (described by Quantum Flavor Dynamics). The forces are induced by corresponding charge-like properties: electrodynamic charge, color charge and weak isospin. These charges can then be used to further subdivide fermions into *quarks* and *leptons*, which will be described in the following sections.

The standard model also provides theoretical predictions about the relations between particles, their masses and the probabilities of certain processes to happen. Because the probability is proportional to the number of times that a process occurs, experimentalists can validate the standard model by counting events with certain outcomes.

1.2.1 Leptons

There are three charged leptons: the *electron* (e), the *muon* (μ) and the *tau* (τ). They carry the electric charge of $-1\,e$ and participate in electrodynamic and weak interactions. For each charged lepton, there is one *neutrino* counterpart (ν_e, ν_μ, ν_τ). Neutrinos are electrically neutral, weakly interacting, massless particles. They remain undetected in current collider detectors. The leptons do not carry color charge and are thus excluded from the strong interaction. An overview about the leptons and their masses can be found in table 1.1.

	electron (e)	muon (μ)	tau (τ)
mass	511.0 keV	105.7 MeV	1.777 GeV
charge	-1	-1	-1
	e neutrino (ν_e)	μ neutrino (ν_μ)	τ neutrino (ν_τ)
mass	< 2 eV	< 0.19 MeV	< 18.2 MeV
charge	0	0	0

Table 1.1: Leptons in the standard model [1, p. 30, p. 690f.].

1.2.2 Quarks

Similarly to the leptons, quarks can be divided into three generations. The first generation contains the stable *up* and *down* quarks, the second generation the *charm* and *strange* quarks and the third generation contains the heavy *top* and *bottom* quarks. Quarks carry a fractional electric charge of either $^2/_3\,e$ or $-^1/_3\,e$. They also carry color charges and take part in the strong interaction as well as in the weak interaction. The three quark generations and their properties are shown in table 1.2.

	up (u)	charm (c)	top (t)
mass	2.3 MeV	1.28 GeV	173.2 GeV
charge	$^2/_3$	$^2/_3$	$^2/_3$
	down (d)	strange (s)	bottom (b)
mass	4.8 MeV	95 MeV	4 GeV
charge	$^{-1}/_3$	$^{-1}/_3$	$^{-1}/_3$

Table 1.2: Quarks in the standard model [1, p. 33].

1.2.3 Bosons

Gauge bosons are mediators of the elementary forces. The electromagnetic force is mediated by the massless *photon* (γ), the strong force by the massless *gluon* (g) and the weak force by the massive Z and $W^{\pm}$ bosons. They couple to the corresponding charges. To account for the massive boson masses, the Higgs mechanism is introduced. The Higgs field gives rise to the Higgs boson (H), for which a candidate has been found in 2012 [2]. All bosons and their properties are listed in table 1.3.

	Photon (γ)	Gluon (g)	Z-Boson (Z)	W-Bosons ($W^{\pm}$)	Higgs (H)
mass	0	0	91.2 GeV	80.4 GeV	126 GeV
charge	0	0	0	± 1	0
interact.	el.-mag.	strong	weak	weak	Higgs

Table 1.3: Bosons in the standard model [1, p. 27].

1.2.4 Antiparticles

For each SM particle, there exists one counterpart with the same mass but an opposite sign for all charge-like properties, called *antiparticle*. Unlike fermions, bosons are their own antiparticles, called Majorana particles.

1.2.5 Ranges

While the electrodynamic and weak couplings decrease with increasing distance, the strong force only grows stronger with a larger distance. This gives rise to the phenomenon of *quark confinement*: as two quarks are separated from each other, new quark-antiquark pairs are created from the energy in between. Because of this, outgoing quarks with high energies form *jets* consisting of hadrons from newly created quarks.

1.3 The Large Hadron Collider

Elementary particles and the standard model are commonly studied using scattering experiments. Such an experiment can be separated into two basic parts: an *accelerator* and a *detector*. Inside the accelerator, the particles are accelerated to high energies using an electric field. They collide with their target inside the detector, which records various properties of the deflected or created particles, such as di-

rection, momentum and energy. One can either accelerate particles on a straight line (*linear accelerator*) or on a circular trajectory (*ring*) which is enforced by dipole magnets. The latter option allows for longer acceleration time but is limited by the emission of Bremsstrahlung. There are also two options for the collision. Either a *fixed target* can be irradiated or two accelerated particle beams can be brought to collision (*collider*). Because of kinematic reasons, a collider experiment can achieve higher energies.

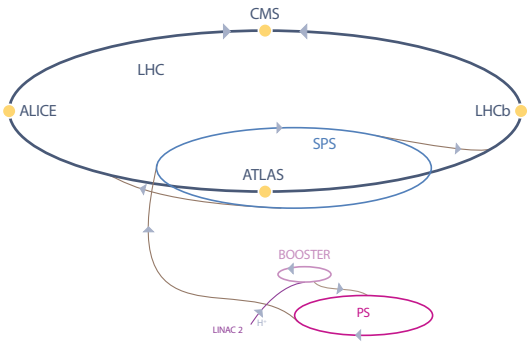

Figure 1.1: The CERN accelerator complex [3, modified]. Shown are the LHC ring, the preaccelerators (Linear Accelerator 2, BOOSTER, Proton Synchrotron, Super Proton Synchrotron) and the four main LHC experiments: CMS, ATLAS, ALICE and LHCb.

The *Large Hadron Collider* (LHC) is a proton-proton ring collider experiment of the European Organization for Nuclear Research (*Conseil Européen pour la Recherche Nucléaire*, CERN), situated between 45 m and 170 m underground near Geneva, Switzerland. The LHC has been installed in the tunnel of the former Large Electron Proton Collider (LEP), having a circumference of 26.7 km [4, 5]. The two proton beams are injected into the LHC after passing a series of preaccelerators, as shown in figure 1.1. It has been designed to provide an energy of 7 TeV per beam, resulting in a center-of-mass energy $\sqrt{s} = 14\,\text{TeV}$. It is currently (2015) running at $\sqrt{s} = 13\,\text{TeV}$, the data analyzed in this work were taken in 2012 at $\sqrt{s} = 8\,\text{TeV}$ with a total integrated luminosity of $19.7\,\text{fb}^{-1}$ [6].

1.4 The Compact Muon Solenoid Detector

The Compact Muon Solenoid Detector (CMS) is an experiment at the LHC. The CMS detector [7] consist of a barrel section along the beam pipe, which is closed off by two end caps. Collisions happen in the center of the barrel at the interaction point. The detector parts and particle footprints can be found in figure 1.2. Close to the *interaction point*, the inner part of the barrel contains a *silicon tracker* used to record trajectories of charged particles. The inner tracker is surrounded by

an *electromagnetic calorimeter* (ECAL) made of lead tungstate crystals. The ECAL measures energy deposited by electrons and photons, combined with the tracker it can distinguish between the two particle types. After the electromagnetic calorimeter follows the *hadron calorimeter* (HCAL). It is a sampling calorimeter composed of layers of brass absorber plates and plastic scintillator fibers. Due to the high density of the brass absorbers, hadronic showers deposit most of their energy inside the HCAL. The HCAL is surrounded by the most important feature of the CMS detector, the *solenoid magnet*. The solenoid coil is cooled down below 10 K, where the NbTi (Niobium-Titanium) conductor becomes superconducting, allowing for currents up to 19 kA. The resulting field strength in the center of the coil is 4 T. The strong magnetic field causes the charged particle tracks to bend, this effect is used to measure the momentum of charged particles. In the outer part one finds the iron return yoke interspersed with *muon drift chambers*, *cathode strip chambers* and *resistive plate chambers*. Since only muons and neutrinos pass through the solenoid and iron yoke and neutrinos remain undetected, the drift chambers can precisely identify muons and assess their momentum.

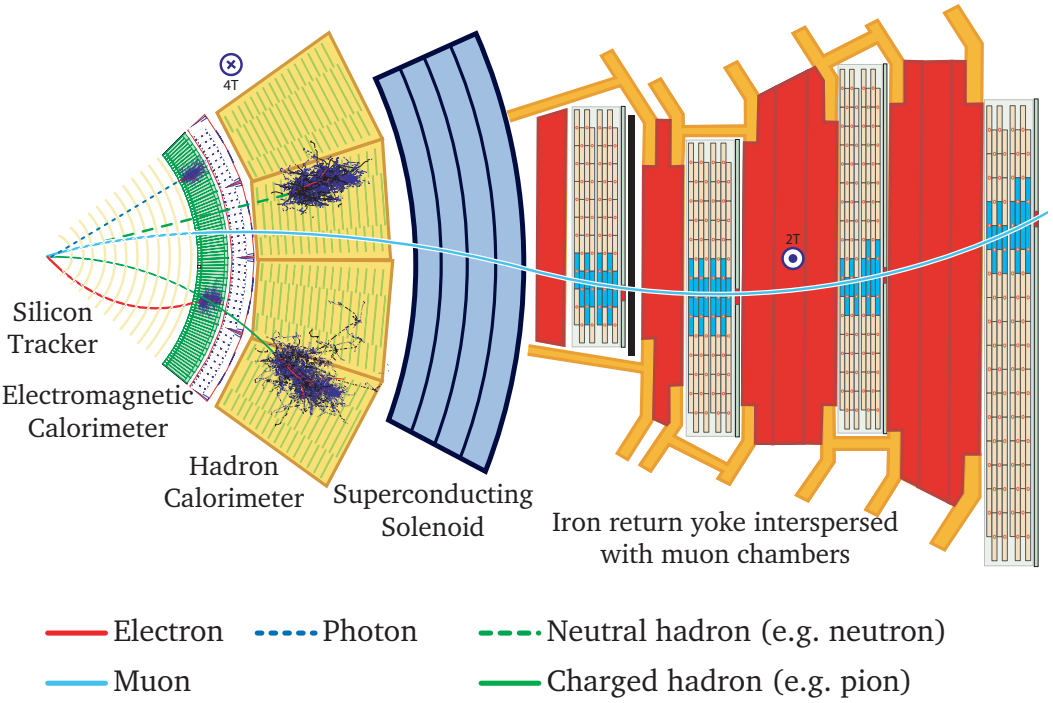

Figure 1.2: Slice through the CMS detector. The different detector layers (tracker, ECAL, HCAL, magnet, muon chambers) are shown, alongside schematic tracks of an electron, a photon, a charged and neutral hadron and a muon. [8, modified]

1.4.1 Coordinate System

The CMS collaboration has defined a spherical coordinate system inside the detector, as illustrated in figure 1.3. The origin is located in the interaction point, the z-axis points along the beam pipe towards the Jura mountains. The x-axis is defined towards the center of the LHC ring. To complete the right-handed orthogonal coordinate system, the y-axis points upwards, orthogonally from the x-z-plane. An azimuthal angle ϕ is defined from the x-axis in the x-y plane. The polar angle θ is measured from the z-axis. One can also define a radial coordinate r in the x-y plane, as distance from the z-axis. [7, p. 2]

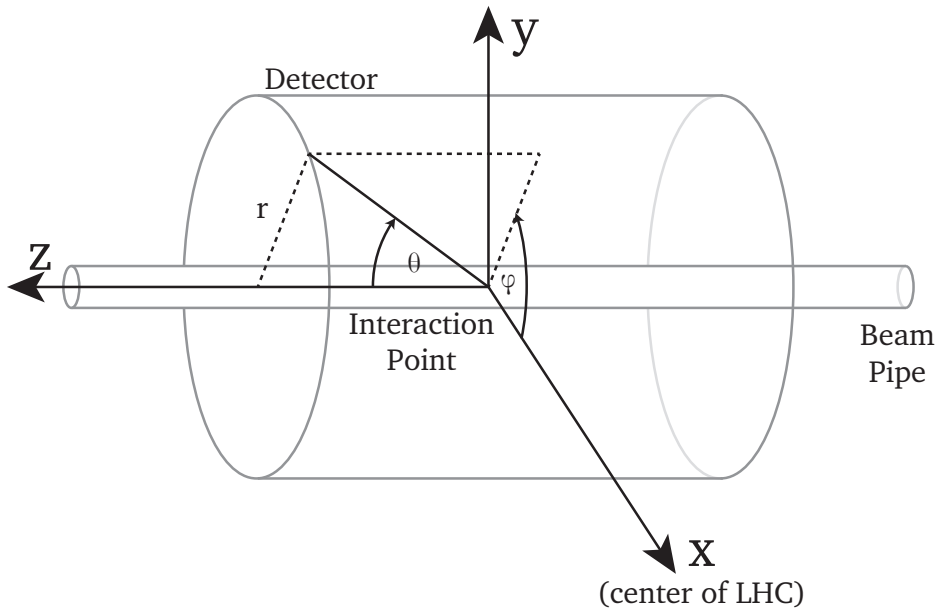

Figure 1.3: The CMS coordinate system.

1.5 Commonly Used Quantities

The proton is a composite particle consisting of quarks and gluons, which each carry an unknown momentum fraction. Because a collision can only occur between these constituents, the initial momentum sum along the z-axis is unknown. Thus, most of the kinematic variables are defined in the transverse (x-y) plane. A particle is assigned a *transverse momentum* $\vec{p}_{\mathrm{T}}$ which only contains the transverse component of its momentum vector. Analogous, only a fraction of the energy is regarded, the *transverse energy* E_{T}. The sum of all transverse momenta in a final state is usually not equals to zero since invisible neutrinos carry away some of the momentum.

This is accounted for by the negative sum of the transverse energy, which is called *missing transverse energy* $\not{E}_{\mathrm{T}}$. Another commonly used quantity is the pseudo-rapidity $\eta = -\ln\tan(\theta/2)$, which has the property that distances $\Delta\eta$ are Lorentz invariant.

1.6 Bunch Crossings, Events, Collisions and Pileup

For acceleration purposes, the protons circulating in the LHC beam pipe are grouped into 2808 *bunches* [5, p. 4]. A *bunch crossing* happens every 50 ns (2012), defining an *event*. During an event, multiple pp-collisions can occur. The collision containing the vertex with the largest transverse momentum is called *hard interaction*. All other collisions that do not correspond to the physics process of the hard interaction are classified as *pileup*.

1.7 Trigger

Since the raw data for each event is about $\mathcal{O}(1\,\mathrm{MB})$, the throughput required to transfer all data would be $\mathcal{O}(20\,\mathrm{TB})$ per second. This amount of data is too much for current hard- and software, thus a selection mechanism is introduced to match the amount of recorded data to the networking, processing and storage capabilities. This system is called *trigger*. The CMS trigger system consists of two layers, the "Level-1 Trigger" (L1), implemented in hardware on site, and the software "High-Level-Trigger" (HLT), located at a computing farm close to the detector. The triggers process the limited input from the detector and decide whether an event is worth storing on disk, based on physics arguments. These requirements reduce the data amount by $\mathcal{O}\left(10^6\right)$.

1.8 Event Reconstruction

Various *reconstruction* (RECO) algorithms are executed offline to reconstruct physics objects from the recorded data. The most important inputs are the bending radius of the particle tracks, energy deposits in the calorimeters and hits in the muon chambers. A notable algorithm that ensures that each input is linked to exactly one physics object has been developed by CMS and is called *Particle-Flow* [9].

2 Model Unspecific Search in CMS

The Model Unspecific Search in CMS (*MUSiC*) [10–12] is an analysis procedure that compares observed data to the standard model expectation from Monte Carlo (MC) simulations. Unlike dedicated analyses that usually only regard a few final states, an unspecific search covers a broad spectrum of final states. This is especially useful since some theories predict small deviations in many final states, which on their own would not be classified as significant, but are relevant in sum.

This chapter will focus on the MUSiC workflow. First, the motivation behind an unspecific search will be illustrated, then the three analysis steps *skimming*, *classification* and *scanning* will be explained.

2.1 Motivation

Since a Higgs boson candidate has been discovered at the LHC [2], all particles predicted by the standard model have been observed. Nevertheless there are many unsolved problems in modern particle physics. The most noticeable examples are the Higgs mass hierarchy problem, the matter/antimatter asymmetry, dark matter and energy and the possibility of unification theories. Various theories, such as supersymmetric extensions (SUSY), propose solutions, but they are currently lacking evidence. Expected signatures of these models could show up in a large range of final states. For some models, narrow resonances are predicted, others result in small deviations in the high-energy tail of the distributions. The MUSiC analysis is sensitive to these kinds of deviations. Additionally, MUSiC can identify deviations between MC and data that originate from non-physics sources, uncovering weaknesses in the Monte Carlo simulation.

2.2 Data Preprocessing

The goal of preprocessing is to gather events from different data sources and convert them to a unified format containing only information relevant to MUSiC. AOD (analysis object data) files of observed as well as simulated events are stored on non-local parts of the computing grid, and preprocessed there. The stripped results are contained in PXL I/O files [13] and returned to the local computing grid.

2.3 Object Identification

During the object identification step, criteria are applied to the objects identified by Particle-Flow during the reconstruction. A summary of the identification requirements, which are developed by dedicated groups within the CMS collaboration, is shown in table 2.1. More detailed information can be found in [6].

Object	$\vec{p}_{\mathrm{T}}$/ GeV	$\lvert\eta\rvert$	Identification Summary
μ	>25	<2.1	track quality, isolation, dedicated high-$\vec{p}_{\mathrm{T}}$
e	>25	<2.5	track quality, isolation, dedicated high-E_{T}
γ	>25	<1.442	isolation, veto against e from conversions
jet	>50	<2.4	anti-$\mathrm{k_t}$ algorithm ($R = 0.5$)
$\not\!\!E_{\mathrm{T}}$	>50		

Table 2.1: Identification criteria for MUSiC physics objects [6].

2.4 Classification

During the classification step, events are grouped into *event classes* (EC) according to their physics object content in the final state, with the definitions from the previous section.

There are three types of event classes: *exclusive*, *inclusive* and *jet inclusive*. Final states of events in exclusive event classes contain exactly the objects indicated in the event class name. Events in the exclusive EC $\boxed{1\mathrm{e} + 1\mathrm{MET}}$ contain only 1 electron and missing transverse energy in the final state, but no other physics objects. Events in the inclusive EC can contain any other objects besides the indicated ones. $\boxed{1\mathrm{e} + 1\mathrm{MET} + \mathrm{X}}$ contains at least 1 electron and missing transverse energy. Additionally, there are *jet inclusive* event classes (e.g. $\boxed{1\mathrm{e} + 1\mathrm{MET} + \mathrm{Njet}}$), that exclusively contain the mentioned objects and zero or additional jets, making the analysis more robust to initial and final state radiation caused by the emission of gluons in the initial or final state.

These definitions imply that each event is contained in exactly one exclusive EC, at least one inclusive EC and at least one jet inclusive EC.

The classification algorithm automatically adjusts the possible event classes to the events actually present in the dataset, dynamically creating new classes.

If the dataset originates from MC simulations, some properties of the simulated processes are shifted within their uncertainties to estimate their impact on the classes. Additionally, the amount of simulated events is rescaled to match the data luminosity.

2.4.1 Kinematic distributions

Three kinematic variables are analyzed: the scalar sum of the magnitudes of all observed momenta $\sum |\vec{p}_{\mathrm{T}}|$, the total invariant mass $M_{\mathrm{inv}} = |\sum \boldsymbol{p}|$ and the missing transverse energy $\not{E}_{\mathrm{T}}$. For each event class and each kinematic variable, one histogram is filled. The histograms are created with variable bin sizes according to the detector resolution [11, p. 52]. Note that the vertical axis of histograms shown in this work carries the unit "Counts per 10 GeV", so to get the absolute number of events in a bin, the vertical data point has to be multiplied by the width of its bin in units of 10 GeV.

2.5 Scanning

The scanning algorithm (*scanner*) searches for the connected bin region with the most significant deviation between the data and MC yield in each histogram.

2.5.1 Region Building

For each histogram, the scanner probes all connected bin regions and calculates a p value for each one.

The set of connected bin regions can be defined and calculated as

$$R = \{(s, e)\, \forall e \in (s + m, l)\, \forall s \in (0, l - m)\} \tag{2.1}$$

where the histogram spans the bins 0 to l and m denotes a minimal number of bins per region which can be configured.

For each region, a p value is calculated. The region of most deviation (smallest p) is called *region of interest*.

2.5.2 The p value

Given the number of total observed events in a region N_{obs} and an expectation $N_{\mathrm{SM}} \pm \sigma_{\mathrm{SM}}$, the p value expresses the probability of obtaining a deviation at least as

significant as the observed one, given only the standard model expectation. Since the calculation deals with results of counting experiments, a Poisson distribution is assumed. The probability of making an observation N given the Poisson mean N_{SM} is

$$P(N) = \frac{e^{-N_{\mathrm{SM}}} {N_{\mathrm{SM}}}^N}{N!} \tag{2.2}$$

To include observations that are more extreme than the observation, the probabilities are summed away from the expectation:

$$p = \begin{cases} \sum_{N=N_{\mathrm{obs}}}^{\infty} \frac{e^{-N_{\mathrm{SM}}} {N_{\mathrm{SM}}}^N}{N!} & \text{if } N_{\mathrm{obs}} \geq N_{\mathrm{SM}} \\ \sum_{N=0}^{N_{\mathrm{obs}}} \frac{e^{-N_{\mathrm{SM}}} {N_{\mathrm{SM}}}^N}{N!} & \text{if } N_{\mathrm{obs}} < N_{\mathrm{SM}} \end{cases} \tag{2.3}$$

Since the Poisson mean is usually not exactly known, due to insufficient simulation and measurement uncertainties (e.g. luminosity), it is evaluated for multiple possible values θ which are weighted by a Gaussian distribution of width σ_{SM} around N_{SM}. This final probability is called p *value*:

$$p := \begin{cases} \sum_{N=N_{\mathrm{obs}}}^{\infty} C \cdot \int_0^{\infty} \mathrm{d}\theta \exp\left(-\frac{(\theta - N_{\mathrm{SM}})^2}{2{\sigma_{\mathrm{SM}}}^2}\right) \cdot \frac{e^{-\theta}\theta^N}{N!} & \text{if } N_{\mathrm{obs}} \geq N_{\mathrm{SM}} \\ \sum_{N=0}^{N_{\mathrm{obs}}} C \cdot \int_0^{\infty} \mathrm{d}\theta \exp\left(-\frac{(\theta - N_{\mathrm{SM}})^2}{2{\sigma_{\mathrm{SM}}}^2}\right) \cdot \frac{e^{-\theta}\theta^N}{N!} & \text{if } N_{\mathrm{obs}} < N_{\mathrm{SM}} \end{cases} \tag{2.4}$$

The constant C denotes a normalization factor that compensates for using 0 as the lower limit of the integral. This is necessary since the Poisson distribution with a negative expectation is not physical.

2.5.3 The $\tilde{p}$ value

For a dedicated analysis looking at one fixed histogram region only, this p value would be sufficient to describe the statistical significance of the deviation. But since the scanning algorithm regards many regions to choose the most significant one from, the probability of finding a deviation somewhere in the distribution is larger than p. This is called *look elsewhere effect*. Since MUSiC is actually interested in the global significance of the distributions deviation, the look elsewhere effect has to be corrected. Therefore the scanning algorithm is applied multiple (10^5) times on randomly diced pseudo-data. For each pseudo experiment, first the expected mean ${N_{\mathrm{SM}}}'$ is randomly shifted from a Gaussian distribution around N_{SM}, having the width of the systematic error σ_{SM}. Taking this new pseudo-mean, a random Poisson number is chosen as new observed value N_{pseudo} which takes the place of N_{obs}. The dicing of pseudo experiments is performed in a correlated way, the chosen

deviation in units of σ_{SM} is shared between all distribution in all event classes. The complete scanning algorithm is repeated for each pseudo-experiment, finding a region of interest and its p value. Finally, the amount of pseudo experiments with

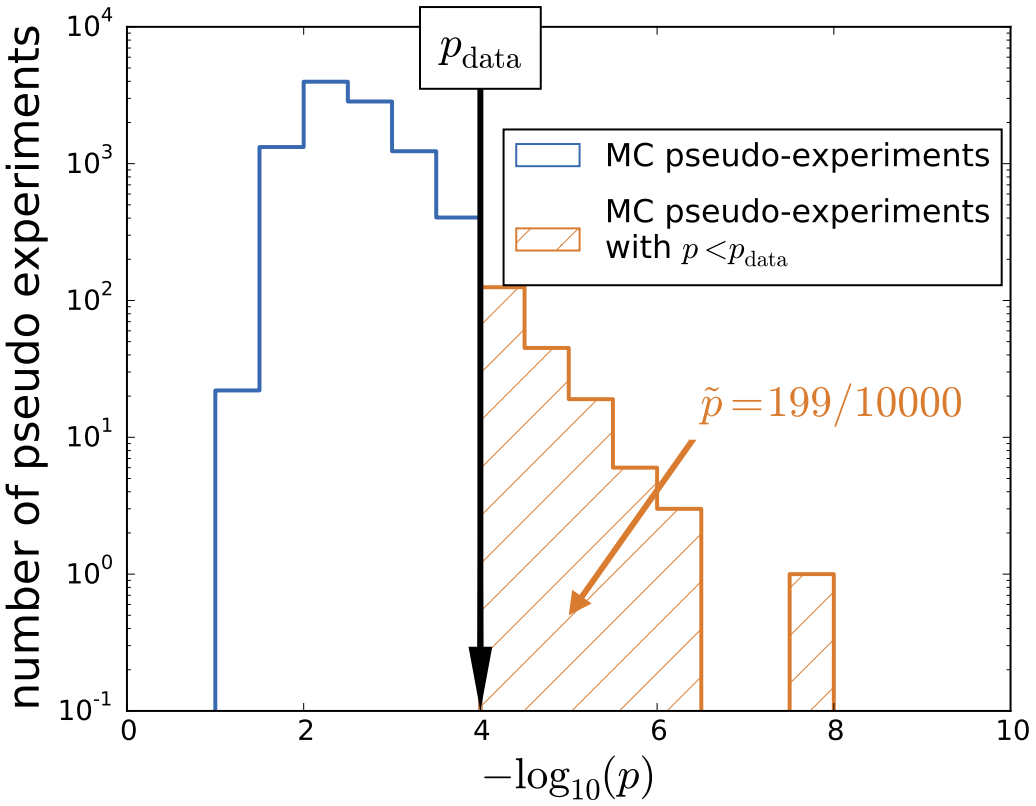

Figure 2.1: Illustration of a $\tilde{p}$ calculation. 10 000 pseudo-experiments have been conducted. The 199 experiments on the right of p_{data} have shown a more extreme deviation somewhere in the distribution, resulting in $\tilde{p} \approx 0.02$.

more significant deviations than the observed one p_{data} is compared to the total amount of pseudo experiments conducted:

$$\tilde{p} := \frac{\text{number of pseudo-experiments with } p < p_{\mathrm{data}}}{\text{total number of pseudo experiments}} \tag{2.5}$$

This calculation is illustrated in figure 2.1.

3 Motivation for a Fast Search Algorithm

The calculation of the p value includes integration over a series. The p value is calculated for each connected bin region, resulting in a runtime of $\mathcal{O}\left(n^2\right)$ with the number of bins in a distribution. Additionally, the runtime increases linearly with the number of classes, distributions and pseudo-experiments.

The typical number of integrals computed during a full scan of 2012 data, one kinematic variable and 10^5 pseudo-experiments is $\mathcal{O}\left(10^9 - 10^{10}\right)$. The computation of each single integral takes about 200 µs (see table A.1 in the appendix), which results in a total computation time of about 2 to 20 CPU-days. The scan is automatically parallelized between worker processes and usually performed on a 64 core machine, thus taking 1 to 9 hours.

The calculation of the integral is performed using simple adaptive integration (QAG, from the GNU Scientific Library). Since this implementation is already highly optimized, further improvement of the integral calculation is difficult.

Since the ultimate quantity of interest is the $\tilde{p}$ value, the goal of this works fast search algorithm, called *Quickscan*, is to reduce the time spent on scanning pseudo-experiments. This will be achieved by reducing the amount of candidate regions, for which the p value integral is calculated.

The new algorithm shall not reduce the amount of regions too far, such that the "true" region of interest is not included anymore, since this would highly impact statistical accuracy. Thus, the deviation of $\tilde{p}$ with and without Quickscan will be observed in the remainder of this work.

Shortening the total scanning time will eventually enable the MUSiC project to increase the number of pseudo experiments and remove other requirements that have been made for optimization, but have a higher impact on statistical accuracy.

4 A Fast Search Algorithm

In this chapter, the basic concept and method of the Quickscan algorithm is summarized. Arising problems are described and their solutions within the algorithm are proposed. Finally, other approaches are mentioned, which have not been implemented in this work's version of Quickscan.

4.1 Concept

The goal of the Quickscan algorithm is to make the scanning process of pseudo-experiments more efficient. The efficiency improvement is achieved by preselecting interesting regions as RoI (region of interest) candidates and discard most other regions before the costly p value calculation. The selection is performed by evaluating a less computation intensive estimator over all possible connected regions, and making a selection of the most significant deviations according to this estimator. Finally, the full p value and its integral are only calculated for the few candidate regions returned by the estimator.

4.1.1 The Estimator

The choice of the estimator plays an important role in the efficiency of the selection. A commonly used quantity in particle physics is

$$\chi = \frac{|N_{\mathrm{obs}} - N_{\mathrm{SM}}|}{{\sigma_{\mathrm{SM}}}'} \tag{4.1}$$

Here the value ${\sigma_{\mathrm{SM}}}'$ describes the expected deviation. This is not σ_{SM} which only includes systematic uncertainties, e.g. finite Monte-Carlo statistics, and has been rescaled to match the data luminosity. ${\sigma_{\mathrm{SM}}}'$ consists of the expected statistical deviation $\sqrt{N_{\mathrm{SM}}}$ and the systematical error σ_{SM}. The two errors are added in quadrature. Because $N_{\mathrm{SM}} \geq 0$, the expression can be simplified:

$${\sigma_{\mathrm{SM}}}' := \sqrt{{\sigma_{\mathrm{SM}}}^2 + \left(\sqrt{N_{\mathrm{SM}}}\right)^2} = \sqrt{{\sigma_{\mathrm{SM}}}^2 + N_{\mathrm{SM}}} \tag{4.2}$$

This value incorporated into the χ-value gives

$$\chi = \frac{|N_{\text{obs}} - N_{\text{SM}}|}{\sqrt{{\sigma_{\text{SM}}}^2 + N_{\text{SM}}}} \tag{4.3}$$

4.1.2 Selection

In each distribution, the χ value is calculated for each connected bin region. The most significant regions are stored in a list. This introduces a parameter of the Quickscan algorithm, n_{regions}, which denotes the *number of candidates* kept in the list.

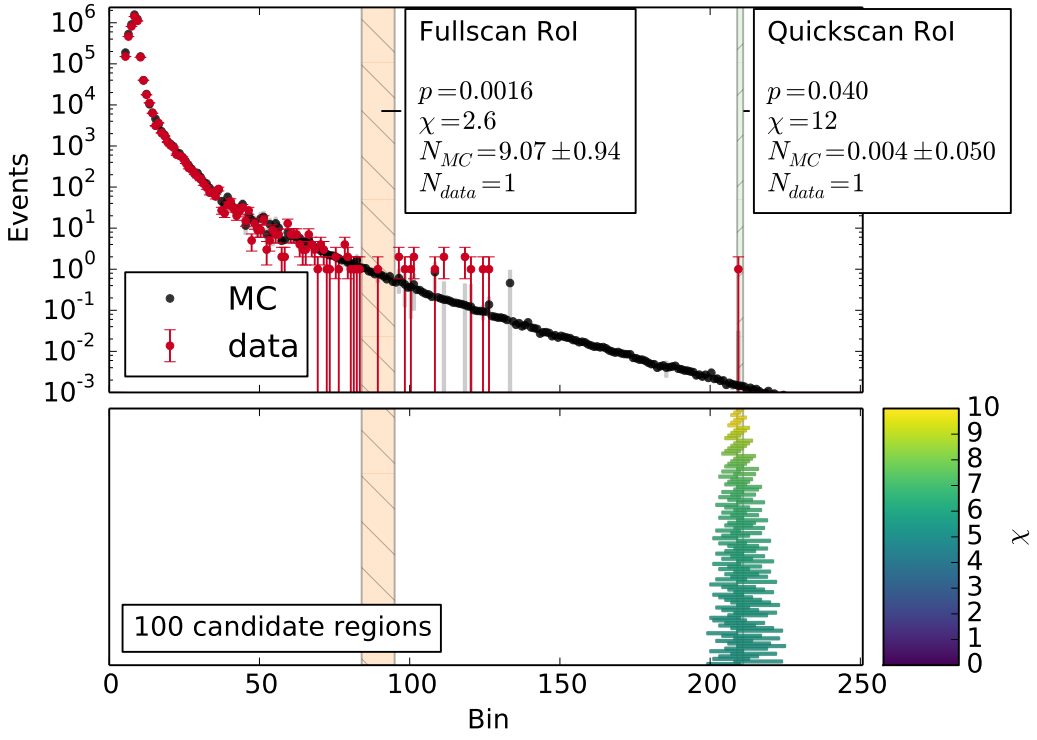

Figure 4.1: Example distribution: Pseudo experiment of the 2e $\sum |\vec{p}_{\text{T}}|$ distribution, without special treatment of nested regions. The upper panel shows the distribution with the bin number (not the physical quantity) on the horizontal axis and the number of events in each bin on the vertical axis. The lower panel shows the 100 selected candidate regions, sorted by χ. The regions chosen by the different approaches are indicated in the upper panel. p denotes the full scan p value, while χ indicates value of the Quickscan estimator. One can see that especially in the regions where the Poissonian approach $\sigma_N = \sqrt{N}$ fails, the χ value is too sensitive. The nested regions suppress different regions in the candidate list, such that the "true" RoI (as found without the Quickscan) is not detected.

4.1.3 Nested Region Handling

Using only the estimator and the selection of top n_{regions} candidates, the algorithm tends to focus around single data points in the tail of the distribution, where the Poissonian approximation $\sigma_N = \sqrt{N}$ does not hold due to a low event count. This effect is illustrated in figure 4.1. The pseudo experiment of the 2e $\sum |\vec{p}_{\text{T}}|$ distribution shows a single event above a low background expectation. The regions with the highest χ gather around this excess. Comparison with a scan without the new algorithm shows that the "true" region of interest is found further left in the distribution, as a deficit with only 1 observed event in 9 expected.

To suppress this behavior, an additional list of criteria is introduced, comparing region A with region B:

- A is nested inside of B
- Excess of data in A: $N_{\text{obs}}(A) > N_{\text{SM}}(A)$
- Excess of data in B: $N_{\text{obs}}(B) > N_{\text{SM}}(B)$
- No additional data in $B \setminus A$: $N_{\text{obs}}(A) = N_{\text{obs}}(B)$

If all these conditions are met, then region A is more significant than region B. The mathematical proof for this criterion is difficult, but it can be motivated as follows: If a region contains an excess of observed event yield over MC event yield, and the region is extended while the amount of observed events remains the same, the difference between N_{SM} and N_{obs} stays equal or is reduced. Since the error σ_{SM} can only stay the same or increase, the total significance of the extended region must be less than the original region, as long as it still contains an excess.

Based on this comparison, regions are immediately rejected if a more significant subregion is already included in the candidate list. The results of this treatment can be observed in figure 4.2. The distribution is exactly the same as in 4.1, but the algorithm does not focus overly on the event in the tail, leading to the correct RoI to be found.

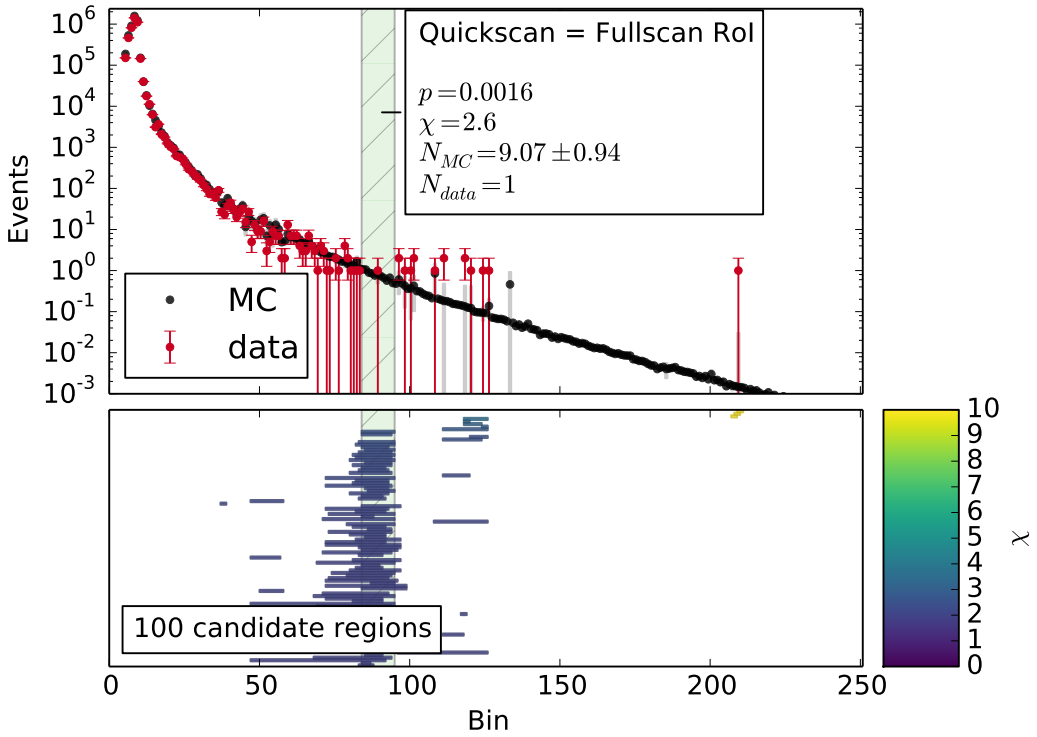

Figure 4.2: Example distribution: pseudo experiment of the 2e $\sum |\vec{p}_{\mathrm{T}}|$ distribution, with special treatment of nested regions. The scheme is explained below figure 4.1. This time, only two candidate regions in the high energy tail are selected.

4.2 The Final Algorithm

The solutions suggested in this chapter are combined in the final algorithm: The Quickscan algorithm maintains a single list of n_{regions} candidates for each distribution in each class. Every time a new candidate is considered for insertion into the list, first the nested region criteria are evaluated. If a parent region is found for which the criteria are fulfilled, the candidate immediately replaces the parent region already in the list. Otherwise, its χ value is computed and compared to the candidates in the list. If it is larger than the lowest χ inside the list, the region is also inserted.

After all connected regions have been considered, the p value is evaluated for the candidate collection and the final region of interest is determined as one of the candidates.

4.3 Running on Pseudo-Experiments Only

The Quickscan method is only applied to pseudo-experiments, not when calculating the p value of measured data, since there might still be cases in which a scan using Quickscan does not find the same region of interest as without. This assures that the data region of interest and its p value are definitely correctly computed.

4.4 Other Investigated Approaches

Another method that has been considered during this work is *Magnitude Binning*. This ad-hoc solution also targets the low-statistic regions by individually treating regions with different N_{SM}. Instead of keeping one candidate list for an entire distribution, Magnitude Binning keeps one candidate list for each magnitude of N_{SM}. The magnitude index is calculated as

$$i = \left\lfloor \log_{n_{\mathrm{base}}}(N_{\mathrm{SM}}) \right\rfloor = \left\lfloor \frac{\log(N_{\mathrm{SM}})}{\log(n_{\mathrm{base}})} \right\rfloor \tag{4.4}$$

This introduces a new parameter, n_{base}, which indicates the size of a magnitude bin. Additionally, the total number of candidate regions is increased by an unknown amount, since the number of orders of magnitude are originally unknown.

This method has been dropped since the other methods proposed in this chapter sufficiently satisfy the Quickscan goals. The Magnitude Binning extension only introduces unnecessary complexity and an additional parameter.

5 Optimization of the Algorithm

In this chapter, the optimal value for the Quickscan parameter n_{regions} is determined. This value highly impacts performance and statistical accuracy. Possible approaches for a trade-off between speed and quality will be proposed.

5.1 Metrics

The optimization is performed in regard of the goals described in chapter 3: Statistical performance and required computation time. The comparison reference in both cases is the execution of the scanning algorithm without Quickscan. This scenario is called *full scan* here.

5.1.1 Statistical Sensitivity

The statistical sensitivity is measured as deviation of $\tilde{p}$. To calculate this deviation, the trial scanning run is compared to a reference run with the same settings. Since the $\tilde{p}$ value is only evaluated for distributions with a RoI p value of < 0.3, the reference as well as the sample run only contain $\tilde{p}$ values for a subset of event classes (usually about half).

For each event class, the $\tilde{p}$ value of the reference run is compared to the corresponding $\tilde{p}$ value of the trial run:

$$\sigma_{\text{rel}} = \frac{\Delta\tilde{p}}{\tilde{p}_{\text{reference}}} = \frac{\tilde{p}_{\text{sample}} - \tilde{p}_{\text{reference}}}{\tilde{p}_{\text{reference}}} \tag{5.1}$$

Usually, a full scan is used for $\tilde{p}_{\text{reference}}$, while a run with the Quickscan yields $\tilde{p}_{\text{sample}}$.

Figure 5.1 shows a comparison between two separate full scans. Here, one full scan is used as reference, the other one as sample. The visible deviation of σ_{rel} is caused by the random dicing of pseudo experiments while calculating $\tilde{p}$ (see section 2.5.3). The conclusion of this illustration is that dicing the pseudo-experiments introduces a statistical uncertainty of $\mathcal{O}(5\,\%)$ on the $\tilde{p}$ value. This value will act as guideline for the Quickscan parameter choice.

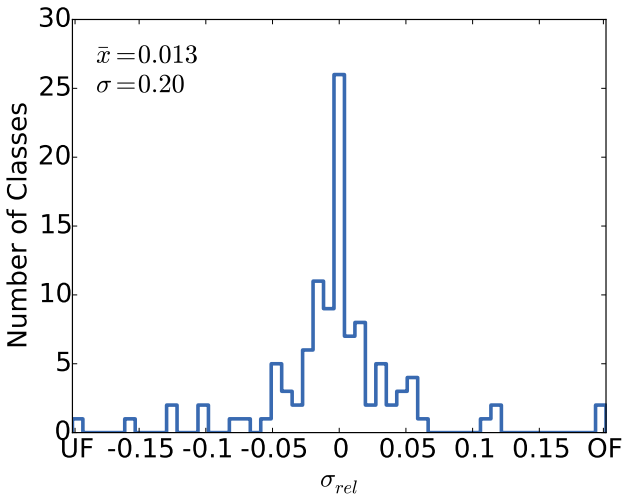

Figure 5.1: Random deviation of σ_{rel}. Obtained by comparing two sets of scan results without the Quickscan algorithm. The distribution spread of $\mathcal{O}(5\,\%)$ originates in the random means of the pseudo experiments. This also demonstrates that only a few classes show no deviation at all.

5.1.2 Computation Time

There are various approaches to comparing the computation costs of computer programs: A commonly used option is to count the number of CPU cycles that a program has used. This method allows for comparisons with a high resolution, but this "pure" CPU time does not represent the time spent on a real-world application. Additionally, it is difficult to implement, especially in an environment like the MUSiC scanning algorithm because of multitasking and the usage of various technologies (Python and C++).

The alternative technique used in this work is *wall-clock time*. The measurement of wall-clock time is performed by acquiring a timestamp before and after the execution of the program. The elapsed time is the difference between those two timestamps. The results of such measurement can be directly transfered to the real-world application because it is performed in the same environment as an execution on a user machine.

This method introduces the following systematic uncertainties:

- CPU usage by other processes: The measurement is performed on a system shared by multiple users. The operating system distributes the available CPU resources between the user processes. Thus, the benchmarked program runs significantly slower if other computation intensive processes are present. In order to suppress this effect, it is ensured that at the time of the measurement, the machine is not occupied by other users. The influence can be lowered this way, from $\mathcal{O}\left(10^{-1}\right)$ to $\mathcal{O}\left(10^{-2}\right)$.

- IO throughput: The files required by the scanning algorithm are stored on shared network drives. Similarly to the CPU usage, the behavior of other users influences the measured wall-time (also $\mathcal{O}\left(10^{-2}\right)$).
- Static overhead: The measurement includes operations that are not being optimized and contribute a constant amount of time. Examples are reading and parsing of parameters and configuration, creating and destroying multitasking worker processes and writing output files. This takes up $\mathcal{O}\left(10^{-2}\right)$ of the total time but is expected to be almost constant.
- Timestamp resolution: Although the operating system's internal clock has a high resolution, the results are only written to file with a resolution of 1 s. In comparison to the other uncertainties, this is negligible ($\mathcal{O}\left(10^{-4}\right)$).
- Time adjustments: The time on the computing host is managed by the Network Time Protocol (NTP). It ensures time synchronization inside the datacenter. If the NTP daemon notices that either the time of the host has shifted or if leap seconds are introduced, the host time is adjusted. This has an effect on the acquired timestamps but is highly negligible in this scenario ($\mathcal{O}\left(10^{-6}\right)$).

The computation improvement is quantified by the ratio between the wall-clock time measurement of the full scan and the Quickscan algorithm:

$$\text{speedup} = \frac{T_{\text{full}}}{T_{\text{quickscan}}} \geq 1 \tag{5.2}$$

5.2 Optimization Environment

The optimization is performed in 60 processes on a 64-core MUSiC host, whose specifications are listed in table 5.1. The wall-time is measured for the entire execution of the main scanning script (`MISMaster.py`).

As shown earlier, σ_{rel} deviates by about 5 % through random dicing. The optimization should occur isolated from this deviation.

The first measure against random influences is to seed the pseudo random number generator (PRNG) with a fixed value. Additional non-determinism is introduced by the parallelization implementation. An optimization called *hit-threshold* allows the Quickscan to stop generating pseudo-experiments once the $\tilde{p}$ value is determined to a sufficient precision. Not all parallelized workers can be terminated in a deterministic sequence, the actual order depends on uncontrollable external factors. Subsequently, the PRNG state is not deterministic. To counteract this effect, the hit-threshold method is turned off.

processors	AMD Opteron Processor 6272
clock speed	2.1 GHz
cores per CPU	8 logical (4 physical)
total number of cores	64
memory	$\approx$ 250 GB
operating system	Linux
kernel version	2.6.32-504.16.2.el6.x86_64
distribution	Scientific Linux release 6.5 (Carbon)
CMSSW version	5.3.14
GCC version	4.7.2
Python version	2.6.4

Table 5.1: Specification of the 64-core machine on which the computation measurement is performed.

Because of degraded overall performance, only 10^3 pseudo-experiments are generated and evaluated, as opposed to up to 10^5 in normal operation mode.

Determinism of this configuration is evaluated by comparing the results of two full scans. The comparison yields absolutely no deviation in σ_{rel}.

5.3 Execution

The optimization run is performed on a subset of data from 2012 taken at $\sqrt{s} = 8$ TeV. Only the $\sum |\vec{p}_{\text{T}}|$ distribution of all exclusive event classes is regarded. Additionally, for a given combination of leptons, all events containing at least 2 jets are summarized in one event class $\boxed{\ldots + \text{2jet} + \text{Njet}}$. In order to save time, classes with a data p value above 0.3 are excluded from the $\tilde{p}$ calculation and thus excluded from the Quickscan, which is only applied on pseudo-experiments. This reduces the number of regarded classes from 214 to 108. The full settings are listed in the appendix, in the left column of table A.2.

For each parameter value, the scanning step is executed 5 times using 60 processes each. The raw measurement results can be found in tables A.3 and A.4 in the appendix. Figure 5.2 illustrates these results. The horizontal axis shows the parameter value for n_{regions}, the vertical axes show the runtime speedup and the number of classes where the Quickscan $\tilde{p}$ value does not deviate from the full scan $\tilde{p}$ within its uncertainty. The vertical error bars on the speedup measurement indicate the uncertainty of the mean speedup value and have been obtained by calculating the

sample error of the 5 trial results. As expected, this deviation is up to $\mathcal{O}\left(10^{-1}\right)$ due to the uncertainties discussed above.

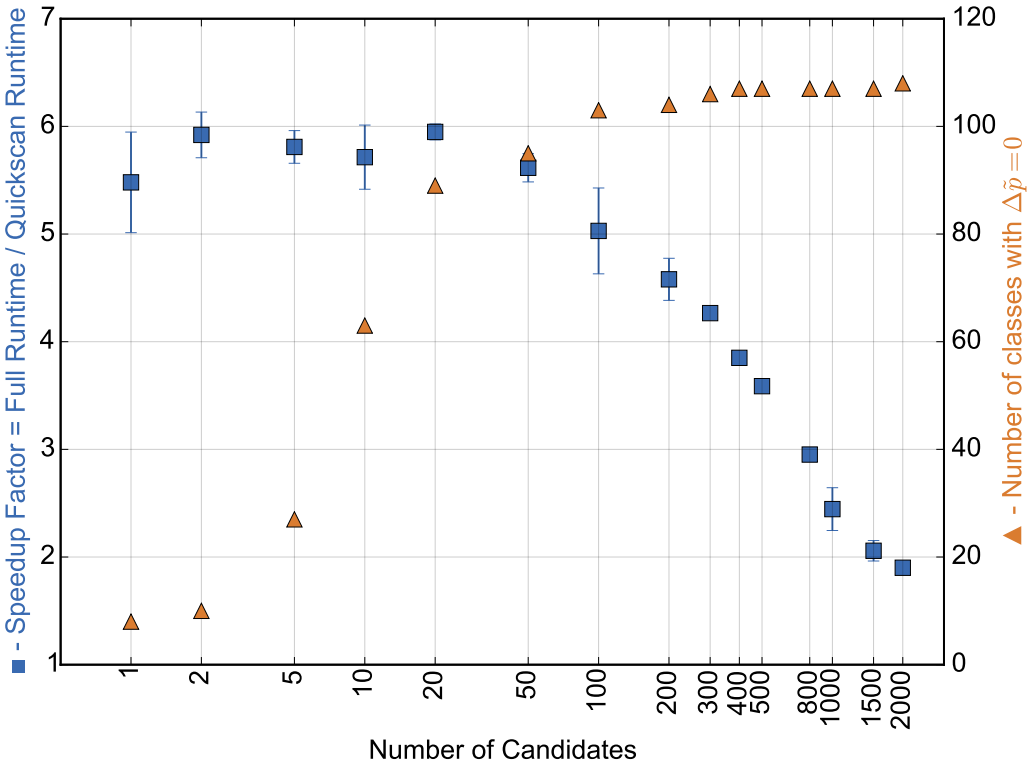

Figure 5.2: Combined results of the runtime and $\Delta\tilde{p}$ measurement for varying numbers of Quickscan RoI candidates. The error bars on the speedup measurement indicate the uncertainty $\sigma_{\overline{T}} = \sigma_T/\sqrt{N}$ on the mean $\overline{T}$ during the 5 trial runs. As expected with a deterministic run, there is no uncertainty of $\Delta\tilde{p}$.

5.4 Analysis

The results match the expectations: a larger number of candidates raises the chances of including the "true" RoI inside the number of candidates. This improves the accuracy of $\tilde{p}$, such that more classes show no deviation ($\Delta\tilde{p} = 0$, $\sigma_{\text{rel}} = 0$). Thus, the number of classes without deviation converges to the total number of classes (108). In the same limit, the speedup value tends towards 1, which indicates that the Quickscan takes the same amount of time as the full scan in that case.

Figure 5.2 shows that even for 1 500 candidates, one class has $\Delta\tilde{p} \neq 0$. The affected event class is 3e, which has a full scan $\tilde{p} = 0.735$ and shows a deviation of $\Delta\tilde{p} = -0.001$. Since this difference is insignificant, it is not further discussed here.

To obtain a recommendation value for n_{regions}, the $\Delta\tilde{p}$ result should be compared with figure 5.1, which showed that the spread induced by random dicing only preserves $\tilde{p}$ for a few classes. Taking 5 candidates, the spread induced by the Quickscan has a similar magnitude. Figure 5.3 shows that for $n_{\text{regions}} > 100$ the width of the distribution is negligible.

Note that while the p value of data p_{data} is unaffected by the Quickscan, a pseudo experiment's p value obtained with the Quickscan is larger or equal to the p value which would have been obtained by the full scan. The full scan calculates the p value for all connected regions, while the Quickscan calculates the p value only for its candidate regions. Thus, for the Quickscan less pseudo experiments show a more significant deviation than p_{data}, resulting in a smaller (or equal) overall $\tilde{p}$ value. Because of that, the σ_{rel} distribution of Quickscan results is asymmetric.

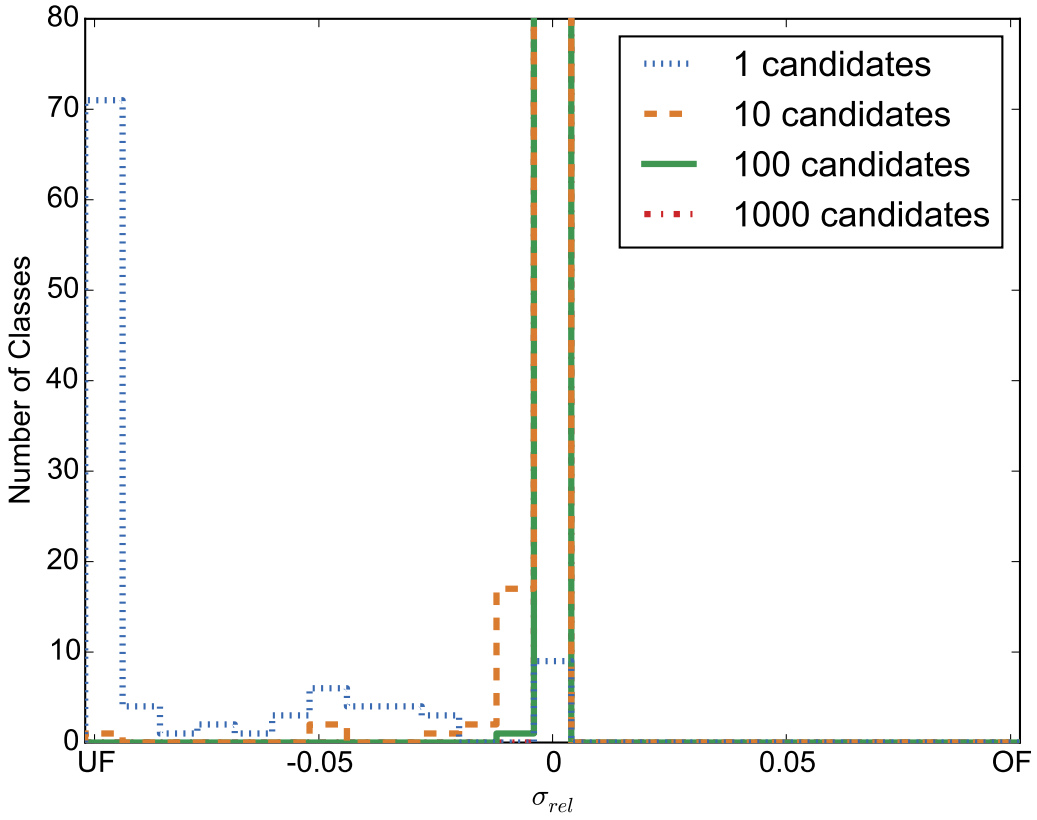

Figure 5.3: Deviation of σ_{rel} due to the Quickscan algorithm. When taking more than 10 candidates, the spread is less than 1 % and can be neglected compared to the other random influences.

5.5 Optimization Conclusion

Concluding, this analysis can make two suggestions for the choice of n_{regions}: one could either argue conservatively and require that the Quickscan influence becomes completely negligible. A possible choice then would be e.g. $n_{\text{regions}} = 1\,000$, but this only provides a moderate speed-up factor of ≈ 2.5 times. If more performance is needed, one could require that both the random influence as well as the influence induced by the Quickscan algorithm become comparable. This is the case at about $n_{\text{regions}} = 10$, where the speedup is already in the saturation region of 6 times.

A sensible choice will be somewhere in between those two extrema:

$$n_{\text{regions}} = 200 \tag{5.3}$$

This choice will be evaluated during the validation run in the following chapter.

6 Running on a Validation Sample

After the entire optimization has been performed on the $\sum |\vec{p}_{\mathrm{T}}|$ distribution of exclusive event classes, the validations must run on a separate subset of data. There are two options: the analysis of inclusive event classes or scanning of a different distribution. The latter option is favored and chosen because it keeps the number of event classes approximately the same. Thus, the validation run is performed on the M_{inv} distribution. As proposed in the previous chapter, the Quickscan number of candidate regions is fixed to $n_{\mathrm{regions}} = 200$.

The other settings for the validation run can be found in table A.2. This configuration does not allow for a deterministic run, thus some random deviation of $\tilde{p}$ is expected. When comparing the result of the Quickscan run with a full scan, the random deviation of $\tilde{p}$ will show up as a finite width of the σ_{rel} distribution. This width is then compared to a σ_{rel} distribution obtained by comparing the results of two full scans.

The results of this validation run are illustrated in figure 6.1. The top figure shows the comparison between the two full scan runs. As expected, the σ_{rel} distribution shows a finite width due to random influences. The other six histograms show the comparison between the full scans and each of the three Quickscan results. None of the distribution means is significantly shifted away from 0. The distribution widths are within their deviations compatible with the broadening due to random fluctuations.

Since the number of pseudo-experiments is significantly greater in this scenario, the constant computation overhead has less influence on the computation time. As shown in table 6.1, a speedup of 9.2 ± 0.5 times is achieved.

Overall, the validation run has shown that the parameter choice $n_{\mathrm{regions}} = 200$ is reasonable. With the aid of the Quickscan, the computation time could be decreased by a factor of about 9 without changing the results.

	Measurement / s	Combined / s	Speedup
Full Scan 1	28 684	$30\,373 \pm 1\,689$	9.2 ± 0.5
Full Scan 2	32 062		
Quickscan 1	3 386	$3\,310 \pm 52$	
Quickscan 2	3 212		
Quickscan 3	3 334		

Table 6.1: Results for the computation time measurement in seconds. Two full scan trials and three Quickscan trials are performed. The second column shows the combined result as mean and its error.

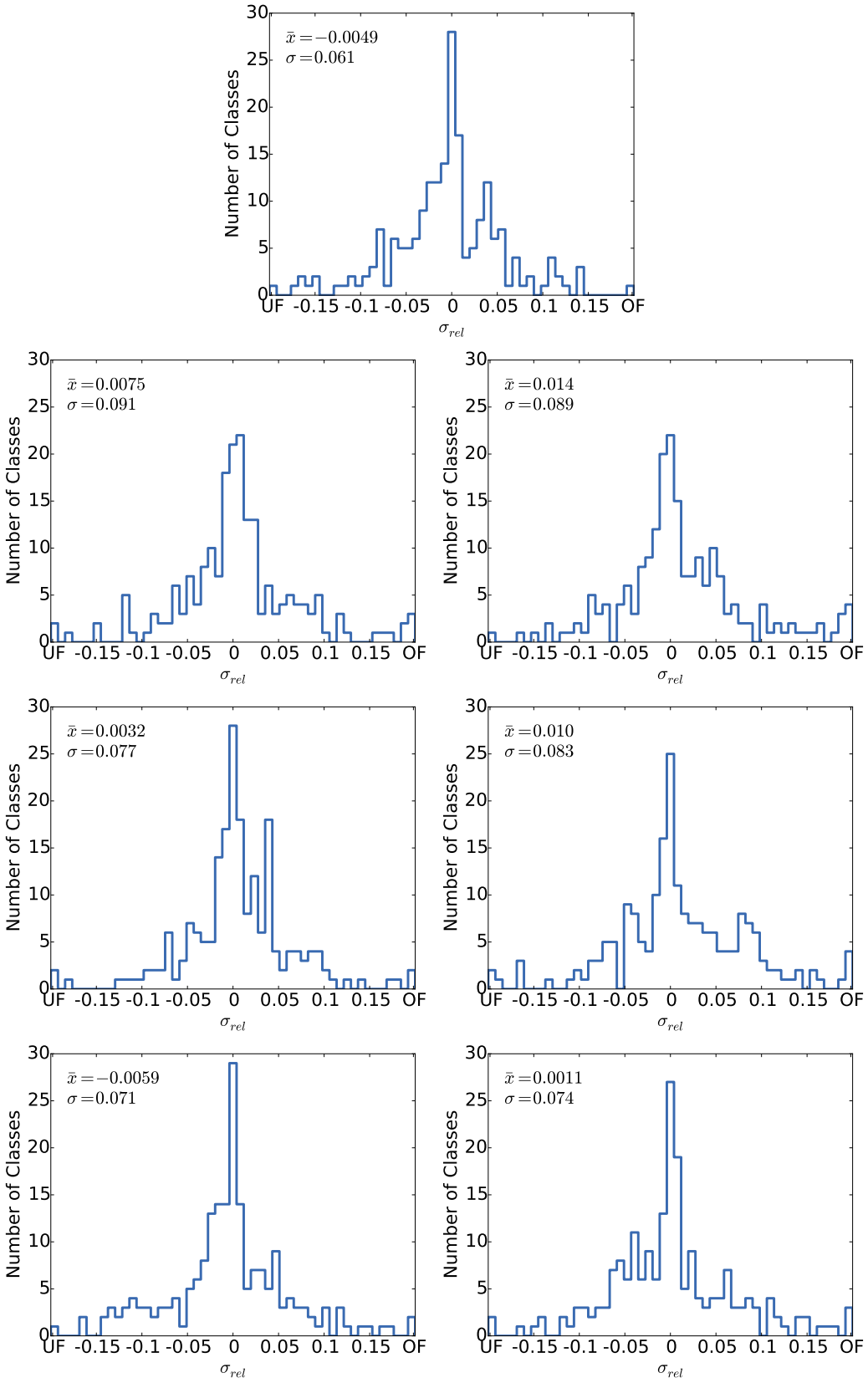

Figure 6.1: Distribution of the relative $\tilde{p}$ deviation σ_{rel}.
Top: comparison between two full scan runs.
Left side: comparison between full scan run 1 and the Quickscan runs 1 to 3.
Right side: comparison between full scan run 2 and the Quickscan runs 1 to 3.

7 Conclusion

The aim of this work was to develop a fast search algorithm for the MUSiC framework. After assessing the current performance situation, the concept of a region preselection step was considered. Multiple aspects of the algorithm were adapted to problems arising in the current implementation. The emerging solution is called *Quickscan*. The Quickscan algorithm uses a less computation intensive estimator to assemble a list of candidate regions. Problems due to low statistics in the high-energy-tail of the kinematic distributions are suppressed by dedicated handling of nested regions. The original p value is subsequently only computed for the few collected candidates. The algorithm has been implemented and tested in this work. A value for the number of candidate regions, $n_{\text{regions}} = 200$ has been proposed here. It has been obtained by an optimization run over a wide parameter range. The choice was additionally validated over a separate subset of data. During this validation run, a performance gain of 920 % was observed. This is mostly due to the smaller number of integrals which remain to be computed after the Quickscan selection.

Additionally, the Quickscan has not only been developed and validated, but also implemented and integrated into the existing MUSiC framework.

7.1 Outlook

Even though the result of this work increases scanning efficiency by almost 10 times, there is room for improvement:

The Quickscan algorithm could greatly benefit from an estimator that improves handling of regions with a low number of events. A possible solution could be to use a Poisson distribution around the number of MC events and calculate its p value back to a Gaussian statistic before combining with the systematical uncertainty. This could lead to a better mathematical understanding of the algorithm, especially since the ad-hoc nested region handling may turn out to be superfluous.

Furthermore, the parallelization of the general scanning algorithm could be improved. The correlated dicing of the pseudo-experiments requires communication between the worker tasks. The correlation could instead be handled by a fixed seed of the worker's pseudo-random number generators. This would make the scanning step trivially parallelizable (either over the classes or over the pseudo-experiments), enabling it to run on a computing grid instead of the 64-core MUSiC host.

Bibliography

[1] **Particle Data Group** Collaboration, K. Olive *et al.*, "Review of Particle Physics," *Chin.Phys.* **C38** (2014) 090001.

[2] **ATLAS, CMS** Collaboration, G. Aad *et al.*, "Combined Measurement of the Higgs Boson Mass in pp Collisions at $\sqrt{s} = 7$ and 8 TeV with the ATLAS and CMS Experiments," *Phys.Rev.Lett.* **114** (2015) 191803, `arXiv:1503.07589 [hep-ex]`.

[3] F. Marcastel, "CERN's Accelerator Complex." CERN Graphic Design service, Oct, 2013.

[4] A. Breskin and R. Voss, *The CERN Large Hadron Collider: Accelerator and Experiments*. CERN, Geneva, 2009. `https://cds.cern.ch/record/1244506`.

[5] L. Evans and P. Bryant, "LHC Machine," *JINST* **3** (2008) S08001.

[6] **CMS** Collaboration, D. Duchardt, T. Hebbeker, S. Knutzen, A. Meyer, and P. Papacz, "MUSiC - A Model Unspecific Search for New Physics in pp Collisions at $\sqrt{s} = 8$TeV." CMS AN-2014/098, 2014.

[7] **CMS** Collaboration, S. Chatrchyan *et al.*, "The CMS experiment at the CERN LHC," *JINST* **3** (2008) S08004.

[8] D. Barney, "CMS slice raw illustrator files," 2011. `https://cms-docdb.cern.ch/cgi-bin/PublicDocDB/ShowDocument?docid=5581`. CMS Document 5581-v1.

[9] **CMS** Collaboration, "Particle-Flow Event Reconstruction in CMS and Performance for Jets, Taus, and MET,". `https://cds.cern.ch/record/1194487`.

[10] H. Pieta, *MUSiC - A Model Unspecific Search in CMS based on 2010 LHC data*. PhD thesis, Technische Hochschule, Aachen, 2012.

http://publications.rwth-aachen.de/record/82887.

[11] P. Papacz, *Model Unspecific Search for new Physics in CMS Based on 2011 Data*. PhD thesis, Technische Hochschule, Aachen, 2014. http://publications.rwth-aachen.de/record/465391.

[12] D. Duchardt, "MUSiC: A Model Unspecific Search for New Physics Based on CMS Data at $\sqrt{s} = 8$ TeV,".

[13] H. Bretz, M. Brodski, M. Erdmann, R. Fischer, A. Hinzmann, *et al.*, "A Development Environment for Visual Physics Analysis," *JINST* **7** (2012) T08005, arXiv:1205.4912 [physics.data-an].

Appendix

A

Time per Integral

Histograms	Number of Integrals	Runtime / s	Per Integral / µs
2 369	23 110 209	5 179.87	224.1
2 374	23 146 885	5 190.3	224.2
2 390	22 968 240	5 198.89	226.4
2 375	23 089 566	5 218.37	226.0
2 355	22 741 383	5 204.23	228.8
2 357	23 079 630	5 172.68	224.1
2 377	23 119 389	5 183.55	224.2
2 389	23 208 732	5 194.77	223.8
2 361	23 288 752	5 221.18	224.2
2 368	22 925 427	5 217.4	227.6
18 991	81 734 778	16 983.6	207.8
19 017	80 847 194	16 970.2	209.9
18 945	80 976 157	16 990.5	209.8
18 993	81 748 828	16 987.3	207.8
19 079	81 471 964	16 985.9	208.5
18 853	81 188 030	16 982	209.2
18 906	81 149 116	17 019.2	209.7
18 960	81 354 379	17 003.4	209.0
18 976	81 185 262	16 967.9	209.0
19 089	81 772 747	16 987.6	207.7

Table A.1: Benchmarking results of the original scanning process. The timing has been measured on a data subset, using a single CPU and includes setting up the process and writing out the results. The agreement of the time per integral between the runs motivates the rough estimate of 200 µs per integral.

MUSiC Scanner Configuration

	Optimization	Validation
Data	2012, $\sqrt{s} = 8\,\text{TeV}$	
Luminosity	$L = 19.7\,\text{fb}^{-1}$	
Event classes	excl.	excl.
Distribution	$\sum \lvert\vec{p}_\text{T}\rvert$	M_inv
N-jet threshold	2	-
p threshold	0.3	0.3
Dicing rounds	1 000	100 000
Hits threshold	-	100
Fill-Up	yes	yes

Table A.2: MUSiC configuration values for the optimization and validation runs.

Optimization $\tilde{p}$ Results

n_{regions}	Classes with $\Delta\tilde{p} = 0$	Percentage of Total Classes
1	8	7.4
2	10	9.3
5	27	25.0
10	63	58.3
20	89	82.4
50	95	88.0
100	103	95.4
200	104	96.3
300	106	98.1
400	107	99.1
500	107	99.1
800	107	99.1
1 000	107	99.1
1 500	107	99.1
2 000	108	100.0

Table A.3: Result of the optimization $\Delta\tilde{p}$ measurement. There are 108 classes in total for which the $\tilde{p}$ value is computed and as such for which the Quickscan is applied. All 5 trial runs yield exactly the same results, thus they are not separately listed here.

Optimization Timing Results

n_{regions}	T_1 / s	T_2 / s	T_3 / s	T_4 / s	T_5 / s	$\overline{T} \pm \Delta T$	Speedup
Full	5 379	5 500	5 331	5 371	5 396	$5\,395 \pm 28$	1.0 ± 0.0
1	1 460	880	973	912	885	$1\,022 \pm 111$	5.3 ± 0.6
2	1 054	870	871	900	886	916 ± 35	5.9 ± 0.2
5	945	1 038	898	909	869	932 ± 29	5.8 ± 0.2
10	1 142	861	985	907	879	955 ± 51	5.7 ± 0.3
20	930	886	889	919	913	907 ± 9	5.9 ± 0.1
50	940	1 073	952	937	915	963 ± 28	5.6 ± 0.2
100	1 555	987	1 031	993	975	$1\,108 \pm 112$	4.9 ± 0.5
200	1 406	1 120	1 111	1 171	1 130	$1\,188 \pm 56$	4.5 ± 0.2
300	1 251	1 330	1 251	1 248	1 247	$1\,265 \pm 16$	4.3 ± 0.1
400	1 373	1 507	1 375	1 395	1 364	$1\,403 \pm 27$	3.8 ± 0.1
500	1 474	1 594	1 489	1 481	1 487	$1\,505 \pm 22$	3.6 ± 0.1
800	1 792	1 860	1 826	1 816	1 846	$1\,828 \pm 12$	3.0 ± 0.0
1 000	3 254	2 045	2 064	2 022	2 041	$2\,285 \pm 242$	2.4 ± 0.3
1 500	3 203	2 544	2 505	2 466	2 518	$2\,647 \pm 140$	2.0 ± 0.1
2 000	2 971	2 765	2 820	2 787	2 863	$2\,841 \pm 36$	1.9 ± 0.0

Table A.4: Result of the optimization computation time measurement. The numbers show the wall-clock time. The combined result $\overline{T} \pm \Delta T$ consist of the sample mean $\overline{T} = \frac{1}{N}\sum_i T_i$ and its deviation $\Delta T = \frac{1}{\sqrt{N}}\sqrt{\frac{1}{N}\sum_i (T_i - \overline{T})^2}$. As expected, the wall-clock time is subject to fluctuations of about 1 % to 10 %. The resulting speedup $\frac{T_{\text{Full}}}{T_{\text{QS}}}$ is calculated using the mean values and is up to 5.9 times.

Danksagung

Ich möchte mich ganz herzlich bei allen Menschen bedanken, die mir beim Erstellen dieser Arbeit zur Seite standen.

Das behandelte Thema schlägt eine Brücke zwischen meinem Studium, der Teilchenphysik, und meinem Hobby, der Softwareentwicklung. Dafür und für die Unterstützung während des Verfassens der Arbeit danke ich ganz besonders Herrn Professor Thomas Hebbeker.

Am III. Physikalischen Institut wurde ich als Bachelor sehr herzlich aufgenommen und danke dafür allen Mitgliedern der Aachen-IIIA-CMS-Gruppe. Eine besonders gute Atmosphäre herrschte dabei innerhalb der MUSiC-Gruppe, bestehend aus Deborah Duchardt, Simon Knutzen, Tobias Pook und Andreas Albert. Durch das Beantworten hunderter Fragen hat die gesamte Physikgruppe mein Physik- und Wissenschaftsverständnis geprägt.

Explizit möchte ich Simon Knutzen und Deborah Duchardt dafür danken, dass sie mich bei der Entwicklung des Quickscans in die richtige Richtung geleitet haben und meine Arbeit im Anschluss probegelesen haben.

Des Weiteren möchte ich mich für das finale Korrekturlesen der Arbeit und für hilfreiche Hinweise und Unterstützung während der Zwischenpräsentationen bei Dr. Arnd Meyer bedanken. Ebenfalls danke ich Herrn Professor Martin Erdmann, der sich bereit erklärt hat, diese Arbeit als Zweitkorrektor zu betreuen und mich zuvor mit den Vorlesungen der Teilchenphysik für eine Bachelorarbeit in diesem Feld begeistern konnte.

Abschließend bedanke ich mich bei meiner Familie, meinen Freunden und meinen Mitbewohnern für die mentale Unterstützung, die ich in sowohl in den stressigeren letzten Monaten als auch im Rest des Studiums genießen konnte.