Faculty of Physics and Astronomy

University of Heidelberg

Bachelor thesis

in Physics

submitted by

Marc Barra

born in Waiblingen

August 2013

Investigation on J/ψ production in Monte Carlo Simulations

with the C^{++} framework RIVET

This Bachelor thesis has been carried out by Marc Barra

at the

Physikalisches Institut

under the supervision of

Dr. Kai Schweda

Abstract: This report is a set of Monte Carlo (MC) studies related to the production of J/ψ meson in pp collisions at 7 TeV. The MC analysis is done with Rivet, a toolkit designed to simplify MC analyses and interface various MC generator. For the generation of MC events, Pythia 8 is used as an example.

The main aspect lies on the comparison between MC outcome and ALICE results about inclusive J/ψ production. Differential studies of the charmonium production are carried out: the production as a function of the J/ψ transverse momentum (p_{T}) and rapidity (y), as well as the dependence to the charged particle density $\mathrm{d}N_{\mathrm{ch}}/\mathrm{d}\eta$. For the p_{T} dependence, the MC inclusive J/ψ signal is separated into its various components: direct production, feed down from χ_{cJ}, ψ(2S) and from B mesons.

Kurzfassung: Der vorliegend Bericht ist eine Zusammenfassung von Monte Carlo (MC) Studien über die Production des J/ψ Mesons in pp Kollisionen bei 7 TeV. Die MC Analyse wird mit Rivet durchgeführt. Einem Werkzeugsatz, um MC Analysen zu vereinfachen, der mit verschiedene MC generatoren verwednet werden kann. Für die Simulation der Kollisionen wird Pythia 8 verwendet.

Das Hauptaugenmerk liegt auf dem Vergleich zwischen MC und ALICE ergebnissen über die inclusive J/ψ Produktion. Verschieden Untersuchungen bezüglich der Charmonium produktion werden durchgeführt: die Produktion als Funktion des transversal Impulses (p_{T}) und der Rapidität (y), ebenso wie die Abhängigkeit von der Dichte geladener Teilchen $\mathrm{d}N_{\mathrm{ch}}/\mathrm{d}\eta$. Für die p_{T} Abhängigkeit wird das inclusive J/ψ Signal in die einzelnen Komponenten aufgeteilt: direkte Erzeugung, Zerfälle von χ_{cJ}, ψ(2S) und von B Mesonen

Contents

Introduction

Already in ancient Greece philosophers made up their minds about the question how matter was constructed. Empedocles (about $494 - 434$ BC) for example was committed to the fact that everything was made out of four "roots", how he called them: soil, air, fire, and water. He said that the transformation from one thing into the other, e.g. from fire to smoke, is just a change in the constituents. A first theory of matter, sin the spirit of the one we have today, was made up by Democritus (about $460 - 370$ BC). He asked the question: "what happens if I divide this stone into two halves, and what happens if I take one of the two halves and divide it again, and again, and again?" He came to the conclusion, that there must be a smallest piece which could not be divided any more. This he called "atom", derived from the greek word *átomos*, the impartible and stated, that many different atoms exist out of which "macroscopic" objects are formed.

Today the concept of atoms is still alive, however we know they are further divisible. The theory behind this is the Standard Model of particle physics. In it, the atoms gain substructure through quarks and gluons. The technological progress over the last hundred years led to possibilities to investigate these substructure. It turned out that this substructure is very complex so that the theory had to be often corrected and new aspects had to be added.

One of these aspects is the charm quark, which became necessary to explain experimental results in 1974. S. Ting [1] at the Brookhaven National Laboratory (BNL) and B. Richter [2] at the Standford Linear ACcelerator (SLAC) measured a narrow resonance with $m = 3.01$ GeV/c^2, today referred to as J/ψ. Even 40 years after its discovery its hadronic production processes are not fully understood.

In this thesis I will investigate some aspects of J/ψ production in Monte Carlo event generators, in particular Pythia 8 is used to obtain results. For analysing the MC output I will use the framework Rivet and "rivetise" two ALICE analyses on the inclusive J/ψ, [3] and [4].

This thesis is organized as follows: Chapter 1 focusses on some features of the J/ψ. Chapter 2 focusses on the working and use of Monte Carlo event generators. Chapter 3 explains the advantages and working principles of Rivet. The main part of this report is dedicated to the explanation of the used ALICE analyses and their rivetised version as explained in chapter 4. Thereafter the results of the simulation will be presented and discussed in chapter 5. Last but not least a summary of the results and an outlook on future topics related to this thesis will be given in Chapter 6.

1 Some aspects of the J/ψ

Quarkonia are bound states between a quark q and its antiquark $\bar{q}$. Since, in the case of light quarks (u,d,s), the binding energy is not small compared to the energy of the components, they have to be treated relativistically. However, in case of heavy quarks (c,b, *"t'"*), the non-relativistic Schrödinger eguation can be applied [5].

$$\left(-\frac{\hbar^2}{2m}\nabla^2 + V\right)\Psi = \imath\hbar\frac{\partial}{\partial t}\Psi \tag{1.1}$$

Where it remains to specify the potential V. Since we have a system of a particle and its anti-particle, we expect a spectrum of different states, similar to positronium, according to the quantum numbers of the system, Fig. 1.1. Three heavy quarks would lead to three different types of quarkonia. However, quarkonia from $t\bar{t}$ do not exist, due to the fact, that the t quark decays, before it can hadronise. As the $c\bar{c}$ system is the important system of this thesis, the $b\bar{b}$ will be not introduced further.

As the potential between two quarks cannot be derived easily from theory, the potential has to be parametrised. One not so complicated approach is the Cornell potential:

$$V(r) = -\frac{4}{3}\frac{\alpha_s(r)\hbar c}{r} + k \cdot r \tag{1.2}$$

where r denotes the distance between two quarks and $\alpha_s r$ is the strong coupling constant, which is not constant at all but rises to small distances. This behaviour gives rise to the *asymptotic freedom* at short distances. The second term is proportional to r, which leads to an increasing potential at large distances. This is the so-called confinement and the reason why quarks are not detected but only hadrons, consisting of quarks.

The decay of the J/ψ in the strong decay channel is suppressed, due to the necessity of at least three gluons. Thus the width of the J/ψ is very small and the lifetime very long, compared to other particles with comparable masses. This does not apply for the $\psi(3370)$, as its mass is high enough, to decay into two D-mesons.

The decay schemes and mechanisms are understood quite well, however this does not apply for the production processes of J/ψ. Due to the fact, that the J/ψ production lies at the boundary between perturbative and non-perturbative QCD, the process can not be calculated easily from theory. In Fig. 1.2 the dominant production process of a $c\bar{c}$ pair is illustrated. To form a bound state, *e.g.* a J/ψ, they have to interact with each other, which turns out to be the more complex part of

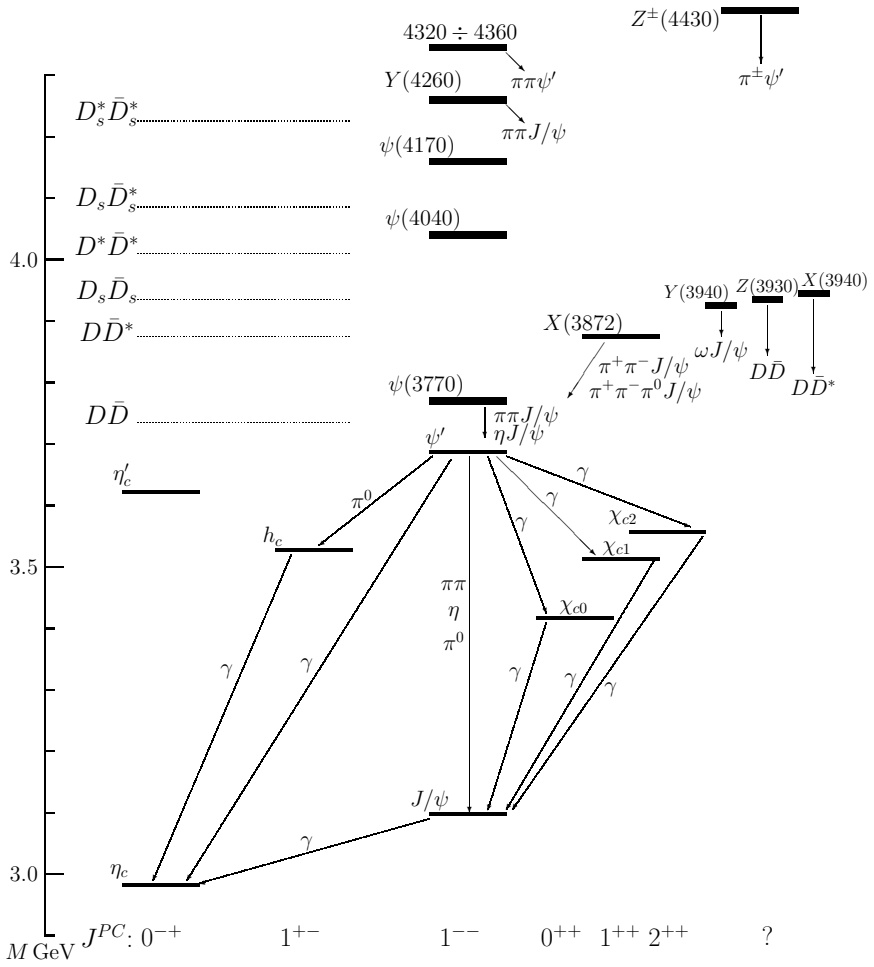


Figure 1.1: The $c\bar{c}$ system and some of the known decays [6].

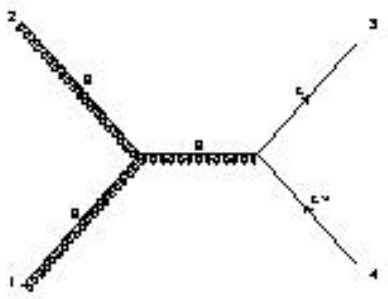


Figure 1.2: Feynman diagram for $c\bar{c}$ production [8].

the creation process. One model to describe this interaction, is the Colour-Singlet Model (CSM). In the CSM model, the $c\bar{c}$ pair needs to have small velocities relative to each other. The binding comes through the fact, that the $c\bar{c}$ pair is already colour neutral [7]. The case of a not colour neutral $c\bar{c}$ pair is described by the Colour-Octet Model (COM). Here the quarks not just fly side by side, but will radiate gluons and thus interact, while forming the bound states. Here the production mechanism is moved more deeply into non-relativistic QCD. There exist other models, but the CSM and the COM are the major ones, to describe J/ψ formation.

In the Colour-octet model, the quarks not just fly side by side, but will radiate gluons and thus interact. Here the production mechanism is moved more deeply into non-perturbative QCD.

2 Monte Carlo event generators

2.1 Monte Carlo methods

Monte Carlo methods use random numbers to obtain numerical results from a priori analytically not solvable problems. A basic example for a Monte Carlo method would be to calculate the number π from a set of random points in a 2-dimensional rectangle, see Fig. 2.1. In general, the precision of the result depends on the count of random numbers. The more random numbers are created the more precise the result will be.

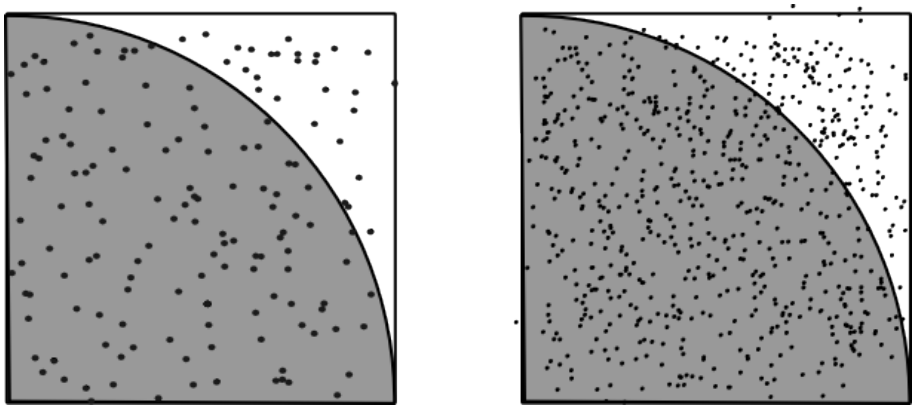


Figure 2.1: Calculation of π via a Monte Carlo method. The points are randomly distributed. The fraction between the points in the grey area and the total number of points in the rectangel is $\pi/4$.

Monte Carlo methods are used in particle physics to simulate particle collisions. However it should be clear, that Monte Carlo simulations in particle physics are a lot more complex than the simple example above. In addition, it is a rather long process in which the collision is simulated and Monte Carlo methods are just one part of the simulation.

There are multiple Monte Carlo event generators which try to simulate the outcome of particle collisions, e.g Pythia 8 [9] , Sherpa [10], Herwig [11], each focussing on different aspects of particle physics. In the following the acronym *MC generator* will be used meaning Monte Carlo event generator.

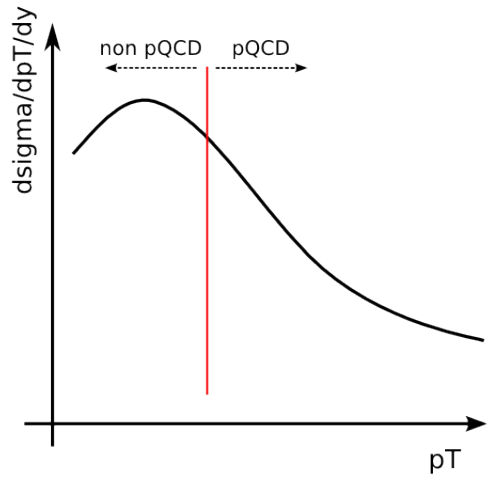


Figure 2.2: Schematic illustration of a p_{T}-spectrum. The red line represents the boundary between perturbative and non-perturbative QCD, with $\alpha_s \approx 1$.

2.2 Use of MC generators

To understand the *final states* of particle collisions, e.g. at the LHC, it is an inevitable consequence that simulations have to be taken into account. This is due to the fact that up to hundreds of unstable particles are produced, in momentum regimes covering many orders of magnitude. To understand the production mechanism of a certain final state, the whole evolution since the first interaction has to be known. In the high momentum regime, the matrix elements can be calculated in the first few orders employing perturbativ QCD. However the higher the order, the more expensive the calculation will be, in means of computing power. In the low momentum regime, meaning with low momentum transfer, we face the problem that

$$\alpha_s = f\left(\frac{\Lambda_{\mathrm{QCD}}}{Q^2}\right) \tag{2.1}$$

increases there. So perturbation theory can not be applied any more, see Fig. 2.2. But also the confinement of quarks can not be embodied easily by ab-initio theory calculations and thus has to be modelled phenomenologically. MC generators are a good possibility to develop and test these models.

In addition to the use of testing of phenomenological models MC generators have multiple purposes:

- Input for modelling of detector effects, so that the background of an investigated signal can be understood better in order to discriminate properly signal from background in data and to determine efficiencies.

- input for the design of new detectors or the improvement of event reconstruction procedures
- Measurement of model parameters, in comparing predictions of MC generators to experimental data

2.3 Different stages of a pp collision in MC simulations

The simulation of proton-proton collisions in MC generators is split into different parts of the process, see Fig. 2.3. The advantage of this lies in the fact, that different parts of the process can be accessed separately and thus can be changed and improved without changing other parts of the simulation. This separation is also physically motivated. The latter categorization originates from the so called factorization between hard and soft parts of the process.

The first step of the simulation is the *primary hard process*. Here partons of incoming beam particles interact in a hard process with large momentum transfer producing an elementary particle (*e.g.* from QCD). As every QCD particle can couple to gluons, this will lead to new gluons and QCD particles. Another process in this step is the conversion from a gluon into a $q\overline{q}$ pair. These particles then will radiate gluons again, and so on. This process is called *parton schowers*, where the momentum transfer, starting from a high momentum transfer at the primary hard process, goes down over all scales so that we reach at some point soft QCD regimes.

The next step is the *hadronization phase*. Here the momentum transfer reaches small values, so that quarks get sensitive to confinement and build up hadrons. These hadrons are mostly hadrons the detector will not see, due to their short lifetime. They will decay into particles which are stable under strong and electromagnetic interaction, *i.e.* protons, π, $K^{\pm}$, electrons, muons, which will then be detected.

The process of gluon radiation is not just restricted to the *parton showers* but can appear in all stages of the collision. This includes also the time before the *primary hard process*, the so called initial state radiator. The radiation of gluons can be simulated via a Monte Carlo phase space integration. But not just the process of gluon radiation has to be taken into account. In the case of electromagnetic interacting particles, also photon radiation has to be calculated. This is done in a similar way.

However, one can imagine, that there will be not just one hard process between two partons of the beam particles, but there is the possibility that a second, a third, and so on, hard process between partons can happen. But not just their occurrence has to be taken into account, but also the interaction and interconnection between neighbouring hard processes. For these hard processes the same procedure of the process described above can be applied, whereas here the energy will be on a smaller scale. This set of secondary hard process contributes significant to the so called *underlying event*.

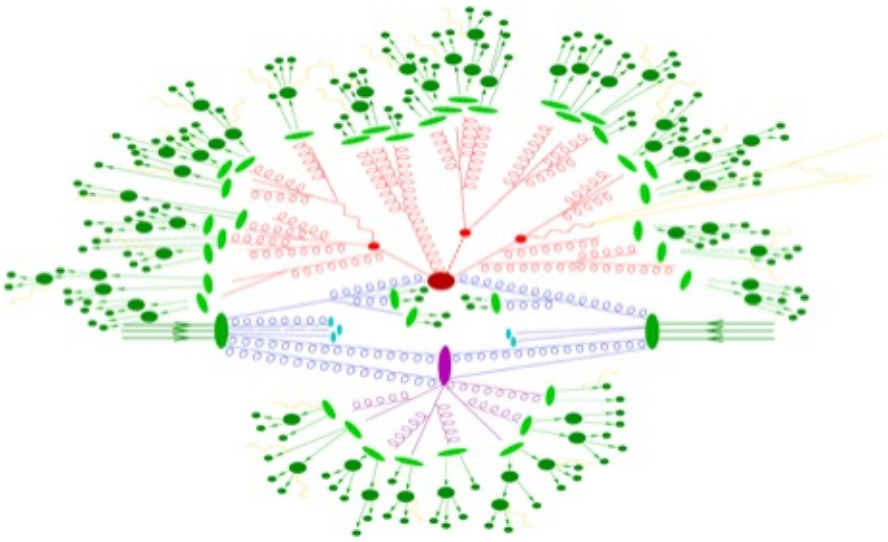


Figure 2.3: Sketch of a proton-proton collision in SHERPA. In red, there is the *primary hard process*. The pink gluons belong to the *parton shower*. The light green circles are hadrons from the *hadronization phase*. The dark green circles are the *final state particles*. Purple denotes the rise of the underlying event [12].

As one can easily imagine it requires a lot of computing power to simulate just one event, it is very difficult to investigate rare processes, like the decay of B_s^0 into $\mu^+\mu^-$, see [13]. Thus it is possible for the user to adapt the simulation to his needs. The first step will be of course to define the conditions of the simulation (beam particles, beam energies, ...) in order to match to the experiment. In case of hadrons as beam particles, the user also has the chance, to choose between different *parton distribution functions*. Since some particles may mainly occur in *soft QCD* interactions, the user can somehow neglect the *very* hard process and focus on the soft processes. Nevertheless, there are particles for which the hard process is of greater importance in their creation. In this case, one can further specify the hard process, *e.g.* $gg \to gg$, $gg \to q\overline{q}$, ...

Since the *soft QCD* processes depend on phenomenological models, the user is able to choose between different models. However, one has to keep in mind, that if no choice is taken at all, there are default settings, which will be used to simulate the event. This does not just refer to *soft* processes but to very parameter which can be specified.

For further and more detailed information about the structure of a simulated event, see [14].

2.4 Pythia 8

Pythia has a slightly long history. It has evolved from Jet Set which was first developed in 1978 to simulate e^+e^- annihilation processes. Since then program has been expanded and new processes has been added. So that Pythia has a know-how from over 30 years of development, but is now also rather cumbersome due to some of the code has not been changed or modulated for a long time. Pythia 8, written in C^{++}, was developed out of Pythia 6, written in Fortran77, in 2007. It tries to get rid of some out-of-date code. Still Pythia 8 is kept very open, meaning that there is a lot of freedom left to the user, to set parameters and choose processes. Thus it can be used to study the behaviour of Pythia in general, but also focus on specific processes and models of interest. Pythia can currently handle e^+e^-, ep, $pp/p\bar{p}$ collisions.

The production mechanisms of J/ψ in Pythia are not fully mastered, due to the fact that the production of J/ψ lies at the boundary of soft and hard QCD. However there are models and processes implemented for colour-singlet and colour-octet production. In case of the colour-singlet production the J/ψ will be created in isolation in a hard process. In case of the colour-octet production of the J/ψ, it is created in a softer process and thus is created during the parton shower[1] processes. [15]

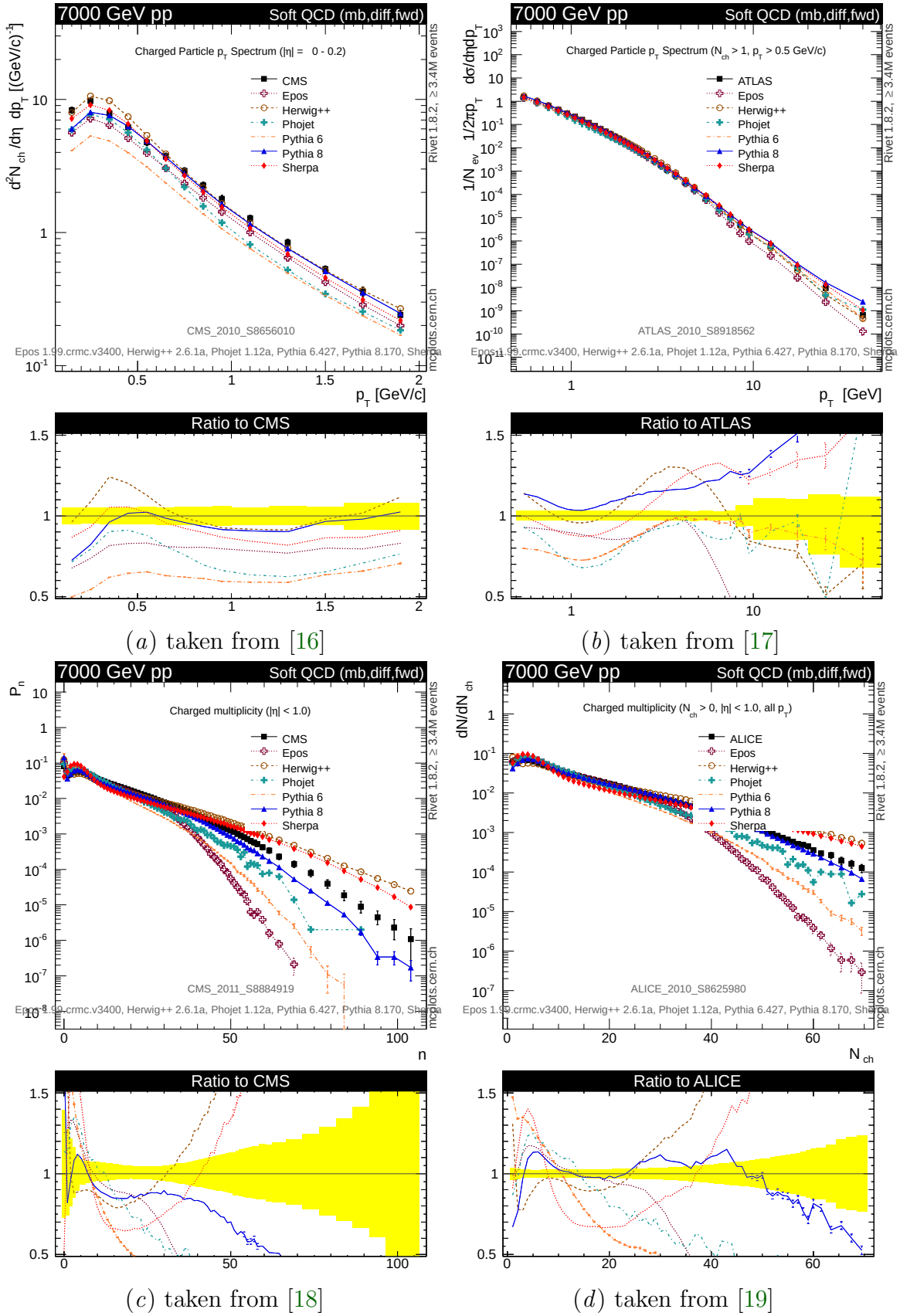
Pythia 8 was chosen, since it can reproduce quite well basic observables of pp collisions at 7 TeV. In Fig. 2.4 some early results from the LHC at $\sqrt{s} = 7$ TeV are shown and compared to the outcome of different MC generators. Namely p_{T} spectra of unidentified charged particles and charged particle multiplicity distributions.

2.5 Tuning of MC generators

It lies in the nature of MC generators that they have a, more or less, large set of relatively free parameters, which have to be adapted to the output of the experiment. The number of parameters is typically in the order of $\mathcal{O}(10-30)$. Examples for these parameters are: the baryon/meson ratio, strangeness production, Λ_{QCD}, but also parameters which have no direct physical analogue itself while are closely connected to observables like transverse momentum and angular distributions.

This freedom in parameters requires the tuning of MC generators which can be done in different ways. Historically, the tuning by hand is one of the most used tuning methods. It requires deep insight into the generators behaviour to the change of parameters and thus is a good possibility to test oneself if the working principles of the MC generator is understood. Due to the fact that it is dependent on the tuner itself, everyone will obtain slightly different results. As the results are not

[1] This is at least the case for Pythia 6.4, whereas in Pythia 8 the production has slightly changed, due to the implementation of a multi-parton interacting models



(*a*) taken from [16]

(*b*) taken from [17]

(*c*) taken from [18]

(*d*) taken from [19]

Figure 2.4: Some plots to illustrate, that PYTHIA 8 has good agreement with early LHC data obtained for pp collisions at $\sqrt{s} = 7$ TeV. Plots from [20].

systematically produced and certainly will not be reproduced, other tuning methods are required.

A second way to tune generators is the brute force-method. A set of n parameters with m possible values for each parameter build up a parameter space of

$$\Omega = m^n \tag{2.2}$$

points. For each of this points the generator has to run at least one time. The best agreement with the data will become the best tune. However this method depends on the choice of the phase space and, as one can see easily in Eq. (2.2), increases dramatically with n and thus requires a lot of simulated events. One can imagine, that simulating just one event for each parameter choice might be not enough, typically there should be about one million events per choice.

The PROFESSOR method [14] takes the approach of parametrisation-based tuning. Here the behaviour of the generator due to the change in parameters is modelled. In comparison to the brute-force method, not so many parameter space points have to calculated and thus requires less simulated events. It is the up to date tuning method, to get systematic approached tunes. It requires comparable histograms from experimental data and MC output. One way to provide these data is RIVET, which was developed together with the PROFESSOR method.

3 Rivet

3.1 Purpose and working

Rivet is a , $C++$ written, library of tools to calculate physical observables from HepMC event record. There is already a large set of already implemented analyses, which can be used for investigations on the behaviour of MC generators. It is intended to be the interface between experiment and MC generator and hence a tool to investigate the output of different MC generators.

HepMC [21] is a convention to store information about the evolution of particle collision in a systematic way into ASCII code. In Fig. 3.1 a small part of an event record in HepMC is shown. These information, *e.g.* momenta, energies, particle identities and the particles history, can be accessed with Rivet. On disadvantage of HepMC event record is, that it require a lot of hard drive capacities, as one single event has already an HepMC event record of about 60.000 lines of ASCII code.

The main idea is, that every published analysis of experimental data should come with a"rivetise" analysis. This then can be used to analyse the output of Monte Carlo event generators and tune them to data. Rivet is able to provide the output in comparable data formats, e.g. the binning of histograms can be the same as for the experimental analysis, a necessity in tuning generators. To rivetise the experimental analysis the most capable person would be from the working group in the collaboration, because they know best the experimental constraints or cuts which have to be taken into account, i.e. rapidity window, energy cuts, analysis tricks, ...

As implied above, Rivet is an easy way to investigate the output of different generators, as long as the generator can provide the output as HepMC event record, see Fig. 3.2. The analysis has to be written only once and can be applied to the output of different MC generators. This has the advantage, that the analysis must not be written specifically for every generator. A huge advantage in the case of the older event generators like Pythia 6 and Herwig 8, which are written in Fortran77. Rivet clearly divides the generating part and the analysis part in a case, that the

```
V -916 0 0 0 0 0 0 3 0
P 1358 323  -1.84e+04 3.35e+03 -8.39e+03 2.05e+04 8.56e+02 2 0 0 -1026 0
P 1359 -311 -3.87e+03 1.09e+03 -7.63e+02 4.12e+03 4.97e+02 2 0 0 -1027 0
```

Figure 3.1: Example for a very small part of the HepMC event record of a pp collision. The values for the momenta and the energy are shortened.

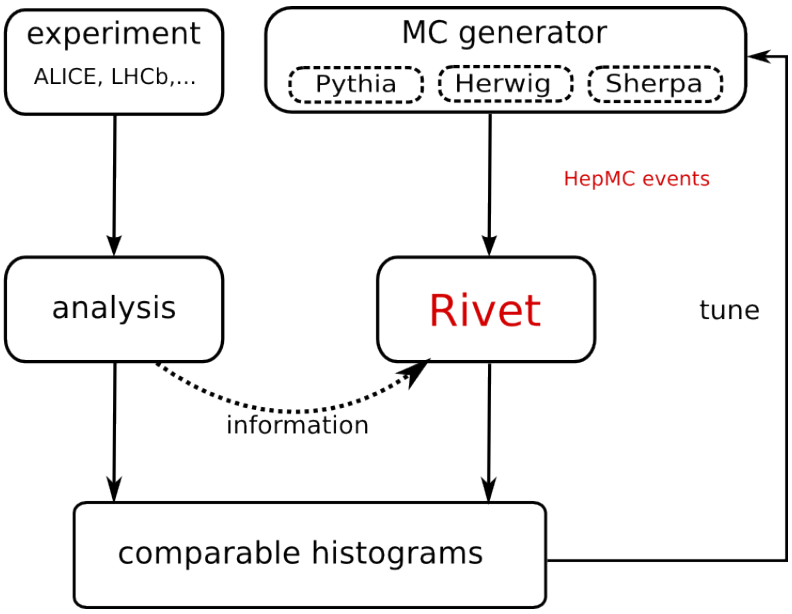


Figure 3.2: Schematic overview of the RIVET usage flow.

analysis has no influence on the generation process.

Especially theoreticians will use RIVET on a different level as they are more interested in the processes of the MC generator itself. Being interested only in the tuning of MC generators, one can use the already implemented analyses and concentrate on the parameters of the generator. Most of the rivetised analyses, or at least their output can be found on [20]. It shows already a large variety of analyses and is still growing, due to the fact, that newly rivetised analysis can easily be uploaded and integrated into the framework.

Lastly RIVET is the interface between the theoreticians and experimentalists. It is a way of smoothing the gap between the experimentalist on the one, who is more or less mainly interested in the analysis of data and the construction of new parts of the detector, and the theoretician on the other hand, who is more interested in fundamental processes and thus the models which are implemented. So that they have a common platform, where they can exchange their experiences.

3.2 Structure of a rivetised analysis

Writing a RIVET analysis is easier than one may think. The structure of the RIVET framework is implemented in a way, that one has not to touch the RIVET code itself for writing an analysis. The analysis is implemented as a class, which will be called during the running of the program. The RIVET analysis is divided into three parts:

- an initialisation process
- the event analysis

- and a finalising routine.

In the initialisation process, the histograms, datapointsets, profiles, and so on, are created, *i.e.* the binning will be set. This can be done ether by hand or automatically, where the former option requires an array with the bin edges or the range of the histogram and the number of bins, while for latter one, the analysis data has to provided in a specific format. The first option can be used to study the output of MC generators, no experimental data to compare to will be available here. However the initial purpose of Rivet is to maintain MC generator output in a comparable format to experimental analysis data, so the latter option is used more frequently. Rivet will automatically take the same binning like the provided analysis data. It is also necessary, to chose the particle type of interest, as the projections have to be specified here.

The second method is the analysing method. As expected, the main part of the analysis happens here. The projections, which were chosen in the previous part are used to identify particles due to certain group characteristics. Some of these groups are:

final state every final state particle

charged final state like final state, but charge is required (no π^0, neutron, ...)

unstable final state every physical but decayed particle (no gluons)

Particles in agreement with the chosen projection will be further processed. They are sorted out due to their type, as usually only the properties of a certain particle is of interest. After filtering all particles of an event to a small set, or maybe none, the kinematic properties of these particles can be calculated with already implemented methods. The information then can be stored into histograms. After all events have been analysed the histograms are normalized in the finalising method. In this method, it is also possible to calculate new observables from the existing histograms (*e.g.* calculate cross sections and integrate distributions). The output will be stored into a histogram that can be used to create plots.

4 Monte Carlo analyses of J/ψ production

4.1 Dependence on rapidity and transverse momentum

The corresponding published analysis is [3]. The analysis code can be found in B. The main purpose of this analysis is to measure the p_T spectrum of the inclusive J/ψproduction in mid ($|y| < 0.9$) and forward ($2.5 < y < 4$) rapidity as well as the p_T-integrated rapidity distribution of the inclusive J/ψ in pp collisions at $\sqrt{s} = 7$ TeV. For information on the contributing particles to the *inclusive* signal see 4.1.

The experimental analysis is based on detector data of the Inner Tracking System (ITS), the Time Projection Chamber (TPC) and the muon spectrometer. At mid rapidity ($|y| < 0.9$), the J/ψ is reconstructed via its decay into a e^+e^- pair (B.R.= $(5.94 \pm 0.06)\%$, [22]). This is then detected in the ITS and the TPC. At forward rapidity ($2,5 < y < 4$) the muon spectrometer measures a $\mu^+\mu^-$ pair coming form the decay J/$\psi \to \mu^+\mu^-$ (B.R.= $(5.93 \pm 0.06)\%$, [22]).

As we investigate not the direct J/ψ production but the inclusive one, J/ψ feed down from particles with a higher mass, mainly ψ(2S), χ_{c_0}, χ_{c_1}, χ_{c_2}, B-mesons, and contribute to the signal. The fractions can be found in Tab. 4.2.

	Particles	mass (GeV/c^2)	$c.\tau$ or width	decay channel	B.R.
prompt →	J/ψ ($c\bar{c}$)	3.097	92.9 keV/c^2	J/$\psi \to e^+e^-$	5.94 %
				J/$\psi \to \mu^+\mu^-$	5.93 %
	χ_{c_0} ($c\bar{c}$)	3.415	10.4 MeV/c^2	$\chi_{c_0} \to$ J/$\psi(1S)\gamma$	1.17 %
	χ_{c_1} ($c\bar{c}$)	3.511	0.86 MeV/c^2	$\chi_{c_1} \to$ J/$\psi(1S)\gamma$	34.4 %
	χ_{c_2} ($c\bar{c}$)	3.556	1.98 MeV/c^2	$\chi_{c_2} \to$ J/$\psi(1S)\gamma$	19.5 %
	ψ(2S) ($c\bar{c}$)	3.686	304 keV/c^2	ψ(2S) $\to$ J/ψ + *anything*	59.5 %
non-prompt	B^0 ($d\bar{b}$)	5.280	455 μm	$B^0, B^\pm, B_s^0 \to$ J/ψ + *anything*	1.16 %
	B^+ ($u\bar{b}$)	5.279	492 μm		
	B_s^0 ($s\bar{b}$)	5.367	449 μm		

Table 4.1: Main characteristics of particles contributing to the *inclusive* J/ψ signal [22]. The table includes mass, decay length or resonance width, as well as considered decay channel and corresponding Branching Ratio (B.R.).

contribution to inclusive J/ψ		
prompt	direct J/ψ	$33-55\%$
	ψ(2S)	$10-15\%$
	χ_{c_0}, χ_{c_1}, χ_{c_2}	$25-35\%$
non-prompt	B-mesons	$10-15\%$

Table 4.2: Contribution to the inclusive J/ψ production with $p_{\mathrm{T}} > 0$ [23].

The processing of the RIVET analysis is as follows. At first the histograms are initialised with the same binning as in [3]. In the analysing method the transverse momentum p_{T} and the rapidity y is calculated for every particle which suites the conditions of the unstable final state projection. The particle will be identified and, in case of a J/ψ, stored into a histogram according do its rapidity. So we get the yield $\mathrm{d}N_{\mathrm{J}/\psi}/\mathrm{d}y$ as a function of rapidity y. To obtain the relative yield we have to divide by the total number of pp events N_{pp}. This leads to:

$$\text{relative yield: } \frac{1}{N_{\mathrm{pp}}} \cdot \frac{\mathrm{d}N_{\mathrm{J}/\psi}}{\mathrm{d}y} \tag{4.1}$$

At a second step only the J/ψ at mid or forward rapidity, each with his own histogram, are further processed. Again the particle will be stored into the specific histograms, this time according to its transverse momentum. By division through N_{pp} we obtain:

$$\text{relative yield: } \left(\frac{1}{N_{\mathrm{pp}}} \cdot \frac{\mathrm{d}N_{\mathrm{J}/\psi}}{\mathrm{d}p_{\mathrm{T}}}\right)_{\mathrm{mid/for}} \tag{4.2}$$

Here we also have to divide through the rapidity window $\mathrm{d}y$[1] which is $\Delta y = 1.8$ at mid and $\Delta y = 1.5$ at forward rapidity. We then obtain:

$$\text{relative yield: } \frac{1}{N_{\mathrm{pp}}} \cdot \frac{\mathrm{d}^2 N_{\mathrm{J}/\psi}}{\mathrm{d}p_{\mathrm{T}}\mathrm{d}y} \tag{4.3}$$

On a next step, the ancestors of the particles are checked and histogram are filled according to the mother particle of the J/ψ.

The division by N_{pp} and $\mathrm{d}y$ will take part in the finalising method. Here also the step from the relative yield to the differential and double differential cross section will be taken. As the number of charged particles and the cross section are connected

[1] dividing by Δy using dy is valid as long as the distribution in the interval does not increases or decreases to much.

processes	cross section
all soft	90.76 $mb(\pm 2.7\%)$
inelastic	$71,36$ $mb(\pm 2.7\%$
hard	0.32 $mb(\pm 2.7\%$

Table 4.3: Cross sections as used by PYTHIA 8 to simulate pp events. They refer to the run cards used during this thesis.

via the luminosity L, a detector dependent value to characterize its performance:

$$N_{\mathrm{J}/\psi} = L_{\mathrm{int}} \cdot \sigma_{\mathrm{J}/\psi} \quad \Rightarrow \quad \mathrm{d}N_{\mathrm{J}/\psi} = L_{\mathrm{int}} \cdot \mathrm{d}\sigma_{\mathrm{J}/\psi} \tag{4.4}$$

$$N_{\mathrm{pp}} = L_{\mathrm{int}} \cdot \sigma_{\mathrm{pp}} \tag{4.5}$$

As the integrated luminosity L_{int} is the same for the pp collisions and the production of J/ψ, merging EQS. (4.1),(4.4) and (4.5) yield:

$$\frac{\mathrm{d}\sigma_{\mathrm{J}/\psi}}{\mathrm{d}y} = \sigma_{\mathrm{pp}} \cdot \frac{1}{N_{\mathrm{pp}}} \frac{\mathrm{d}N_{\mathrm{J}/\psi}}{\mathrm{d}y} \tag{4.6}$$

So, multiplying the relative yield with the total pp cross section σ_{pp} leads to the differential cross section. The same formalism can be applied to get the double differential cross section.

The total number of pp events is accessible in RIVET. This accounts also for the pp cross section σ_{pp} which is used by the generator to generate the events. It depends on the simulated processes, *e.g.* inelastic interactions, hard scatterings ..., as can be seen in TAB. 4.3.

4.2 Dependence on charged particle multiplicity

The corresponding published analysis is [4]. Again the analysis code can be found in B. The aim of this analysis is to get the J/ψ as a function of the charged particle multiplicity, as can be seen in FIG. 4.1. This is a much more complex analysis than the previous one, due to the fact that not only the particle itself has to be measured, but also all other charged particles in the event are important in this analysis.

On the ordinate of FIG. 4.1, the relative yield of J/ψ in a certain rapidity window dy is normalized to the mean value of J/ψ in Minimum Bias events. In case of simulation, the normalization is due to the mean value of J/ψ produced for the simulated sample. The abscissa shows the number of charged particles, with $|\eta| < 1$, in the event the J/ψ is produced, normalized to the mean number of charged particles per event.

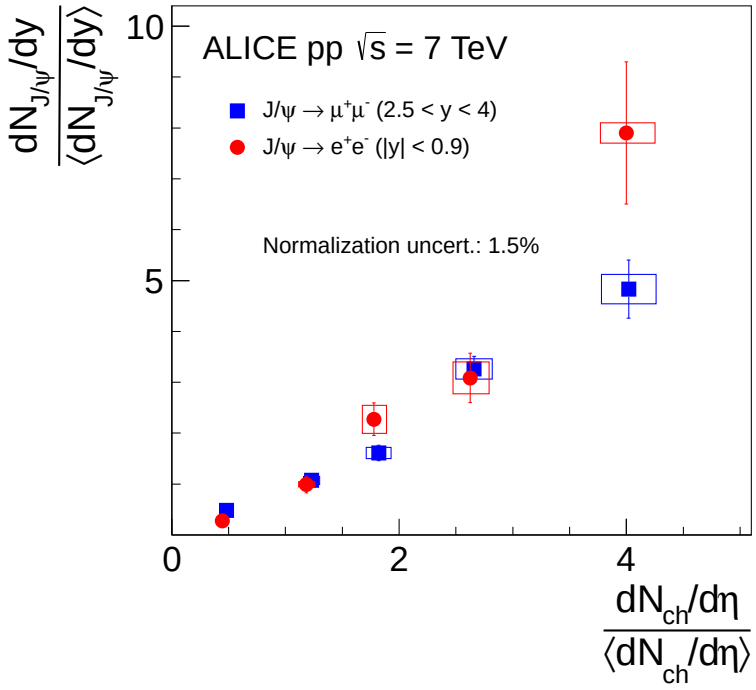


Figure 4.1: The yield of J/ψnormalized to the mean J/ψ yield in Minimum Bias events, as a function of charged particle multiplicity normalized to the mean charged particle multiplicity in Minimum Bias events[4].

So, to be more precise, Fig. 4.1 shows

$$\frac{\frac{1}{N_{\mathrm{pp,s}}}\left(\frac{\mathrm{d}N_{\mathrm{J}/\psi}}{\mathrm{d}y}\right)_i}{\left\langle\frac{1}{N_{\mathrm{pp,s}}}\cdot\frac{\mathrm{d}N_{\mathrm{J}/\psi}}{\mathrm{d}y}\right\rangle_{\mathrm{s}}} \tag{4.7}$$

Where the subscript s denotes the sample of simulated events for which a given set of processes is activated. As the mean number of J/ψ in minimum bias events can differ from the actual measuring, especially in MC generations these two quantities can differ. The subscript i represents the i^{th} bin of $\mathrm{d}N_{\mathrm{ch}}/\mathrm{d}\eta$

The experimental data shows a more or less linear increase of the yield towards higher multiplicities.

As the abscissa has to be normalized this time too, it is not possible to book comparable histograms a priori. In addition, the mean number of charged particles has also to be calculated from the MC output. In comparison to the first analysis, we now have to use two projections, since there are two groups of particles in which we are interested. The first one is the J/ψ which can be produced and decay in the production process, thus we use again the unstable final state projection. However in the case of the number of charged particles per event N_{ch}, we are just interested in

charged final state particles, which can be detected as final-state particles. Having two projections now, requires two scans of the event. The first one is done to calculate the number of charged particles per event, which will be stored into a histogram, and the second one is to check whether a J/ψ appears in the event. This then will be stored according to the number of charged particles per event.

4.3 Gaining statistics

One task that came up within this thesis is the question of statistics. One Million simulated pp-collisions, in the following referred to as runs, produce about 250 J/ψ at mid and 140 J/ψ at forward rapidity. The simplest way to gain more statistics would be to just simulate more events per run. This gives rise to the challenge of computing power, one million events need, depending on the run card, up to 28 hours computing time (on the ALICE farm in Heidelberg) to be processed. In the end we want to have something in the order of 25 million events. This means a run over 30 days, with hopefully no server restart, glitch or high load from other users. A solution is the parallelisation of the generation.

Parallelisation gives rise to a new issue. Since Eq. (4.1), the relative yield is already normalised to the number of events per run. The output histograms of different runs can not be merged[2]. As the total number of pp collisions N_{pp} is the sum over all processed events:

$$N_{\mathrm{pp}} = \sum_{runs} N_{\mathrm{pp},i} \tag{4.8}$$

where i denotes a certain run. So the analysis program was changed to have now:

- no normalization
- bookkeep any information related to normalisation into a `txt`-file

In the end, these information are stored into different files, which will be read by a Root-macro. This macro takes the required information, *i.e.* the histograms and the information needed for normalisation, merges the histograms and creates the plots.

4.4 Processing of events

As Pythia 8 is a highly-developed MC generator with a lot of implemented processes, it requires some choices from the user on how to simulate particle collisions. These choices are stored in a so called *run card*, which is provided as an input to

[2]merging is not possible in Rivet 1.8 but is foreseen in Rivet 2.1

the generator. There are default settings which will be used in case nothing is provided to the generator. However, at least the beam parameters should be specified, *i.e.* beam particles, beam energies. On top of that the user us free to decide out of a catalogue of a lot of processes, which ones he want to enable. There are a lot more processes implemented, than QCD processes, *e.g.* electroweak processes, but also processes beyond the standard model like SUSY processes. Another possibility is, to forbid certain decays so that unstable particles will not decay and become stable ones.

In the following, two *run cards* will be discussed. One is focussing on inelastic soft QCD processes ,C, and the other on hard QCD processes ,C. Although in the real collisions both kinds of processes will have a certain impact, they should not be mixed in the simulation due to the fact that this likely leads to double-counting. As J/ψ are mainly created in QCD processes, the other processes will not be implemented.

Now, what has to be done to start one job?

- create a `fifo`
- start the PYTHIA simulation. This means reading the runcard and redirect the output as HepMC to the created `fifo`.
- start RIVET and read the generators output. Store the output of Rivet in a certain file directory, following a given structure, so that the information needed to merge the histograms can be accessed systematically.

As this requires a few lines of code to be typed into the shell, a bash-script, see B, is created and used for starting jobs. Since the computational load of one simulation is not negligible, the jobs will be distributed over different servers on the ALICE farm in Heidelberg.

5 Results

5.1 p_t- and rapidity-dependent J/ψ production

In general the Figs: 5.1 and 5.2 show, that in case of simulated inelastic soft processes, the production rate of J/ψ is about a factor of 2 higher than the ALICE measurement.

In case of enabled hard QCD processes, Fig. 5.3 and Fig. 5.4, the simulation is about two orders of magnitude lower than the values from the experiment. This is something we expect, since in the lower part of the momentum regime perturbation theory is not valid any longer and thus hard QCD processes are not expected to produce meaningful results. We expect however, that the simulation would deliver better results at higher momentum regimes, $p_T > 10$ GeV/c^2, which than can be compared to ATLAS [17] and CMS [24] measurements, which naturally can take data at higher transverse momenta, in case of ATLAS up to 100 GeV/c.

Comparing the values of Fig. 5.1 and Fig. 5.2 shows the expected lower cross section at forward rapidity. The ratio spreads over a range of over 40% at mid and just about 20% at forward rapidity, whereas this may be due to the bigger uncertainties at mid rapidity.

The differential cross section $d\sigma/dy$ as a function of rapidity, as can be seen in Fig. 5.5, shows the same picture as above. At soft QCD the cross section is too high, by a factor of 2 and at hard QCD the crossection is about 1 to 2 orders of magnitude smaller than the ALICE data.

According to Pythia 8, in the low p_T region, soft processes play an important role for J/ψ production and are the main contributors to the total yield, what was expected.

5.2 Contributions to the inclusive signal

inelastic soft QCD The contribution to the inclusive signal can be taken from the ratio plots of Fig. 5.1 and 5.4. For the inelastic soft QCD simulation at mid rapidity, Fig. 5.1, the data shows, that the $\psi(2S)$ production is underestimated, since it is expected to be between 10 and 15%, Tab. 4.2. Whereas the signal from χ_{c_J} seems slightly too high with 40 to 45%.

The non-prompt component is at low p_T of the order 10% at rises to $15-20\%$ at 6 GeV/c. This linear rise is expected and willl go on to very high p_T, as

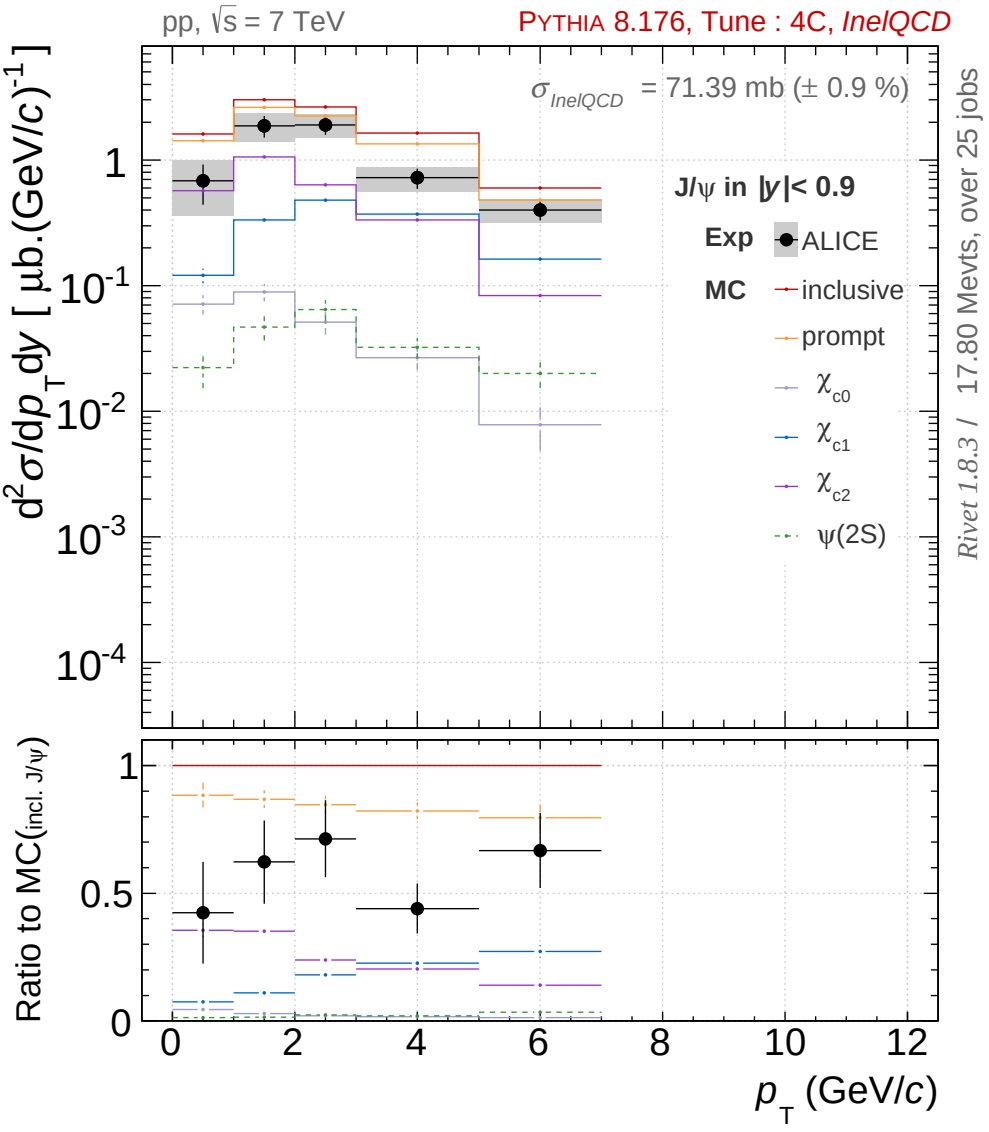


Figure 5.1: Double differential cross section $\mathrm{d}^2\sigma/(\mathrm{d}p_\mathrm{T}\mathrm{d}y)$ as a function of p_T at **mid rapidity** $|y| < 0.9$. Simulated 25 million pp events with PYTHIA 8 (tune 4C), only inel QCD processes enabled.

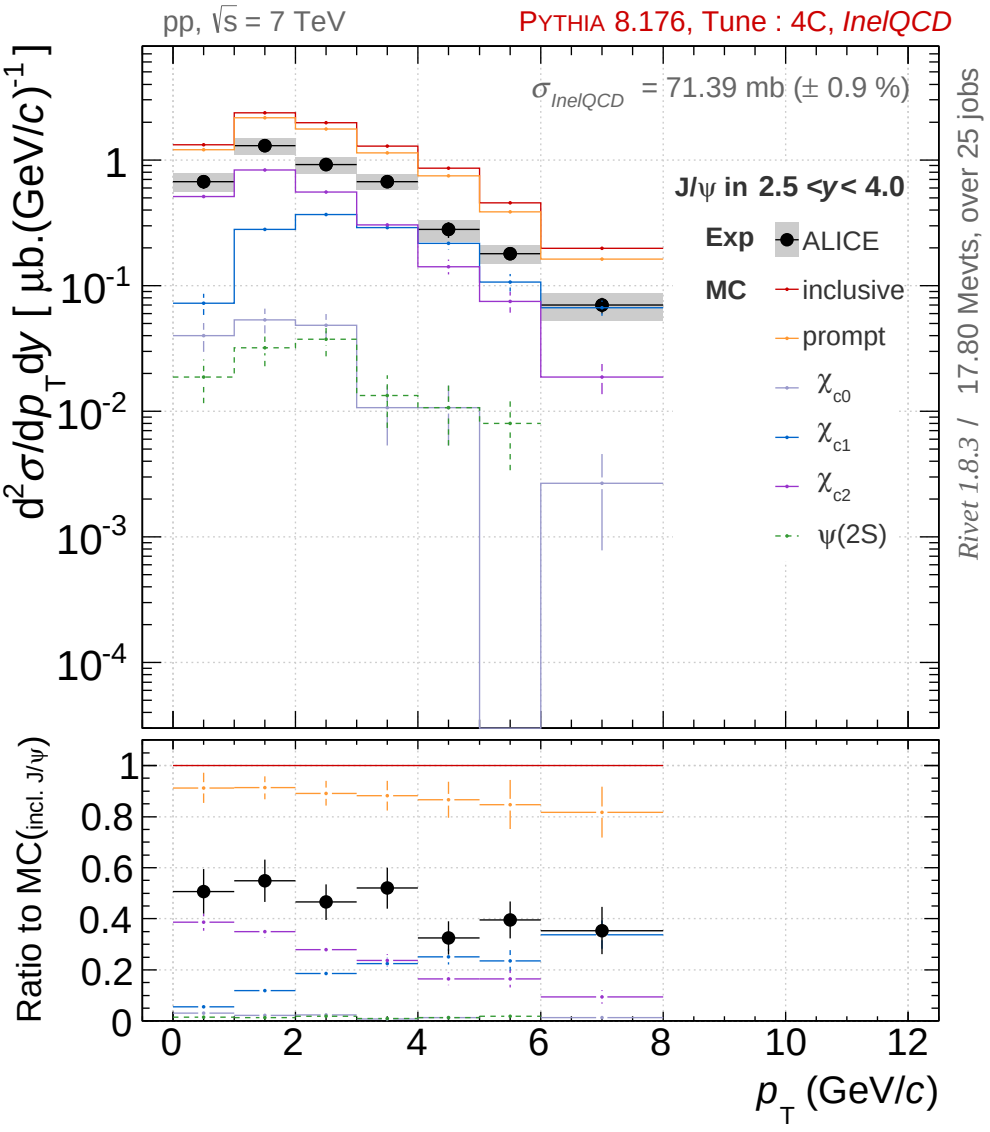


Figure 5.2: Double differential cross section $\mathrm{d}^2\sigma/(\mathrm{d}p_\mathrm{T}\mathrm{d}y)$ as a function of p_T at **forward rapidity** $2.5 < y < 0.9$. Simulated 25 million pp events with PYTHIA 8 (tune 4C), only inel QCD processes enabled.

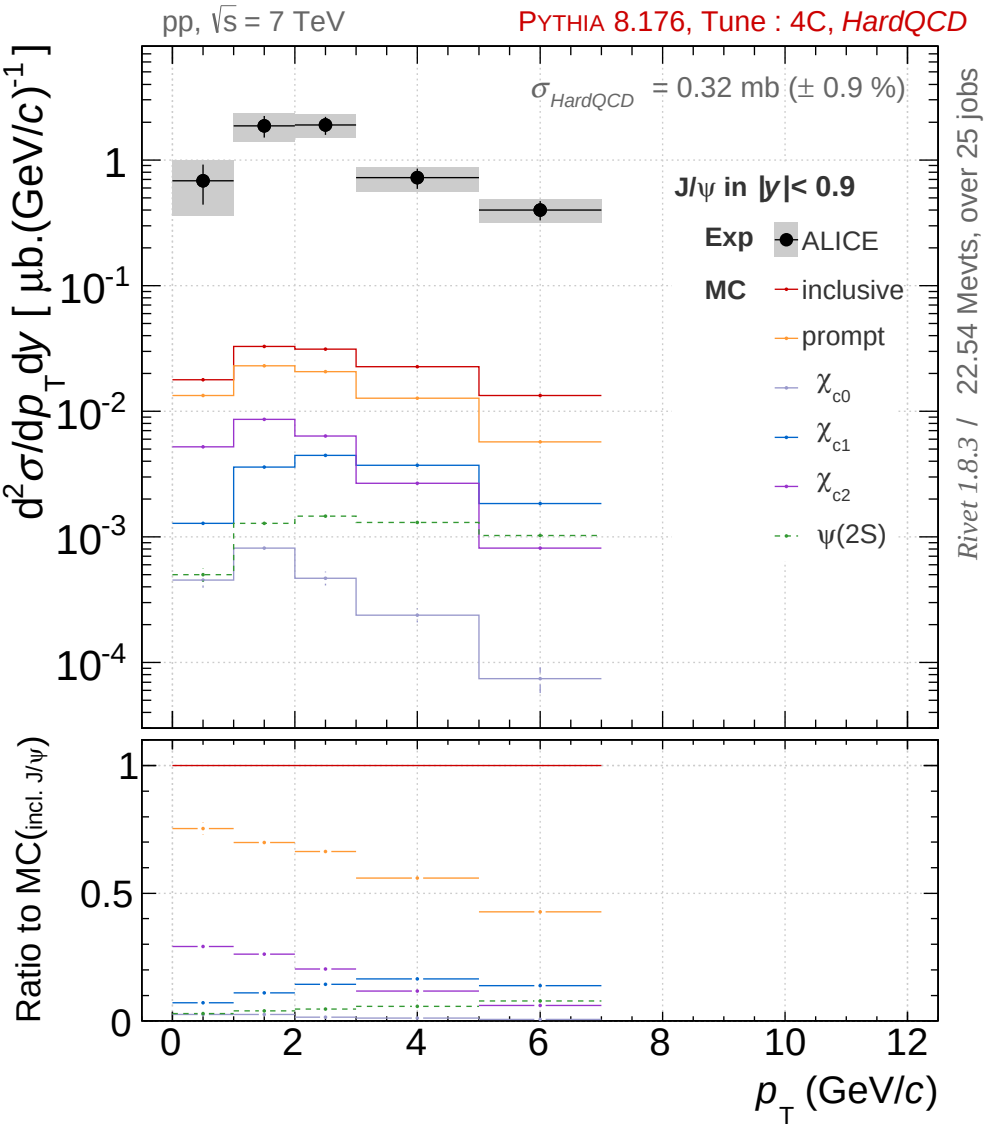


Figure 5.3: Double differential cross section $\mathrm{d}^2\sigma/(\mathrm{d}p_\mathrm{T}\mathrm{d}y)$ as a function of p_T at **mid rapidity** $|y| < 0.9$. Simulated 25 million pp events with PYTHIA 8 (tune 4C), only hard QCD processes enabled.

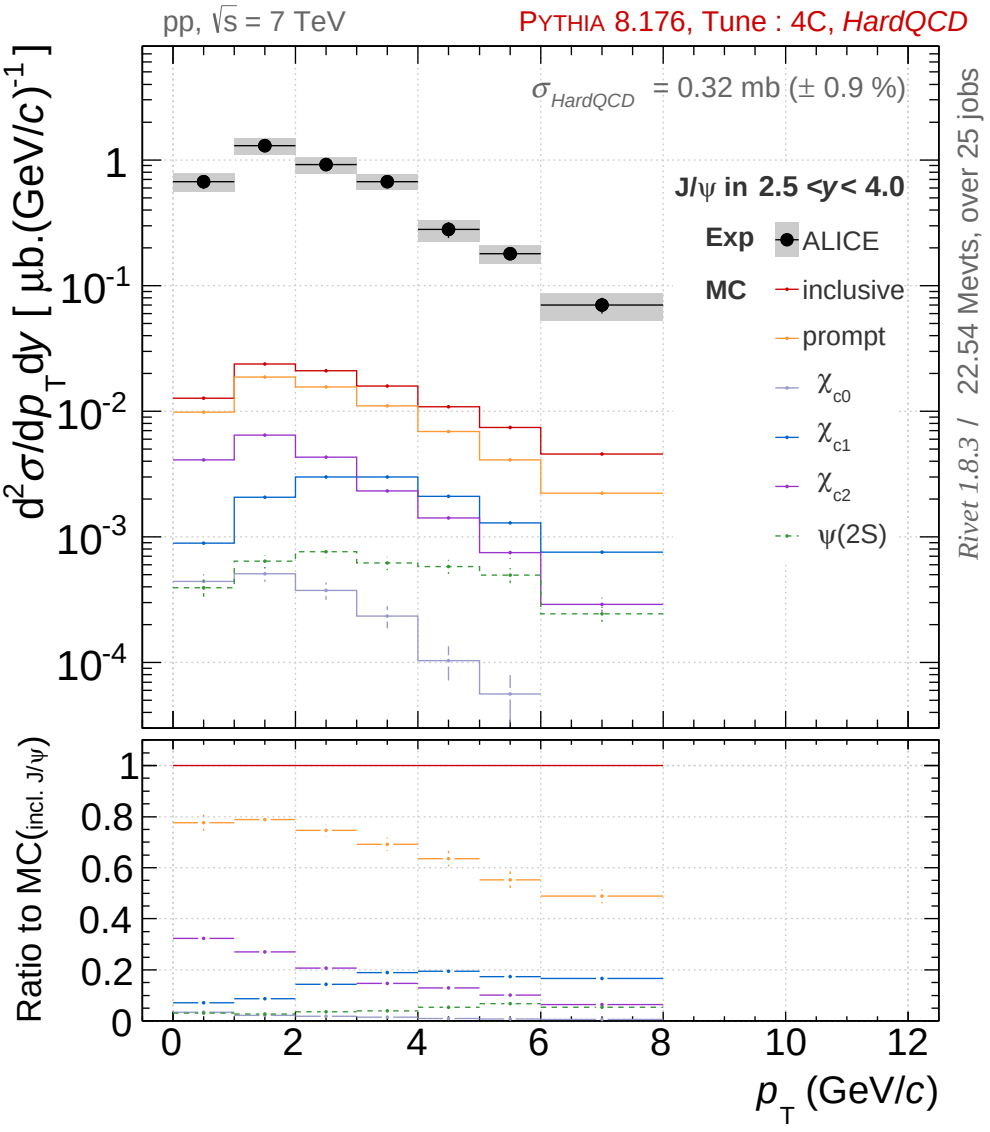


Figure 5.4: Double differential cross section $\mathrm{d}^2\sigma/(\mathrm{d}p_\mathrm{T}\mathrm{d}y)$ as a function of p_T at **forward rapidity** $2.5 < y < 0.9$. Simulated 25 million pp events with PYTHIA 8 (tune 4C), only hard QCD processes enabled.

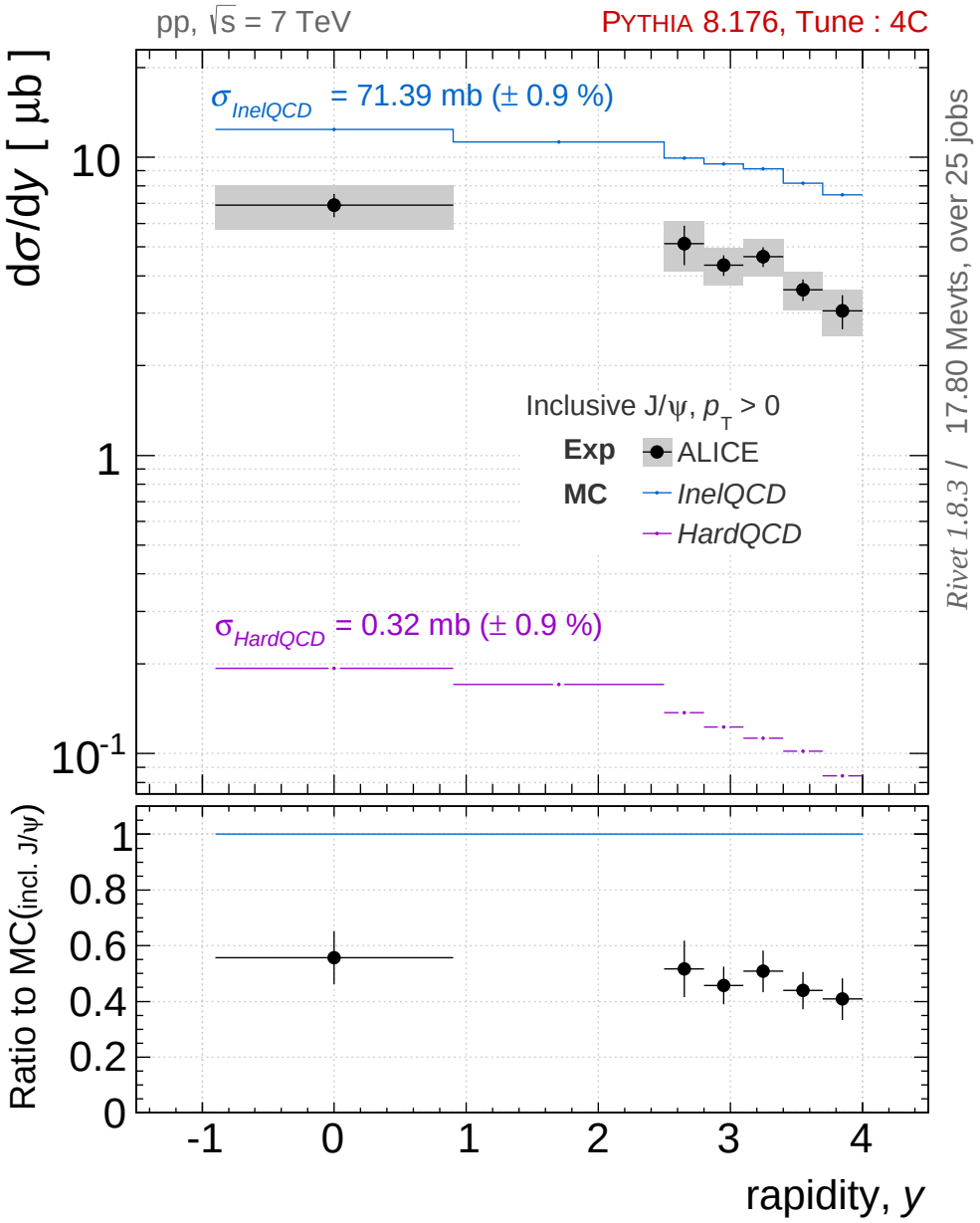


Figure 5.5: Differetial cross section dσ/dy as a fuction of rapidity y. Simulated 25 million pp events with PYTHIA 8 (tune 4C).

measured in LHCb [25] and CDF [26].

hard QCD In the hard QCD simulation however, Fig. 5.4, the contribution from B-mesons is too high and increases too fast. This is an indication, that hard QCD processes rather create b quarks than c quarks at high transverse momenta.

5.3 Multiplicity dependence of J/ψ production

In Tab. 5.1 the mean charged particle density $\langle dN_{ch}/d\eta \rangle$ is shown as calculated in the simulation. For the hard QCD simulation, the value is too high, while the inelastic QCD simulation the describes the measured value within the uncertainties.

The distributions themselves show different pictures. The hard QCD simulation, Fig. 5.8 , shows the N_{ch} distribution of hard QCD events and is, as expected, significantly different from the minimum bias data.

In case of the inelastic soft QCD simulation, Fig. 5.6, the distribution at low charged particle densities ($dN_{ch}/d\eta < 35$) is described very well, as already indicated in Fig. 2.4*(d)*. However the inelastic soft QCD simulation underestimates the probability at high charged particle densities which can be seen also in Fig. 2.4*(c)*.

ALICE	6.01 ± 0.01
Pythia 8 soft QCD	4.59 ± 2.02
Pythia 8 inel QCD	6.28 ± 1.10
Pythia 8 hard QCD	18.83 ± 1.06

Table 5.1: Comparing $\langle dN_{ch}/d\eta \rangle$ from different measurements. ALICE data taken from [4].

The J/ψ as a function of charged particle density, as simulated in soft QCD Fig. 5.7, can be divided into three parts. The first part until $dN_{ch}/d\eta \approx 6-7$ shows a more or less linear increase. Until $dN_{ch}/d\eta \approx 12-13$ we can see a plateau, and in the last part, $dN_{ch}/d\eta > 12-13$ we can see a linear decrease towards high charged particle densities.

By dividing these boundary values through the $\langle dN_{ch}/d\eta \rangle_{softQCD}$ of Tab. 5.1 we obtain the charged particle multiplicity and can compare the shape of the distribution to Fig. 4.1. In mid rapidity we see a linear increase in the data until $dN_{ch}/d\eta \approx 1 \times \langle dN_{ch}/d\eta \rangle_{softQCD}$. After that we have the plateau until the decreasing part, which begins at $dN_{ch}/d\eta > 2 \times \langle dN_{ch}/d\eta \rangle_{softQCD}$. However, the ALICE data in Fig. 4.1 shows a different picture. Here the linear increase goes at least until $dN_{ch}/d\eta \approx 4 \times \langle dN_{ch}/d\eta \rangle_{softQCD}$.

The simulated data shows the same behaviour although the two curves do not superimpose within statistical uncertainties. This is something one might suggest while regarding Fig. 4.1.

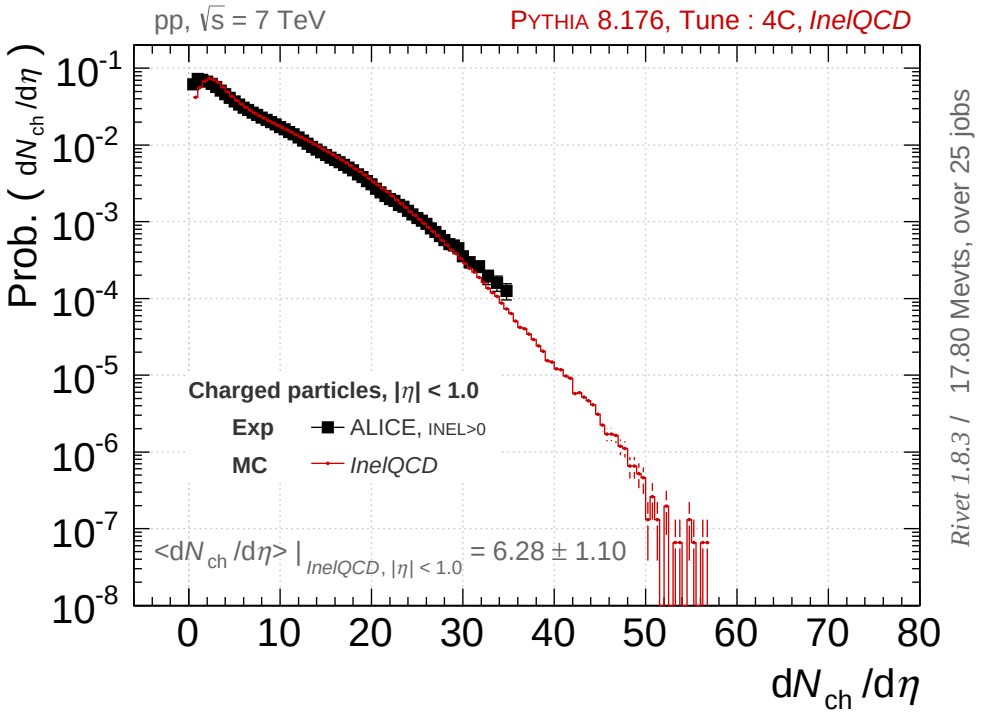


Figure 5.6: Prob. for $\mathrm{d}N_{\mathrm{ch}}/\mathrm{d}\eta$ in MC soft QCD simulation compared to ALICE [4].

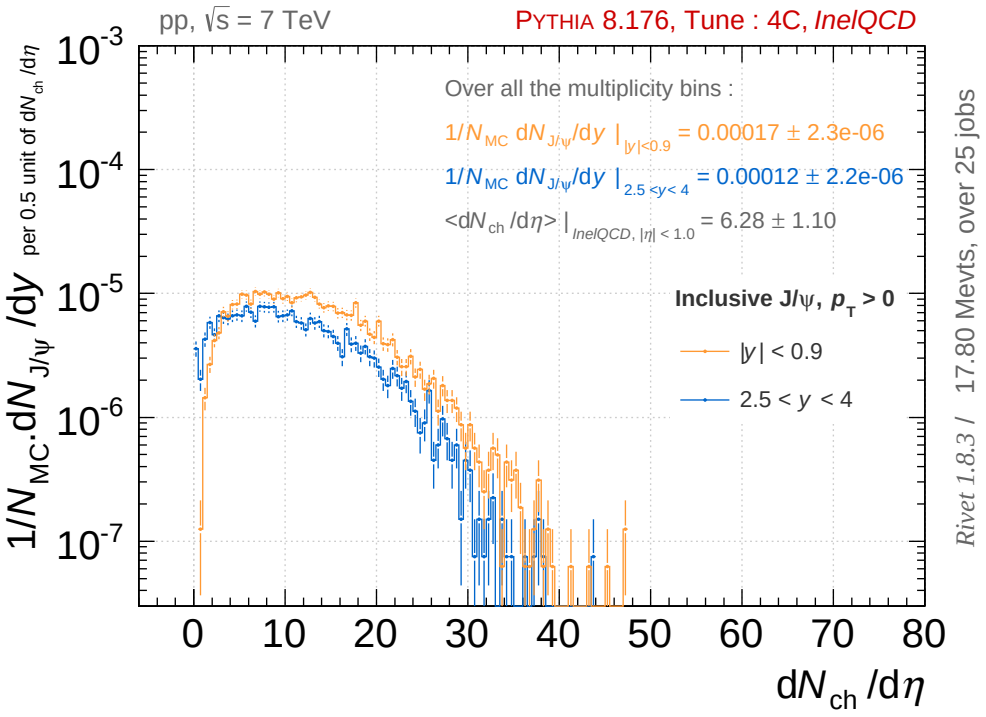


Figure 5.7: Yield as a function of charged particle density for soft QCD, at mid and forward rapidiy compared to ALICE [4].

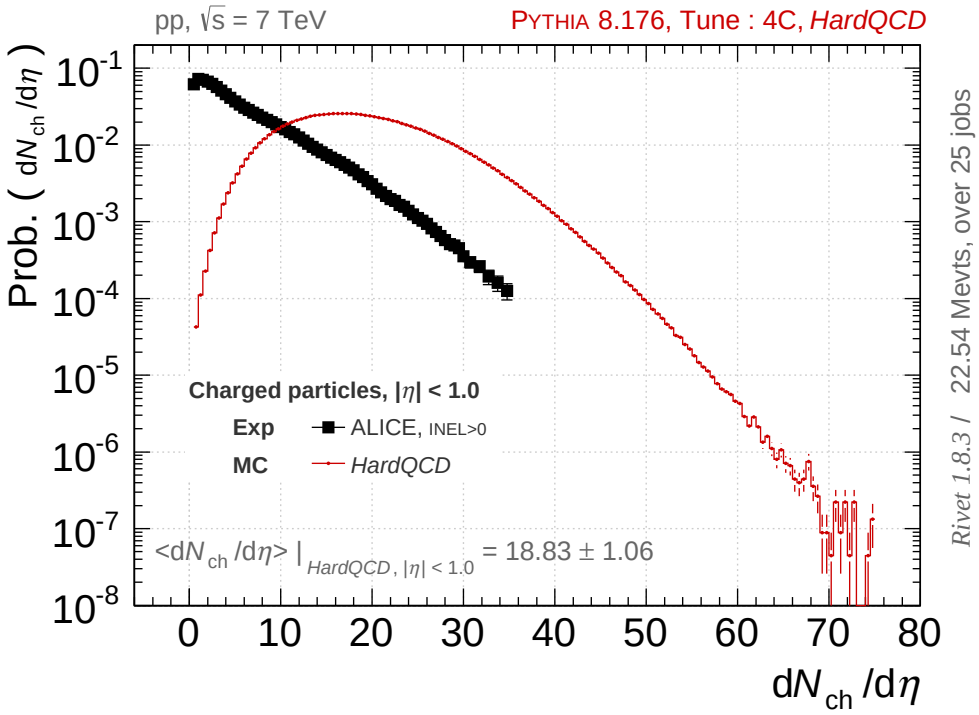


Figure 5.8: Prob. for $\mathrm{d}N_{\mathrm{ch}}/\mathrm{d}\eta$ in MC hard QCD simulation compared to ALICE [4].

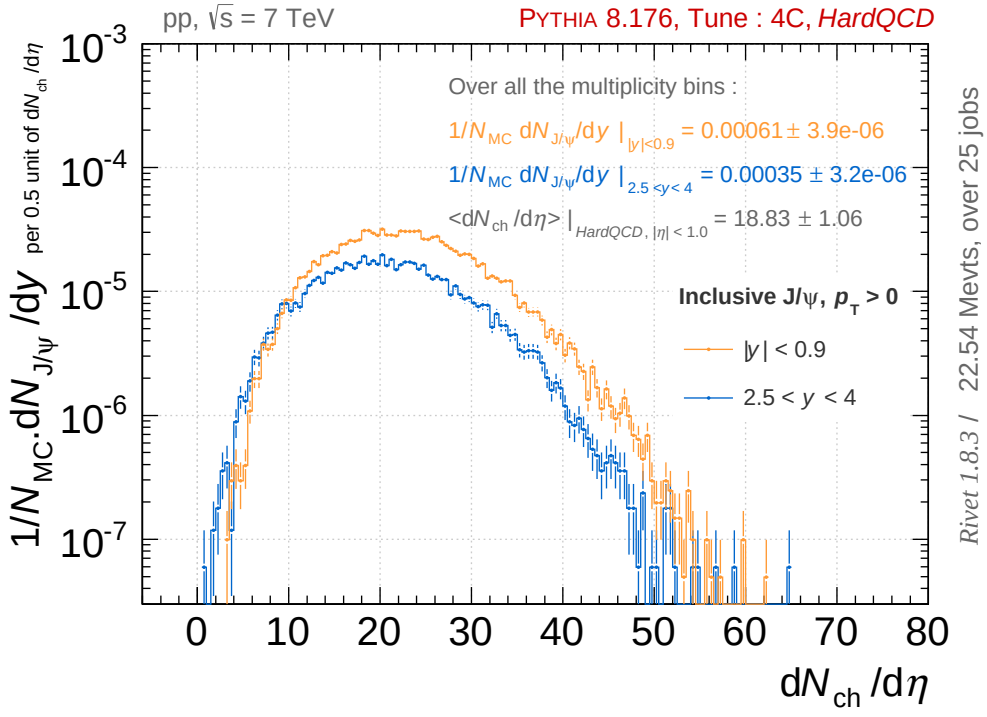


Figure 5.9: Yield as a function of charged particle density for hard QCD, at mid and forward rapidiy compared to ALICE [4].

The hard QCD simulation, Fig. 5.9 at mid rapidity, can be divided into two parts. A linear increase until $\mathrm{d}N_{\mathrm{ch}}/\mathrm{d}\eta \approx 20$. And afterwords a linear decrease. Translating the changing point into multiplicity would lead to $\mathrm{d}N_{\mathrm{ch}}/\mathrm{d}\eta \approx 1 \times \langle \mathrm{d}N_{\mathrm{ch}}/\mathrm{d}\eta \rangle_{hardQCD}$ which is not covered by the data at all. However dividing by the mean value of the ALICE measurement would lead to $\mathrm{d}N_{\mathrm{ch}}/\mathrm{d}\eta \approx 3 \times \langle \mathrm{d}N_{\mathrm{ch}}/\mathrm{d}\eta \rangle_{ALICE}$. Compared to the ALICE data about the charged particle multiplicity, which can be found in Fig. 4.1, the hard QCD result is not as well suited to explain this behaviour as the inelastic soft QCD result. Here the results at mid and forward rapidity show also a similar trend.

6 Summery and Outlook

The good agreement of Pythia 8 simulation to early data from the LHC at 7 TeV could not be confirmed for the J/ψ production. However this was something we expected, as the J/ψ production in general is a difficult topic to investigate.

The shape and tendency of our inelastic soft QCD simulation are a good start, which points int the right direction. Also the charged particle density can be described very well. However the J/ψ production as a function of charged particle multiplicity could only be confirmed at very low charged particle multiplicities. Towards higher Charged particle multiplicities, ALICE and MC data show different results. To further investigate this discrepancies there are tow possibilities.

- Extend the ALICE analysis to higher multiplicities, that the further behaviour can be investigated.
- Use different MC generators, to see whether this behaviour is specific for Pythia 8.

A new generator, which could be used, to simulate these events, would be Sherpa. As it was intended to use it in this thesis to. The program is already installed on the farm at the Physikalisches Institut in Heidelberg and configured in a way, that Rivet is able to analyse the Output.

Another possibility would be to further investigate the behaviour of Pythia 8 due to the change of parameters and processes. As over two hundred processes, which can individually be turned on and off, are already implemented.

Appendix

A Lists

A.1 List of Figures

A.2 List of Tables

B Analysis code

$p_{\mathbf{t}}$- and y-dependence

```
     // -*- C++ -*-
#include "Rivet/Rivet.hh"
#include "Rivet/Analysis.hh"
#include "Rivet/RivetAIDA.hh"
#include "Rivet/Tools/Logging.hh"
#include "Rivet/Projections/UnstableFinalState.hh"
#include <math.h>
#include <iostream>

namespace Rivet {

  class my_jpsi_analysis : public Analysis {
  public:
    my_jpsi_analysis()
      : Analysis("my_jpsi_analysis")
    {    }


  public:
    void init() {
UnstableFinalState ufs( -8, 8, 0.0*GeV);
addProjection(ufs,"UFS");
_h_dNjpsi_pt_mid  = bookHistogram1D(1,1,1);
_h_dNjpsi_pt_for  = bookHistogram1D(2,1,1);
_h_dNjpsi_y_tot = bookHistogram1D(3,1,1);

_h_dN_nonb_mid = bookHistogram1D(1,1,2);
_h_dN_chi0_mid = bookHistogram1D(1,1,3);
_h_dN_chi1_mid = bookHistogram1D(1,1,4);
_h_dN_chi2_mid = bookHistogram1D(1,1,5);
_h_dN_psi_mid = bookHistogram1D(1,1,6);

_h_dN_nonb_for = bookHistogram1D(2,1,2);
_h_dN_chi0_for = bookHistogram1D(2,1,3);
_h_dN_chi1_for = bookHistogram1D(2,1,4);
```

```
_h_dN_chi2_for = bookHistogram1D(2,1,5);
_h_dN_psi_for = bookHistogram1D(2,1,6);
    }
    void analyze(const Event& event) {
    const double weight = event.weight();
const UnstableFinalState& unstable = applyProjection<UnstableFinalState>(event, "U

foreach (const Particle& p, unstable.particles()) {
const double pT = p.momentum().pT();
const double y = p.momentum().rapidity();
if (p.pdgId() == 443) {
//pdgId 443-> J/psi
_h_dNjpsi_y_tot -> fill(y,weight);
if (fabs(y)<0.9) {
_h_dNjpsi_pt_mid->fill(pT,weight);
if ( ! (p.hasAncestor(511)  ||  p.hasAncestor(-511) || // B0 and cc
p.hasAncestor(531) || p.hasAncestor(-531)  || // Bs0 and cc
p.hasAncestor(521) || p.hasAncestor(-521) || //B+-
p.hasAncestor(541) || p.hasAncestor(-541))) {    //Bc+-
_h_dN_nonb_mid -> fill(pT,weight);
}
// Some higher resonances of B mesons can also decay into
// J/psi but they will mostly decay via the "stable" B mesons : B0, B+, Bs.
if (p.hasAncestor(10441)) _h_dN_chi0_mid->fill(pT,weight);
if (p.hasAncestor(20443)) _h_dN_chi1_mid->fill(pT,weight);
if (p.hasAncestor(445)) _h_dN_chi2_mid->fill(pT,weight);
if (p.hasAncestor(100443)) _h_dN_psi_mid->fill(pT,weight);
//higher resonances of ccbar can also decay into J/psi but with very low B.R. (e.g
}
if (y>2.5 && y<4.) {
_h_dNjpsi_pt_for->fill(pT,weight);
if ( ! (p.hasAncestor(511)  ||  p.hasAncestor(-511) || // B0 and cc
p.hasAncestor(531) || p.hasAncestor(-531)  || // Bs0 and cc
p.hasAncestor(521) || p.hasAncestor(-521) || //B+-
p.hasAncestor(541) || p.hasAncestor(-541))) {    //Bc+-
_h_dN_nonb_for -> fill(pT,weight);
}
if (p.hasAncestor(10441)) _h_dN_chi0_for->fill(pT,weight);
if (p.hasAncestor(20443)) _h_dN_chi1_for->fill(pT,weight);
if (p.hasAncestor(445)) _h_dN_chi2_for->fill(pT,weight);
if (p.hasAncestor(100443)) _h_dN_psi_for->fill(pT,weight);
}
}
}
```

```
}

    void finalize() {

scale(_h_dNjpsi_pt_mid, 1./(1.8));
scale(_h_dNjpsi_pt_for, 1./(1.5));
scale(_h_dNjpsi_y_tot,1.);

scale(_h_dN_nonb_mid,    1/(1.8));
scale(_h_dN_chi0_mid, 1/(1.8));
scale(_h_dN_chi1_mid, 1/(1.8));
scale(_h_dN_chi2_mid, 1/(1.8));
scale(_h_dN_psi_mid,  1/(1.8));

scale(_h_dN_nonb_for,    1/(1.5));
scale(_h_dN_chi0_for, 1/(1.5));
scale(_h_dN_chi1_for, 1/(1.5));
scale(_h_dN_chi2_for, 1/(1.5));
scale(_h_dN_psi_for,  1/(1.5));

std::cout << "------------------ my_jpsi_analysis ------------------" << endl;
std::cout.precision(12);
std::cout << "my_jpsi_analysis / Crosssection(pT): " << crossSection() <<endl;
std::cout << "my_jpsi_analysis / Sum_of_weights(pT): " << sumOfWeights() <<endl;
}

  private:
    AIDA::IHistogram1D *_h_dNjpsi_pt_mid;
AIDA::IHistogram1D *_h_dNjpsi_pt_for;
AIDA::IHistogram1D *_h_dNjpsi_y_tot;
AIDA::IHistogram1D *_h_dN_nonb_mid;
AIDA::IHistogram1D *_h_dN_chi0_mid;
AIDA::IHistogram1D *_h_dN_chi1_mid;
AIDA::IHistogram1D *_h_dN_chi2_mid;
AIDA::IHistogram1D *_h_dN_psi_mid;
AIDA::IHistogram1D *_h_dN_nonb_for;
AIDA::IHistogram1D *_h_dN_chi0_for;
AIDA::IHistogram1D *_h_dN_chi1_for;
AIDA::IHistogram1D *_h_dN_chi2_for;
AIDA::IHistogram1D *_h_dN_psi_for;
  };
  // The hook for the plugin system
  DECLARE_RIVET_PLUGIN(my_jpsi_analysis);
}
```

Multiplicity dependence

```
     // -*- C++ -*-
#include "Rivet/Rivet.hh"
#include "Rivet/Analysis.hh"
#include "Rivet/RivetAIDA.hh"
#include "Rivet/Tools/Logging.hh"
#include "Rivet/Projections/ChargedFinalState.hh"
#include "Rivet/Projections/UnstableFinalState.hh"
#include "Rivet/Tools/ParticleIdUtils.hh"

#include <math.h>
#include <iostream>

namespace Rivet {
  class my_jpsi_mult : public Analysis {
  public:

    my_jpsi_mult()
      : Analysis("my_jpsi_mult")
    {    }
  public:
    void init() {
const ChargedFinalState cfs(-1.,1.,0.); //pT cut?
    addProjection(cfs,"cfs");
const UnstableFinalState ufs(-8.,8.,0.);
addProjection(ufs,"ufs");
_hist_mult_mid = bookHistogram1D("JpsiMultDepdce_mid", 150, 0,75);
_hist_mult_for = bookHistogram1D("JpsiMultDepdce_for", 150, 0,75);
_hist_Nch  = bookHistogram1D("NchDistribution",150,0,75);
//_dps_mult  = bookDataPointSet("multiplicity","mult","dNch","dNjpsi");
Njpsi_total_mid = 0;
Njpsi_total_mid_NoWeight = 0;
Njpsi_total_for = 0;
Njpsi_total_for_NoWeight = 0;
    }

    /// Perform the per-event analysis
    void analyze(const Event& event) {
    const double weight = event.weight();
const ChargedFinalState& charged = applyProjection<ChargedFinalState>(event,"cfs")
```

```
const UnstableFinalState& unstable = applyProjection<UnstableFinalState>(event,"ufs")
int   Nch_in_evt = 0;
float PseudoRapCoverage = 2*1.0; // for the Nch assessment
foreach ( const Particle& p, charged.particles()) {
const double eta = p.momentum().eta();
if ( PID::isHadron(p.pdgId()) || PID::isLepton(p.pdgId())  )
if ( fabs(eta) <= 1. ) Nch_in_evt++ ;
}

_hist_Nch -> fill(Nch_in_evt/PseudoRapCoverage, weight);
   foreach (const Particle& p, unstable.particles()) {
const double y = p.momentum().rapidity();
if( p.pdgId() == 443) {
if ( fabs(y) <= .9 ){
_hist_mult_mid -> fill( Nch_in_evt/PseudoRapCoverage, weight );
Njpsi_total_mid += weight;
Njpsi_total_mid_NoWeight++;
}// end mid rap
if ( y > 2.5 && y < 4. ){
_hist_mult_for -> fill( Nch_in_evt/PseudoRapCoverage, weight );
Njpsi_total_for += weight;
Njpsi_total_for_NoWeight++;
}// end forward rap
}// end if J/psi
}// end loop unstable part.
    }

    void finalize() {
scale(_hist_mult_mid, 1/(1.8)); // division by the rapidity coverage for J/psi
scale(_hist_mult_for, 1/(1.5)); // division by the rapidity coverage for J/psi
std::cout<< "---------------------- my_jpsi_mult ----------------------" << endl;
std::cout.precision(12);
std::cout<< "my_jpsi_mult / Sum_of_weights(mult): "       <<  sumOfWeights()
std::cout<< "my_jpsi_mult / Njpsi_total_mid: "            <<  Njpsi_total_mid
std::cout<< "my_jpsi_mult / Njpsi_total_mid_NoWeight: "   <<  Njpsi_total_mid_NoWeigh
std::cout<< "my_jpsi_mult / Njpsi_total_for: "            <<  Njpsi_total_for
std::cout<< "my_jpsi_mult / Njpsi_total_for_NoWeight: "   <<  Njpsi_total_for_NoWeigh
    }
  private:
double Njpsi_total_mid;
double Njpsi_total_mid_NoWeight;
double Njpsi_total_for;
double Njpsi_total_for_NoWeight;
AIDA::IHistogram1D *_hist_Nch;
```

```
AIDA::IHistogram1D *_hist_mult_mid;
AIDA::IHistogram1D *_hist_mult_for;
//AIDA::IDataPointSet *_dps_mult;
//AIDA::IDataPointSet * _h_data_mult;
  };
  // The hook for the plugin system
  DECLARE_RIVET_PLUGIN(my_jpsi_mult);
}
```

Bash script

```
     #!/bin/bash


# nohup bash Bash_SubmitSubjob_Rivet.sh pythia8 1 >/home/mbarra/output_Subjob1.txt

export generator=$1 # pythia8, sherpa, herwig++
export subjobID=$2  # subjob id to be chosen by the user
RIVPATH=$HOME/bachelor/Rivet/runs
PYTHPATH=$HOME/bachelor/Pythia8/run
if [[ "$generator" == "" || "$subjobID" == "" ]]
    then
        echo "No generator provided and/or no subjobID provided... exit !"
        exit
    fi

if [[ "$generator" != "pythia8" && "$generator" != "sherpa" && "$generator" != "he
    then
        echo "Wrong generator name. Only pythia8, sherpa, herwig++... exit !"
        exit
    fi

export StartingTime=‘date‘

source $HOME/bachelor/scripts/settotal.sh

mkdir $RIVPATH/output/${generator}-‘date +%m_%d‘-sub${subjobID}
mkfifo $RIVPATH/fifos/run${subjobID}.fifo

case $generator in
    "pythia8" )
```

```
$PYTHPATH/main42.exe $PYTHPATH/Tune4C-Inelastic.cmnd
$RIVPATH/fifos/run${subjobID}.fifo
1>$RIVPATH/gen_info/output-${generator}-‘date +%m_%d‘-${subjobID}.txt
2>$RIVPATH/err/outputErr-${generator}-‘date +%m_%d‘-${subjobID}.txt &

rivet -a my_jpsi_analysis -a my_jpsi_mult -H
$RIVPATH/output/${generator}-‘date +%m_%d‘-sub${subjobID}/out.aida
$RIVPATH/fifos/run${subjobID}.fifo
1> $RIVPATH/run_info/outputRivet-${generator}-‘date +%m_%d‘-${subjobID}.txt
2>$RIVPATH/err/outputRivetErr-${generator}-‘date +%m_%d‘-${subjobID}.txt
;;
    "sherpa" )
            echo "not implemented jet"
    ;;
esac

cd $RIVPATH/output/${generator}-‘date +%m_%d‘-sub${subjobID}
compare-histos out.aida
cd

rm -f $RIVPATH/fifos/run${subjobID}.fifo
export EndingTime=‘date‘

echo "Machine used for subjob -${subjobID}- :  ‘hostname‘"
echo "Subjob started at : ${StartingTime}"
echo "Subjob ended at   : ${EndingTime}"

exit
```

C Pythia parameter files

HardQCD

```
! File: Test-HardQCD-17july13.cmnd
! Test program for J/psi vs. mult
! Author : Antonin Maire
! -> all parameter choices that are physically
! non-sense can all be attributed to him :-)

! 1) General simulation settings
Main:numberOfEvents = 1000000         ! number of events to generate
Main:showChangedSettings = on        ! print changed flags/modes/parameters
Next:numberCount = 5000              ! for counter display, while processing

! 3) Beam parameter settings. Values below may be redundant with default ones.
Beams:idA = 2212                    ! first beam,  p = 2212, pbar = -2212
Beams:idB = 2212                    ! second beam, p = 2212, pbar = -2212
Beams:eCM = 7000.
Beams:eA = 3500.
Beams:eB = 3500.
Beams:pzA = 3500.
Beams:pzB = -3500.

! 4) Random seed settings
Random:setSeed = on
Random:seed = 0

! 5) Settings for hard-process generation internal to Pythia8.
! Warning 2: you must not mix processes
! from the SoftQCD and HardQCD process groups,
! since this is likely to lead to double-counting.
SoftQCD:minbias = off
HardQCD:all = on
PhaseSpace:pTHatMin = 10.            ! minimum pT in hard processes :
! from Peter Skands, everything below 20 GeV/c will still produce some results
but one can highly question if it makes any sense physically
! Default pTHatMin = 0
HardQCD:gg2ccbar = on
```

```
HardQCD:qqbar2ccbar = on
HardQCD:hardccbar = on
HardQCD:gg2bbbar = on
HardQCD:qqbar2bbbar = on
HardQCD:hardbbbar = on
Charmonium:all = on
SecondHard:generate = on
SecondHard:TwoJets = on
SecondHard:PhotonAndJet = on
SecondHard:Charmonium = on
SecondHard:Bottomonium = off
SecondHard:TwoBJets = on

! 6) Switch on/off some key components of the simulation, for comparisons.
PartonLevel:all = on              ! continue or stop after hard process
PartonLevel:MI = on               ! multiple interactions
PartonLevel:ISR = on              ! initial-state radiation
PartonLevel:FSR = on              ! final-state radiation
HadronLevel:Hadronize = on        ! hadronization
HadronLevel:Decay = on            ! decays

! 7) Decide if some heavy-flavour particles can decay or not
411:onMode = on     ! D mesons
421:onMode = on
431:onMode = on
413:onMode = on
423:onMode = on
433:onMode = on
511:onMode = on     ! B mesons
521:onMode = on
531:onMode = on
541:onMode = on
513:onMode = on
523:onMode = on
533:onMode = on
543:onMode = on
443:onMode = on     ! quarkonium
10441:onMode = on
20443:onMode = on
445:onMode = on
100443:onMode = on
553:onMode = on
100553:onMode = on
5122:onMode = on
```

```
! 8) Multiple interactions and impact parameter picture.
! Note: these values are illustrative only, not to be taken seriously.
#MultipleInteractions:pT0Ref = 2.
#MultipleInteractions:ecmRef = 1960.
#MultipleInteractions:ecmPow = 0.16
#MultipleInteractions:pTmin = 0.2
#MultipleInteractions:bProfile = 2
#MultipleInteractions:coreRadius = 0.4
#MultipleInteractions:coreFraction = 0.5
#MultipleInteractions:expPow = 1.

! 9) color reconnection
BeamRemnants:reconnectColours = on

! 10) Bremsstrahlung (?) in decays, a priori PHOTOS equivalent
ParticleDecays:allowPhotonRadiation = on
ParticleDecays:FSRinDecays = on

! 11) Tune choice
Tune:pp = 5           ! default Pythia8 tune = tune 4C = number 5
```

Inelastic soft QCD

```
# settings of Pythia 8 wrapper program
Main:numberOfEvents = 1000000      ! number of events to generate
Main:timesToShow = 0               ! show how far along run is this many times
Main:timesAllowErrors = 3          ! abort run after this many flawed events
Main:showChangedSettings = on      ! print changed flags/modes/parameters
Main:showChangedParticleData = on  ! print changed particle and decay data
Next:numberShowEvent = 0           ! suppress full listing of first events
Next:numberCount = 5000             ! for counter display, while processing

# random seed
Random:setSeed = on
Random:seed = 0

# Beam parameter settings.
Beams:idA = 2212               ! first beam, p = 2212, pbar = -2212
Beams:idB = 2212               ! second beam, p = 2212, pbar = -2212
Beams:eCM = 7000               ! CM energy of collision

# Minimum Bias process (as taken from one of pythia8 example)
```

```
# SoftQCD:minbias = on          ! minimum bias QCD processes
SoftQCD:inelastic = on          ! activate MinBias, SingleDiffr,
! DoubleDiffr, CentralDiffr = all except elastic

# Process setup: min-bias
# Use this for ordinary min-bias (assuming Rivet analysis
# correctly suppresses the diffractive contributions.)
# SoftQCD:all = on # this for min-bias incl diffraction

# Set cuts
# Use this for hard leading-jets in a certain pT window
PhaseSpace:pTHatMin = 0      # min pT
PhaseSpace:pTHatMax = 7000   # max pT

# Use this for hard leading-jets in a certain mHat window
PhaseSpace:mHatMin = 0      # min mHat
PhaseSpace:mHatMax = 7000   # max mHat

# Makes particles with c*tau > 10 mm stable:
ParticleDecays:limitTau0 = On
ParticleDecays:tau0Max = 10.0

# Tune setup:
Tune:pp = 5     ! 4C = default
```

D Bibliography

[1] S. Ting, *Experimental observation of a heavy particle J.*, *Phys. Rev. Lett.* **33** (1974) 1404–1406. dx.doi.org/10.1103/PhysRevLett.33.1404, DOI link.

[2] G. Abrams, D. Briggs, W. Chinowsky, G. Friedberg, C.E.and Goldhaber, et al., *The Discovery of a Second Narrow Resonance in e+ e- Annihilation.*, *Phys. Rev. Lett.* **33** (1974) 1453–1455. dx.doi.org/10.1103/PhysRevLett.33.1453, DOI link.

[3] ALICE Collaboration, *Rapidity and transverse momentum dependence of inclusive J/ψ production in pp collisions at $\sqrt{s}$ = 7 TeV*, *Phys. Lett. B* **704** (2011) 442–455. Erratum : Phys. Lett. B, 718 (2012) 692-698 = 10.1016/j.physletb.2012.10.060. arxiv.org/abs/1105.0380, DOI link.

[4] ALICE Collaboration, *J/ψ production as a function of charged particle multiplicity in pp collisions at $\sqrt{s}$ = 7 TeV*, *Phys. Lett. B* **712** (2012) 165–175. arxiv.org/abs/1202.2816, DOI link.

[5] D. Griffiths, *Introduction to Elementary Particles*. WILEY-VCH, Weinheim, 1st ed., 2008.

[6] M. Voloshin, *Charmonium*, *Prog.Part.Nucl.Phys.* **61** (2008) 455–511, [arXiv:0711.4556].

[7] J.-P. Lansberg, *J/ψ, ψ' and υ production at hadron colliders: a review*, *Intern. J. of Modern Phys. A* **21** (2006) 3857–3916. arxiv.org/abs/hep-ph/0602091, DOI link.

[8] Alwall, Johan and Herquet, Michel and Maltoni, Fabio and Mattelaer, Olivier and Stelzer, Tim, *MadGraph 5 : Going Beyond*, *JHEP* **1106** (2011) 128, [arXiv:1106.0522].

[9] *Pythia 8 : parameters and settings*, web. home.thep.lu.se/ torbjorn/pythia81html/SaveSettings.html.

[10] *Sherpa Manual*, Web. sherpa.hepforge.org/doc/.

[11] M. Bahr, S. Gieseke, M. A. Gigg, D. Grellscheid, K. Hamilton, O. Latunde-Dada, S. Platzer, P. Richardson, M. H. Seymour, A. Sherstnev, J. Tully, and B. R. Webber, *Herwig++ Physics and Manual*, *Eur. Phys. J. C* **58** (2008) 639–707. arxiv.org/abs/0803.0883, DOI link.

[12] S. Schumann, *The sherpa monte carlo for bsm physics*, . www.thphys.uni-heidelberg.de/ schumann/talks/brussels07.pdf.

[13] *CMS Collaboration* collab., S. Chatrchyan et al., *Measurement of the $B_s \to \mu\mu$ branching fraction and search for $B_0 \to \mu\mu$ with the CMS Experiment*, arXiv:1307.5025.

[14] A. Buckley, H. Hoeth, H. Lacker, H. Schulz, and J. E. von Seggern, *Systematic event generator tuning for the LHC*, *Eur. Phys. J. C* **65** (2010) 331–357. arxiv.org/abs/0907.2973, DOI link.

[15] Aafke Kraan, *MC production of quarkonium in CMS*, . indico.cern.ch/getFile.py/access?contribId=18&sessionId=6&resId=0&materialId=slides&confId=20453.

[16] *CMS* collab., *Transverse-momentum and pseudorapidity distributions of charged hadrons in pp collisions at $\sqrt{s} = 7$ TeV*, *Phys. Rev. Lett.* **105** (May, 2010) 022002. 26 p. arxiv.org/abs/1005.3299, DOI link.

[17] *ATLAS* collab., *Measurement of the differential cross-sections of inclusive, prompt and non-prompt J/ψ production in proton-proton collisions at $\sqrt{s} = 7$ TeV*, *Nuclear Physics B* **850** (2011) 387–444. arxiv.org/abs/1104.3038, DOI link.

[18] *CMS* collab., *Charged particle multiplicities in pp interactions at $\sqrt{s} = 0.9$, 2.36, and 7 TeV*, *J. High Energy Phys.* **1101** (2010) 079. arxiv.org/abs/1011.5531, DOI link.

[19] ALICE Collaboration, *Charged-particle multiplicity measurement in proton-proton collisions at $\sqrt{s} = 7$ TeV with ALICE at LHC*, *Eur. Phys. J. C* **68** (2010), no. 3-4 345–354. arxiv.org/abs/1004.3514, DOI link.

[20] A. Karneyeu, D. Konstantinov, W. Pokorski, S. Prestel, A. Pytel, and P. Skands, *McPlots*, Repository of MC plots comparing High Energy Physics event generators to experimental data. mcplots.cern.ch/.

[21] Matt Dobbs et al, *Hepmc 2, a c^{++} event record for monte carlo generators*, 2010. lcgapp.cern.ch/project/simu/HepMC/206/HepMC2_user_manual.pdf.

[22] J. Beringer et al., *Review of Particle Physics (Particle Data Group)*, *Phys. Rev. D* **86** (2012) 010001. pdg.lbl.gov, DOI link.

[23] A. Maire (for the ALICE Collaboration), *Measurements of inclusive J/ψ production in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV with the ALICE experiment*, in *QCHS 2012* (P. of Science, ed.), 2013. arxiv.org/abs/1301.4058. PoS(Confinement X)211.

[24] *CMS* collab., *J/ψ and $\psi(2S)$ production in pp collisions at $\sqrt{s} = 7$ TeV*, *J. High Energy Phys.* **02** (2012) 11. arxiv.org/abs/1111.1557, DOI link.

[25] *LHCb* collab., *Production of J/ψ and Υ mesons in pp collisions at $\sqrt{s} = 8$ TeV*, *J. High Energy Phys.* (2013). arxiv.org/abs/1304.6977.

[26] *CDF* collab., *Measurement of the J/ψ meson and b-hadron production cross sections in $p\bar{p}$ collisions at $\sqrt{s} = 1960$ GeV*, *Phys. Rev. D* **71** (2005) 032001. arxiv.org/abs/hep-ex/0412071, DOI link.

Acknowledgements

First of all, I want to thank Dr. Kai Schweda, who encouraged me to write my Bachelor thesis in the field of Particle Physics and invited me into the ALICE group in Heidelberg. Furthermore, I want to thank PD Dr. Klaus Reygers for reading and evaluating my thesis.

I am very grateful to Dr. Antonin Maire, for his permanent help and support. Also I want to thank him for proofreading my thesis and for his help when time passed by to fast.

I want to thank Jochen Klein for installing Rivet, Pythia 8 and all its components on the farm in Heidelberg.

Moreover I want to thank Michael Winn for his help regarding theoretical problems and his opinion on my work.

Last but not least I want to thank Maria Heinz, Halil Cakir, Tobias Denz and Yifei Wang for the inspirating and good working atmosphere.

Erklärung:

Ich versichere, dass ich diese Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den (Datum) .....