

The Impact of Different Monte Carlo Models on the Cross Section Measurement of Top-Pair Production at 7 TeV Proton-Proton Collisions

by

Claudius Krause

September 2010

Bachelor Thesis

at

Brandenburg University of Technology, Cottbus

Supervisors

Prof. Dr. Wolfgang Lohmann

Dr. Domnic Merkt

Abstract

From detectors, we record signals, for example hits in the tracker system or deposits in the calorimeters. In order to obtain the momentum of the particles, we need to run certain algorithms to reconstruct tracks in space. More sophisticated algorithms are needed to identify the flavour of quarks, such as b-tagging algorithms. Several steps are needed to test these algorithms. Collision products of proton-proton events are generated using Monte Carlo techniques and their passage through the detector is simulated. After that, the algorithms are applied and, from the differences between input and output quantities, the signal efficiency and the mistagging rate can be obtained. Therefore, it is very important to have the possibility to simulate events accurately. There are, however, many different approaches and algorithms realized in programs, so the question arises if the choice of the Monte Carlo generator influences the measured quantities. In this thesis, two commonly used Monte Carlo generators, SHERPA and MadGraph/MadEvent, are compared and the differences in the selection efficiency of semimuonic $t\bar{t}$ events are estimated. In addition, the distributions of kinematic variables are shown. A special chapter about the matching of matrix elements with parton showers is included. The main algorithms, CKKW for SHERPA and MLM for MadGraph/MadEvent, are introduced.

Contents

Abstract	I
Contents	III
List of Figures	IV
List of Tables	V
Abbreviations	VI
Symbols	VII
1 Physics and Detectors	1
1.1 The Standard Model of Particle Physics	1
1.2 The Top Quark	2
1.3 The Large Hadron Collider	3
1.4 The Compact Muon Solenoid (CMS) Experiment at the LHC	4
2 Introduction to Monte Carlo Simulations	6
2.1 Monte Carlo in general	6
2.2 SHERPA	6
2.3 MadGraph/MadEvent	7
2.4 Matrix Elements Calculation vs. Parton Showering	7
2.5 The CKKW and the MLM Algorithm	8
3 Comparison of MadGraph/MadEvent Samples with SHERPA Samples	10
3.1 Cross Section	10
3.1.1 From the geometric cross section to the general cross section	10
3.1.2 Cross section measurements at LHC	12
3.1.3 Comparison of the obtained cross sections	12
3.2 Distributions of Kinematical Variables	13
3.2.1 Description of the different Jet algorithms	13
3.2.2 Jet p_T	14
3.2.3 Jet η	16
3.2.4 Number of Jets	18
3.2.5 Sum of E_T for Jets	20
3.2.6 Leading Jet p_T	22
3.2.7 Leading Jet η	24
3.2.8 Muon p_T	26
3.2.9 Muon η	27
3.2.10 Missing Energy in Transverse Direction	27
3.3 Selection Algorithm	28
3.3.1 Explanation of the Algorithm	28
3.3.2 Cut Flow Tables	29
3.3.3 Control plots	30
4 Conclusion	32
5 Acknowledgement	33
6 Declaration of Academic Honesty - Selbstständigkeitserklärung	34
References	35

A	Settings for SHERPA and MadGraph/MadEvent	38
A.1	SHERPA: Run.dat	38
A.2	MadGraph: Header of .lhe file	42
A.3	Configuration file of FastSimulation	49

List of Figures

1	Feynman diagrams for $t\bar{t}$ pair production from QCD [CMS-CR]	2
2	Feynman diagrams for single t production from electro-weak physics [CMS-CR]	2
3	3 dimensional view of the CMS detector [1]	4
4	Sliced view of the CMS detector, with particle tracks [2]	5
5	Schematical representation of event generation in SHERPA [3]	7
6	Schematical view of the described backwards clustering algorithm	9
7	p_T [GeV] distribution of all jets for each generator	14
8	Logarithmic p_T [GeV] distribution of all jets for each generator	14
9	Comparison of the p_T [GeV] distribution of all jets for the kt algorithms	15
10	Comparison of the p_T [GeV] distribution of all jets for the antikt algorithms	15
11	Comparison of the p_T [GeV] distribution of all jets for the iterative Cone algorithm	15
12	Comparison of the p_T [GeV] distribution of all jets for the sisCone algorithm	16
13	η distribution of all jets for each MC generator	16
14	Comparison of the η distribution of all jets for the kt algorithms	17
15	Comparison of the η distribution of all jets for the antikt algorithms	17
16	Comparison of the η distribution of all jets for the iterative Cone algorithm	17
17	Comparison of the η distribution of all jets for the sisCone algorithm	18
18	Number of jets for both generators	18
19	Comparison of the number of jets distribution for the kt algorithms	19
20	Comparison of the number of jets distribution for the antikt algorithms	19
21	Comparison of the number of jets distribution for the iterative Cone algorithm	19
22	Comparison of the number of jets distribution for the sisCone algorithm	20
23	Sum E_T [GeV] for each MC generator	20
24	Comparison of the Sum E_T [GeV] distributions for the kt algorithms	21
25	Comparison of the Sum E_T [GeV] distributions for the antikt algorithms	21
26	Comparison of the Sum E_T [GeV] distributions for the iterative Cone algorithm	21
27	Comparison of the Sum E_T [GeV] distributions for the sisCone algorithm	22
28	p_T [GeV] of the leading jet for both generators	22
29	Comparison of the p_T [GeV] distribution of the leading jet for the kt algorithms	23
30	Comparison of the p_T [GeV] distribution of the leading jet for the antikt algorithms	23
31	Comparison of the p_T [GeV] distribution of the leading jet for the iterative Cone algorithm	23
32	Comparison of the p_T [GeV] distribution of the leading jet for the sisCone algorithm	24
33	η of the leading jet	24
34	Comparison of the η distribution of the leading jet for the kt algorithms	25
35	Comparison of the η distribution of the leading jet for the antikt algorithms	25
36	Comparison of the η distribution of the leading jet for the iterative Cone algorithm	25
37	Comparison of the η distribution of the leading jet for the sisCone algorithm	26
38	p_T [GeV] of the muon	26
39	η of the muon	27
40	Missing E_T [GeV] for both generators	27
41	Number of jets after selection step 1	30
42	p_T [GeV] of the muon after step 1	31
43	p_T [GeV] of the muon after step 7	31

List of Tables

1	Particles in the Standard Model [Povh03]	1
2	Vector bosons in the Standard Model [Povh03, Lohm09]	1
3	Final states of $t\bar{t}$ pair production	3
4	Signatures of different particles inside the CMS detector	5
5	Cut flow table for the selection algorithm, $\sigma_{\text{SHERPA}} = \sigma_{\text{MadGraph}} = 110 \text{ pb}$	29
6	Cut flow table for the selection algorithm, $\sigma_{\text{SHERPA}} = 125.5 \text{ pb}$, $\sigma_{\text{MadGraph}} = 104.1 \text{ pb}$	29
7	Calculated efficiencies of the selection algorithm	29
8	Fraction of surviving events per selection step	30

Abbreviations

ALICE	A large ion collider experiment
ATLAS	A toroidal LHC apparatus
CERN	Conseil Européen pour la Recherche Nucléaire, now known as European Organization for Nuclear Research
CKKW	Matching algorithm implemented in SHERPA, named after the authors
CKM matrix	Cabibbo-Kobayashi-Maskawa matrix
CMS	Compact Muon Solenoid
CMSSW	Compact Muon Solenoid Software
DESY	Deutsches Elektronen Synchrotron - German Electron Synchrotron
Ecal	Electromagnetic calorimeter
Hcal	Hadron calorimeter
HF	Hadron Forward calorimeter
LEP	Large Electron-Positron Collider
LHC	Large Hadron Collider
LHCb	Large Hadron Collider beauty
LHCf	Large Hadron Collider forward
MC	Monte Carlo
ME	Matrix element
MET	Missing energy in transverse direction
MLM	Matching algorithm implemented in MadGraph/MadEvent, named after the author
NLL	Next to leading logarithmic order
NLO	Next to leading order
PS	Parton showering
QCD	Quantum chromodynamics
SHERPA	Simulation for high energy reactions of Particles (a Monte Carlo event generator)
TOTEM	Total elastic and diffractive cross section measurement

Symbols

Particles

e^-	Electron
e^+	Positron
μ^-	Muon
μ^+	Antimuon
τ^-	Tauon
τ^+	Antitauon
$\nu_e, \bar{\nu}_e$	Electron neutrino or electron antineutrino, respectively
$\nu_\mu, \bar{\nu}_\mu$	Muon neutrino or muon antineutrino, respectively
$\nu_\tau, \bar{\nu}_\tau$	Tau neutrino or tau antineutrino, respectively
u, \bar{u}	Up quark or up antiquark, respectively
d, \bar{d}	Down quark or down antiquark, respectively
s, \bar{s}	Strange quark or strange antiquark, respectively
c, \bar{c}	Charm quark or charm antiquark, respectively
b, \bar{b}	Bottom quark (sometimes called beauty quark) or bottom antiquark, respectively
t, \bar{t}	Top quark (sometimes called truth quark) or top antiquark, respectively
γ	Photon
g	Gluon
W^-	Gauge boson of the weak interaction, with negative electric charge
W^+	Gauge boson of the weak interaction, with positive electric charge
Z^0	Gauge boson of the weak interaction, with no electric charge (neutral)
H	Higgs boson
$\ell, \bar{\ell}$	Charged lepton in general (e^- , μ^- or τ^-), or antiparticle of a charged lepton, respectively
q, \bar{q}	Light quark (u, d, c, s) or light antiquark ($\bar{u}, \bar{d}, \bar{c}, \bar{s}$), respectively

Mathematical Symbols

∞	Infinity
\mathbb{N}	Set of natural numbers, defined as positive integers (1, 2, 3, ...)
E_{cms}	Center of mass energy
η	Pseudorapidity, used to describe the angle with respect to the beam axis $\eta = -\ln[\tan(\frac{\theta}{2})]$
x_T	Transverse value of a physical quantity x
m_x	Mass of particle x
τ_x	Lifetime of particle x
σ	Cross section
\mathcal{L}	Luminosity
L	Time-integrated luminosity
ϵ	Efficiency

Physical Constants

e	Elementary charge, $e = 1.602 \cdot 10^{-19}$ C
\hbar	Reduced Planck constant, $\hbar = \frac{h}{2\pi} = 6.62606896 \cdot 10^{-34}$ Js, Set equal to 1 in natural units
c	Speed of light in vacuum, $c = 299792458 \frac{\text{m}}{\text{s}}$, Set equal to 1 in natural units, therefore the mass, the momentum, and the energy can be given in units of eV.

1 Physics and Detectors

1.1 The Standard Model of Particle Physics

Our picture of the world has changed constantly in the last century. We learned that matter is made up of elementary particles. According to their spin, they can be divided into two groups. Particles with a spin $s = n\hbar$ ($n \in \mathbb{N}$) obey Bose-Einstein statistics and are therefore called bosons. Particles with a spin $s = \frac{2n-1}{2}\hbar$ ($n \in \mathbb{N}$) are called fermions because they obey Fermi-Dirac statistics. As far as we know today, the fundamental particles are six quarks and six leptons and their corresponding antiparticles. Antiparticles have the same properties as their “normal” partners, except that the electric charge, the colour and the third component of the weak isospin have the opposite sign. The masses are the same. The fundamental particles and antiparticles are fermions ($s = \frac{1}{2}$)¹ and form three families or generations. From the measurement of the Z^0 width, we can exclude the existence of a fourth generation of fermions with a light neutrino [Povh03]. In Table 1, the three families of particles are listed and their electric charge, colour and weak isospin are given. The colours for strongly interacting particles are red, blue, and green.

Fermions	Family			Electric charge	Colour	Weak Isospin	
	1	2	3			left-hd.	right-hd.
Leptons	ν_e	ν_μ	ν_τ	0	-	$\frac{1}{2}$	-
	e^-	μ^-	τ^-	$-1e$	-	$\frac{1}{2}$	0
Quarks	u	c	t	$+\frac{2}{3}e$	r,b,g	$\frac{1}{2}$	0
	d	s	b	$-\frac{1}{3}e$	r,b,g	$\frac{1}{2}$	0

Table 1: Particles in the Standard Model [Povh03]

The interactions between the particles can be described with four fundamental forces - the gravitation, the electromagnetic, the strong, and the weak force. We can describe the latter three within the Standard Model of particle physics [PDG08, Povh03, Denn07]. A Grand Unified Theory, which unifies gravitation (and general relativity) with the other interactions, does not yet exist. In the Standard Model, interactions are mediated by vector or gauge bosons ($s = 1$). They couple to different kinds of charges carried by the particles. The properties of the vector bosons are summarized in Table 2. For the sake of completeness, the properties of the gravitation are given there too, even if gravity is not part of the Standard Model.

Interaction	Couples to	Exchange particle(s)	Rest mass	Range	Rel. strength
strong	colour charge	8 gluons (g)	0	1 fm	1
electromagn.	electric charge	photon (γ)	0	∞	10^{-2}
weak	weak charge	W^+, W^-, Z^0	$m_{W^\pm} = 80.4 \text{ GeV}$, $m_{Z^0} = 91.2 \text{ GeV}$	10^{-3} fm 10^{-3} fm	10^{-5}
gravitation	mass	graviton (hyp.)	0	∞	10^{-40}

Table 2: Vector bosons in the Standard Model [Povh03, Lohm09]

The gluons can couple to themselves because they also have a colour charge. The same is valid for the vector bosons of the weak interaction and the weak charge. The range of the forces is limited by the mass of the vector bosons. The range of the electromagnetic force is infinite due to the zero rest mass of the photons. Because of the gluon self-coupling, the range of the strong force is limited even with zero rest mass for the gluons. The mediator of gravitation, the graviton, is only a hypothetic particle which has not yet been seen in any experiment.

So far, this model is in agreement with experiments, but there are still many unsolved mysteries. It is, for example, still unknown where the masses of the gauge bosons come from. The pure Standard Model predicts them to be massless. Adding the masses by hand will break the gauge symmetry. A solution,

¹As convention, we set $\hbar = c = 1$.

independently proposed by Higgs [Higgs64] and Englert and Brout [Brout64], introduces a scalar field, the Higgs field. This field produces the masses of the W^\pm and the Z^0 . This can be compared with the Meissner-Ochsenfeld-effect in superconductors. At low energies or temperatures in the superconducting phase, the magnetic fields are expelled from the superconductor. The photons, mediator of the field, drop off exponentially at the surface of the substrate, as if they gained a mass. This is also called spontaneous symmetry breaking [Povh03].

The theory of this so-called Higgs mechanism proposes four gauge bosons, but three of them are absorbed by the W^\pm/Z . The fourth one is still free. This is the Higgs boson. It is now the only particle of the Standard Model which had not been found in an experiment. Because of its high mass ($m_H > 114.4$ GeV [PDG08]), it needs large energies to be produced. The particle accelerator Large Hadron Collider, which will be described in Chapter 1.3, will reach the necessary energy. Another particle of interest, which will be produced in large numbers at the LHC, is the top quark. Its properties are given in the next chapter.

1.2 The Top Quark

The top quark, with a mass of $m_t = 171.2 \pm 2.1$ GeV, is the heaviest quark known so far. Its electric charge is $q = \frac{2}{3}e$, and the third component of the weak isospin is $T_3 = +\frac{1}{2}$. At the LHC, it will be produced in several fundamental processes of proton-proton collisions. These processes can be divided into two groups. The first is top-antitop pair production due to the strong force and is therefore described by quantum chromodynamics. Figure 1 shows the leading order Feynman diagrams of these processes – gluon-gluon scattering and quark-antiquark annihilation. At the LHC, there will be about 13 % $t\bar{t}$ pair production due to quark-antiquark annihilation and 87 % due to gluon-gluon scattering [DESY]. This group dominates the top quark production in hadron colliders.

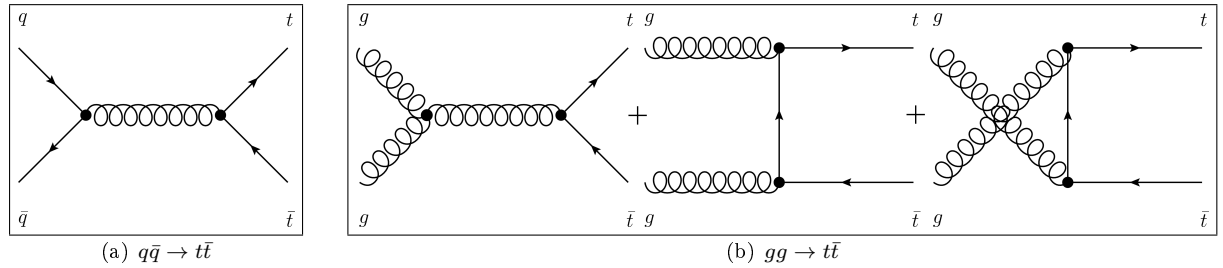


Figure 1: Feynman diagrams for $t\bar{t}$ pair production from QCD [CMS-CR]

In the second group, top quarks can be generated as single quarks (without the antitop, or vice versa). These processes are mediated by a W -boson. The Feynman diagrams in leading order for single top production are shown in Figure 2.

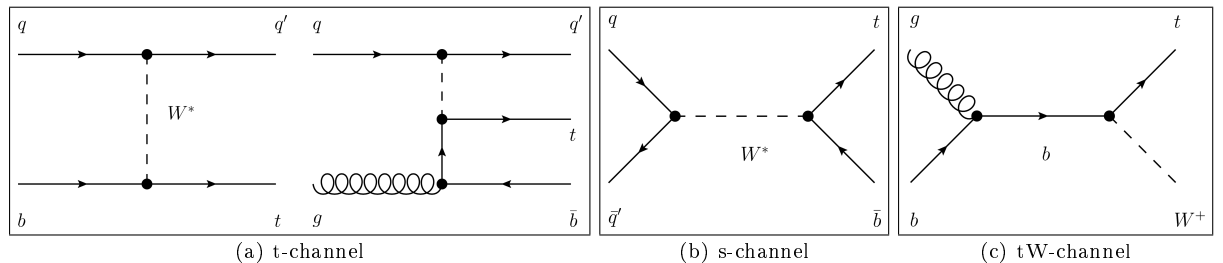


Figure 2: Feynman diagrams for single t production from electro-weak physics [CMS-CR]

It is very difficult to identify top quarks from single production because they have a less distinctive signature and significantly larger backgrounds [PDG08]. First measurements have been published by the experiments at the Tevatron [Garc10].

The top quark has a very short lifetime of $\tau_t \approx 0.5 \cdot 10^{-24}$ s [PDG08]. This time is even shorter than

the time needed for hadronization, typically $t_{\text{had}} \approx 10^{-23}$ s. This time is needed for the partons to form Hadrons. The top quark, however, decays before hadronizing and we cannot observe any top-flavored hadrons. In the investigations, top quarks can be treated as quasi-free particles. Their dynamics are only determined by electro-weak and perturbative strong interactions [Bigi86]. The top quark decays with a probability of 99.4% [PDG08] weakly to a bottom quark and a W^+ , whose further decay can be divided into three classes, shown in Table 3. The ℓ stands for any charged lepton, e , μ , or τ . We can get additional jets in the detector because of gluon radiation in the initial or final state.

Name	Process	Branching fraction
nonleptonic / all-jets	$t\bar{t} \rightarrow W^+bW^-\bar{b} \rightarrow q\bar{q}'bq''\bar{q}'''\bar{b}$	46.2%
semileptonic / lepton + jets	$t\bar{t} \rightarrow W^+bW^-\bar{b} \rightarrow q\bar{q}'b\ell\bar{\nu}_\ell\bar{b} + \bar{\ell}\nu_\ell b\bar{q}\bar{q}'\bar{b}$	43.5%
dileptonic	$t\bar{t} \rightarrow W^+bW^-\bar{b} \rightarrow \bar{\ell}\nu_\ell b\ell'\bar{\nu}_{\ell'}\bar{b}$	10.3%

Table 3: Final states of $t\bar{t}$ pair production

Several selection schemes exist to identify $t\bar{t}$ events. They suppress the main background processes from W/Z +jets and QCD events. In Chapter 3.3, a scheme to identify semimuonic $t\bar{t}$ events is described. However, all information about the top quark is obtained from the experiments at Tevatron. These experiments recorded a small number of events compared to the expected production rate at the LHC. The LHC will be introduced in the next chapter.

1.3 The Large Hadron Collider

Due to the very small dimensions of the subatomic particles, we can investigate their structure only by scattering experiments. For such experiments, particle accelerators are needed. The Large Hadron Collider (LHC) at CERN (Geneva, Switzerland) is currently the machine reaching the highest energy in proton-proton collisions. The current record is $E_{\text{cms}} = 7$ TeV and was set on 30th March 2010. The LHC was planned and constructed in the last 20 years and built in the 26.659 km tunnel of the Large Electron-Positron collider (LEP). Along the circle, there are six experiments built by collaborations from all over the world. The two multi-purpose experiments are named “**A** Toroidal **L**HC **A**pparatus” (ATLAS) and “**C**ompact **M**uon **S**olenoid” (CMS). They were designed differently in order to perform discoveries independently for cross-confirmation. The CMS-Experiment is described in more detail in the next chapter. The other detectors are “**A** Large **I**on **C**ollider **E**xperiment” (ALICE), “**L**arge **H**adron **C**ollider **b**eauty” (LHCb), “**T**otal **E**lastic and diffractive cross section **M**easurement” (TOTEM) and “**L**arge **H**adron **C**ollider **f**orward” (LHCf). The latter three are smaller than the first three and designed for special purposes. As an example, TOTEM will detect the forward particles, which are not accessible by the other experiments in that way, and will be also a luminosity monitor for the LHC.

The LHC will be running at 7 TeV for several months to collect a large amount of data. With these data, the known realm of physics should be reestablished and maybe some new physics could be discovered. In 2014, an upgrade to 14 TeV center-of-mass energy is planned. This is the energy, that the LHC was designed for. At this energy, the colliding partons carry around 1 TeV energy.

1.4 The Compact Muon Solenoid (CMS) Experiment at the LHC

The Compact Muon Solenoid Experiment (CMS) is located near Cessy, France. The CMS collaboration includes more than 3000 scientists from 38 countries and 183 institutes. It is one of the largest collaborations in the history of mankind. In Figure 3, a picture of the CMS detector in 3D is shown.

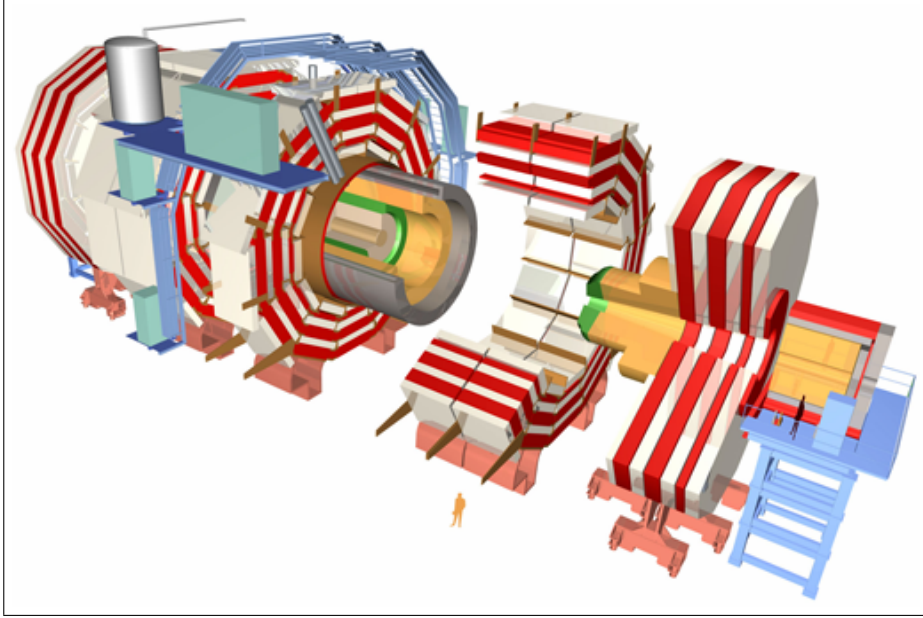


Figure 3: 3 dimensional view of the CMS detector [1]

The CMS detector was built to detect and measure particles with excellent performance. The momentum of charged particles is measured with the use of a strong magnetic field, forcing charged particles to move on a helix due to the Lorentz force. The radius of the helix, r , depends on the momentum, p , and the charge of the particle, q , like

$$r = \frac{p}{qB}, \quad (1)$$

where B is the magnetic field. The uncertainty of the momentum measurement depends for a given track length on the sagitta of the circle segment. Hence a smaller radius, made by a large magnetic field, is of advantage. The relative error in the transverse momentum measurement, $\frac{\sigma_{p_\perp}}{p_\perp}$, is given by

$$\frac{\sigma_{p_\perp}}{p_\perp} \sim \frac{p_\perp \sigma_x}{B}, \quad (2)$$

where σ_x is the error in position measurement of the space points along the track [Kass]. The field in CMS is created by a superconducting solenoid and has a strength of $B = 4$ T. This solenoid has a length of 13 m and a diameter of 7 m [CMS1]. This size allows it to accommodate other detecting devices inside the solenoid. This compact design gives the detector its name. The subsystems are arranged in different layers. Each layer has a different task in the detection of the particles created in the collisions. The tracker system is located in the inner shell, close to the interaction point (IP). It is made of silicon strip and silicon pixel detectors. These can record the track of a charged particle with a precision of about $10 \mu\text{m}$ [Kass]. This is the key to very accurate momentum measurements. The tracker system is surrounded by the electromagnetic calorimeter, ECal. ECal measures the energy of photons and electrons/positrons. It is made of lead tungstate crystals (PbWO_4), a dense scintillating material. The next layer is the hadron calorimeter, HCal. It measures the energy of hadrons produced in the collisions. Muons, created in the collisions, are identified in muon chambers. This is the only subsystem which is not inside the solenoid. The muon is a particle like an electron but is about 200 times heavier ($m_\mu = 105.7 \text{ MeV}$). The cross section, σ , of bremsstrahlung in matter goes with $\sigma \sim \frac{1}{m_\mu^2}$. Therefore, the probability of bremsstrahlung of a muon is 40000 times less than the probability of an electron. Since muons do not interact strongly

they also pass the HCal. This is why the muon chamber is placed at the very outside of the detector as it is very unlikely that other particles will reach this subsystem. The actual position of the muons is measured with drift tubes, while cathode strip chambers and resistive plate chambers work as triggers. In order to determine the whole energy of each event, the detector should be closed hermetically by calorimeters. The end caps, perpendicular to the beam axis, fulfill this task. The cylindrical part of the detector is called barrel. In the forward region, close to the beam pipe, is the Hadron Forward calorimeter (HF). It detects particles at low polar angles.

Altogether, the CMS detector is 21 meters long and has a diameter of 15 meters [CMS2]. It weighs 12500 tonnes. In Figure 4, one can see a slice of the CMS detector with some particle tracks indicated. Table 4 explains the identification of different particles within the detector. Each particle has a specific signature. Neutrinos can be measured by calculating the missing transverse energy and momentum, respectively. The longitudinal quantities cannot be defined because, in proton-proton collisions, it is uncertain which quark or gluon carries what part of the whole energy or momentum.

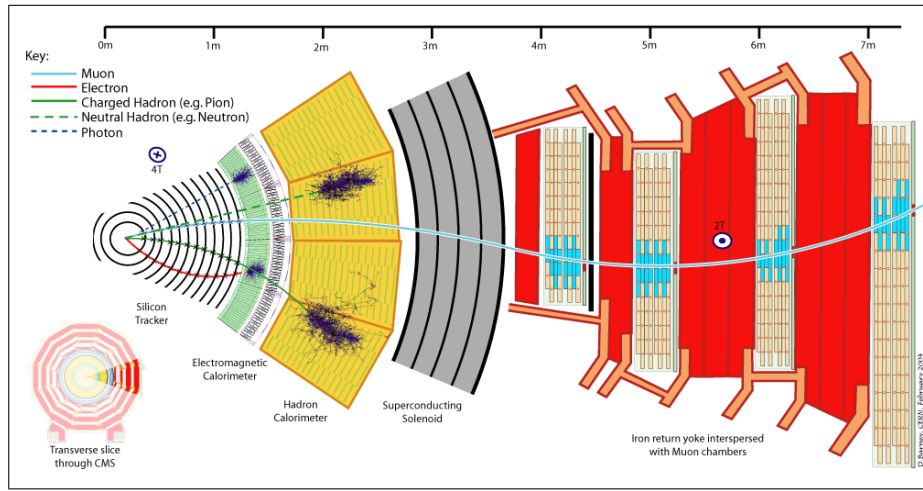


Figure 4: Sliced view of the CMS detector, with particle tracks [2]

Particle	Tracker	ECal	HCal	Muon chamber
electron (e^\pm)	curved track, according to electric charge	shower	-	-
muon (μ^\pm)	curved track, according to electric charge	-	-	curved track, according to electric charge
photon (γ)	-	shower	-	-
charged hadron	curved track, according to electric charge	-	shower	-
neutral hadron	-	-	shower	-
neutrino (ν_ℓ)	-	-	-	-

Table 4: Signatures of different particles inside the CMS detector

Due to hadronization, the observed particles in the final state of proton-proton collisions are clustered together in certain regions. This “large amount of hadronic energy in a small angular region” [Cata93] is called jet. In order to find jets in the data a clear definition is needed. This definition is given by different jet algorithms described in Chapter 3.2.1.

2 Introduction to Monte Carlo Simulations

2.1 Monte Carlo in general

Monte Carlo generators in particle physics are programs that simulate particle collisions, so called events, with the use of the Monte Carlo method. This method is used in many different ways, e.g. in computational physics for solving problems with a large number of degrees of freedom, for example in fluids. It is based on the law of large numbers and a random number generator. The following example can illustrate the main concept of the idea. The volume of the intersection of a torus and a cone can be computed by scattering a large number of points homogeneously in a cubic volume, containing the intersection. From the ratio of points in the intersection and the volume of the cube, we can estimate the volume of the intersection [Bohm10]. With the number of points used for the calculation, we can tune the accuracy of the volume due to the law of large numbers.

The random number generator can be used to generate randomly distributed events. This is used in particle physics. Some of these programs work as follows. The known theory of particle interactions (for example the Standard Model) and some experimental results, not provided by theory (like the values of the CKM matrix elements), are given as input. The program computes transition matrix elements for the interaction based on that input. For this integration, the Monte Carlo method is used as a numerical approach. From the matrix elements, the program knows the probability of certain events. The random event generator can now generate the final state distribution based on the given probabilities. Depending on the program used and the results needed by the user, the program can continue with these particles to generate hadrons and particle showers. In this thesis, the impact of the use of two different Monte Carlo models on the measurement of the cross section of $t\bar{t}$ events is shown. In addition, distributions of kinematic quantities will be compared. The generators which were used are SHERPA and MadGraph/MadEvent. A short introduction on these programs is given in the following chapters.

2.2 SHERPA

SHERPA [Glei03, Glei08] is a Monte Carlo event generator used to simulate particle collisions with complete hadronic final states [SHERPA2]. The name is an abbreviation for **S**imulation for **H**igh **E**nergy **R**eactions of **P**articles. Processes such as lepton-lepton, lepton-photon, photon-photon, and hadronic collisions can be generated. The program includes all physics of the Standard Model and some extensions, for example the Minimal Supersymmetric extension.

The program itself is built in a modular way according to the different phases in event generation. Each module can be run separately, which offers a broad and flexible use of SHERPA. In this work, the complete chain of all modules is used. Quoted according to the SHERPA homepage [SHERPA3] it works as follows:

“Schematically this translates into

- 1. considering the production of heavy and/or highly energetic particles through appropriate matrix elements that are exact at some perturbative order and respect quantum interferences etc.,*
- 2. radiating lighter particles - like for instance gluons and photons - which are softer and tend to be collinear with their emitters through the parton shower, which correctly resums the leading terms to all orders in perturbation theory,*
- 3. the eventual decay of heavy unstable particles, again through matrix elements, again dressed by parton showers to fill the phase space of multiple soft emission in the perturbative regime,*
- 4. the hadronization of the quarks and gluons into hadrons, which in many cases may not be stable,*
- 5. their decays into long-lived, i.e. practically stable, hadrons that finally hit the detector.”*

Figure 5 illustrates these steps graphically. The matching of additionally emitted gluons with matrix element or parton shower treatments is done with an algorithm called CKKW. The goal and the working principle of this algorithm is described later.

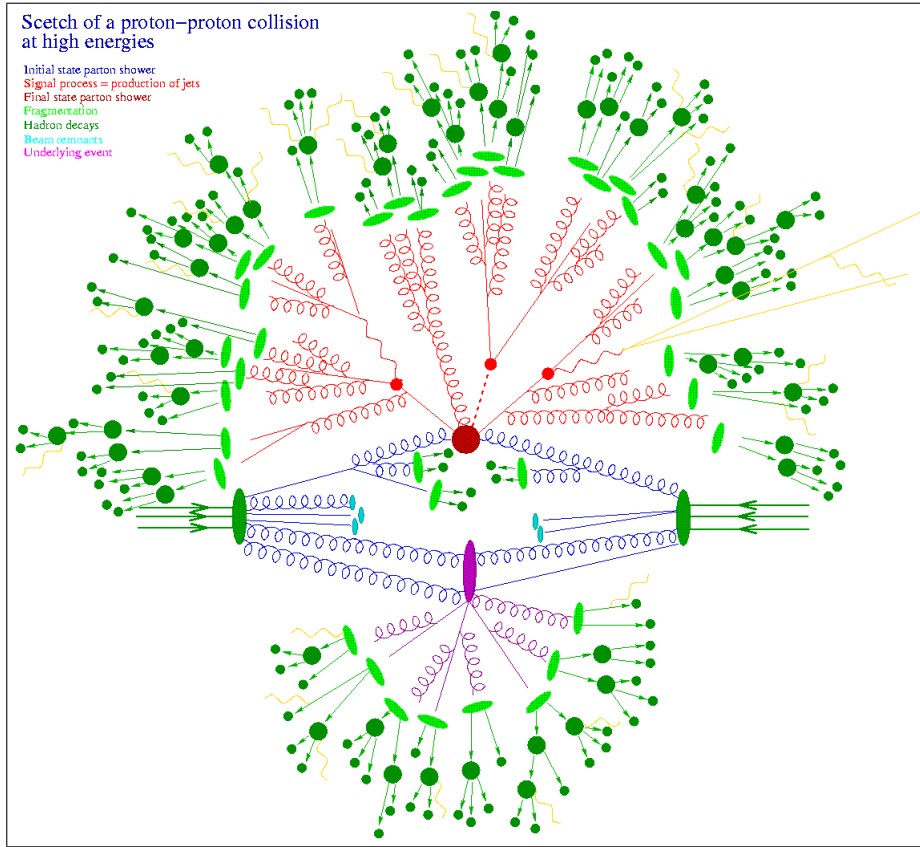


Figure 5: Schematical representation of event generation in SHERPA [3]

2.3 MadGraph/MadEvent

The steps of event generation are similar to SHERPA, but the structure of the program is different. As in SHERPA, not only the Standard Model, but several other physics models are included. The user-defined process (with up to 9 external particles) is calculated by the matrix element generator MadGraph [Stelz94]. This program generates the amplitudes of all relevant subprocesses and the mappings for the integration over the phase space. In the next step, the event generation is done by the program MadEvent [Malt02]. The events are stored in a standard output format called Les Houches Event Files (.lhe) [Boos01]. The showering and hadronization step is done by other programs, for example PYTHIA [PYTHIA]. One of the main differences to SHERPA is that the showering and hadronization step is completely separated from the matrix element generation. The matching of the parton showering with the matrix element calculation is also different. It is done with a modified MLM algorithm. A description of matrix element calculation and parton showering is given in the next chapter. Whenever it is referred to MadGraph within this work, the event generation with MadGraph, MadEvent and PYTHIA in total is meant, as long as nothing else is mentioned.

2.4 Matrix Elements Calculation vs. Parton Showering

The main features of the particle collisions at the LHC, and especially the $t\bar{t}$ -production, are the large number of final state particles and therefore the large number of jets. This is a result of the high center-of-mass energy in the collision, which produces heavy particles with a very short lifetime. They decay into many stable particles inside the detector. Additional jets are created due to gluon emission in the initial or final state. There are two main methods to deal with these jets [Krau02, Erik07].

The calculation can be done with the use of matrix elements (ME). They can be given at any perturbative order, but mostly the lowest perturbative order, called tree-level, is used. As a simplification, the final state particles are identified with their jets, which is legitimate at tree-level. This correlation is sometimes

more difficult at higher perturbative orders. The main advantages of the ME calculation are that they are exact at their perturbative order and that they take all interference effects into account. But there are a few disadvantages. With a large number of final state particles, a large number of MEs has to be calculated because more and more diagrams are leading to the given final state. The next problem is that the fragmentation and the hadronization are not part of the ME calculation because MEs are calculated at parton level. The fragmentation and the hadronization refer to the formation of colourless states out of the partons due to the confinement of the QCD. Therefore, fragmentation and the hadronization describe the evolution of the partons to the observed hadrons. Sometimes the decay of unstable particles is included in the definition of hadronization. However, these processes are done with phenomenological models, which are independent of the ME calculation and, therefore, need an extra tuning of their parameters. This tuning needs to be done for each center-of-mass energy. This is an impossible task, which limits the use of ME calculations.

Another way to deal with the extra emissions is the use of parton showers (PS). In this approximation, we have to distinguish between initial and final state radiation [PYTHIA]. It is valid in next-to-leading-logarithmic-order (NLL). The final state radiation starts at a scale Q_{cut}^2 , which is set by the hard process. It is time-like ($Q^2 > 0$) and decreases with each branching process until it reaches a lower cut-off value, Q_0^2 . The evolution is described with the use of Sudakov form factors, which gives a probability for the emission. The description of the initial state radiation is a little more complicated because the shower evolution must be done “backwards” to be treated like the final state shower. This is necessary, because the initial state radiation can only be calculated if the scale of the hard process is known. It starts at this scale Q_{cut}^2 and decreases to Q_0^2 (which does not have to be equal to the value of Q_0^2 of the final state radiation). This time, it is space-like ($Q^2 < 0$). Altogether, the parton showering leads to a realistic description of the complete final state, because it works at hadron level. The partons can easily be converted to jets. This combination of the hard scale and the fragmentation scale is the main advantage of this approach. Therefore, a tuning of the fragmentation parameters for every single energy is no longer needed. But contributions from interference are not considered, so this approach is not appropriate in all situations. It is e.g. a poor simulation of events with widely separated jets [CKKW].

It is obvious, that a combination of these two approaches will lead to the best results. But there is one major problem; some events will be counted twice. Therefore, it is needed to divide the phase space into two disjointed regions. One in which ME calculations are used and one in which PS is used. Algorithms performing such a division are called merging or matching algorithms. The two main algorithms used today are the CKKW-algorithm [CKKW] as implemented in SHERPA and the MLM-algorithm, which is, in a modified way, implemented in MadGraph/MadEvent. In the next chapters a short overview on these two algorithms is given.

2.5 The CKKW and the MLM Algorithm

We now want to have a closer look at the CKKW-algorithm which is implemented in SHERPA. It is denoted after its authors (S. Catani, F. Krauss, R. Kuhn, B.R. Webber) [CKKW]. The algorithm is applied to the matrix element calculation before the parton showering is performed. The initial and final state partons are clustered with the use of a quantity k_{\perp} [Cata91, Cata92, Cata93]. Two particles (i) and (j) are considered as belonging to one jet if the value²

$$k_{\perp}^{(ij)2} = 2 \min\{p_{\perp}^{(i)}, p_{\perp}^{(j)}\}^2 \left[\cosh(\eta^{(i)} - \eta^{(j)}) - \cos(\phi^{(i)} - \phi^{(j)}) \right] \quad (3)$$

is lower than the user-given cut $k_{\perp, \text{cut}}^2$ and the transverse momentum of each jet is larger than $k_{\perp, \text{cut}}$ [Höche06, Alwall08]. This clustering is performed until a $2 \rightarrow 2$ core process is remaining. This gives a “shower history” where for each node or vertex a value of $k_{\perp, i}$ can be obtained [Erik07]. Now each chain is weighted by two different weights. The first one is the strong coupling constant at each vertex (where the values of $k_{\perp, i}$ are used to calculate α_S) divided by the coupling which was used in the ME calculation. Second, an analytical Sudakov weight is calculated for each line between the vertices. Terms that would also appear in the PS more likely are suppressed by the weighting at the ME level. Now the showering is performed.

The MLM scheme [MLM1, MLM2], which is implemented in MadGraph/MadEvent, is a little bit different from the one originally proposed. The MLM algorithm is named after its author (M. L. Mangano). The

²This formula is valid for hadron-hadron collisions.

algorithm implemented in MadGraph is a little closer to the CKKW-scheme used in SHERPA and works as follows. The first steps are similar to the steps in the CKKW-algorithm. The partons are clustered with the use of the k_{\perp} -measure and the nodes are weighted according to the strong coupling - but there is no Sudakov weighting applied. Instead, the parton showering is performed without any vetos. After the showering and before the hadronization, a k_{\perp} -jet finding algorithm is used to form jets. (The original proposed MLM scheme uses a cone algorithm.) The cut value for the jet definition has to be specified in addition. In order to avoid misunderstandings, we denote it as $q_{\perp, \text{cut}}$. Then, the partons are matched to the jets in the following way. The algorithm starts with the parton with the largest transverse momentum. The closest jet in the (η, ϕ) -plane is assigned to the parton if the q_{\perp} is lower than the defined $q_{\perp, \text{cut}}$. The parton and the jet match. Now, the jet will be removed from the set of available jets, and the algorithm does the same steps for the next parton until all partons are considered. If a parton does not match to a jet, for example because two partons are too close together to create separate jets, the event will be rejected. If the number of generated partons is smaller than the process-given maximum number of partons, the number of jets must equal the number of partons. Additional jets are only allowed in the case that the maximum number of partons is generated and the additional jet has a smaller q_{\perp} than the softest matched jet.

In Figure 6, one can see an example, how the clustering algorithm generates the shower history (adopted from [Erik07]).

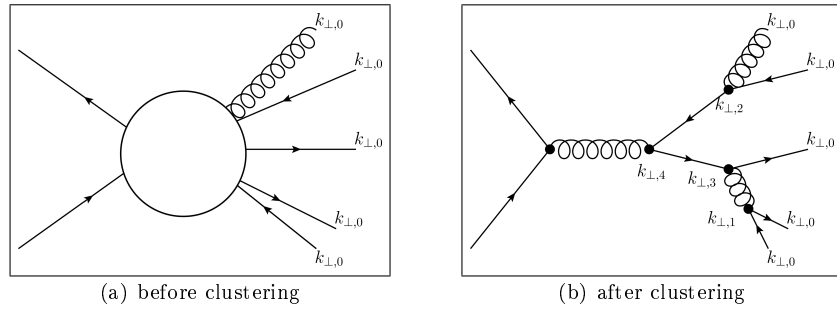


Figure 6: Schematical view of the described backwards clustering algorithm

3 Comparison of MadGraph/MadEvent Samples with SHERPA Samples

In order to find the difference between SHERPA and MadGraph, we need to simulate events of the same final state in both programs and then we can compare the results of both. The process chosen for analysis was

$$pp \rightarrow t\bar{t} + X \text{ jets}, \quad X \leq 2$$

All three decay channels (dileptonic, semileptonic, and hadronic) are allowed. According to the current status of the LHC, the energy per beam was chosen to be $E = 3.5$ TeV, meaning in total $E_{\text{cms}} = 7$ TeV. The Standard Model was the used physics model.

The SHERPA simulations were done with SHERPA 1.2.1. Within SHERPA, the fragmentation and hadronization, including the decay of unstable particles, was performed by the Lund package, which is PYTHIA-like and, therefore, closer to MadGraph. The MadGraph/MadEvent simulations have been performed using MadGraph Version 4.4.44. Here, the decay was treated by the implemented decay package. The showering was done with PYTHIA 6.4.

For both programs, the $k_{\perp, \text{cut}}$ -value for the matching algorithm was set to $k_{\perp, \text{cut}} = 30$ GeV, while the jet definition value at parton level, $q_{\perp, \text{cut}}$, was set to $q_{\perp, \text{cut}} = 20$ GeV. The whole set of parameters, given to the programs with the corresponding files, is listed in Appendix A of this work. For the analysis, about 10^5 events were used. Because of the rejection of events from MadGraph after the showering (due to the MLM algorithm), we generated about $3 \cdot 10^5$ events with MadGraph.

After the generation, the events were processed with the FastSimulation of CMSSW_3_7_0 and stored at the dCache storage system of DESY. The parameter file of the FastSimulation for MadGraph is given in the Appendix A as well. The corresponding file for SHERPA differs only in the section about the imported file. Therefore, it is not given in addition. The further analysis was done with the FWLite package of CMSSW. This package implements the use of ROOT.

In the following, the cross section, which is an important quantity to characterize the interaction probability between particles will be introduced. The different cross sections obtained from the programs are given and discussed. After that, the distributions of the main kinematic variables in both simulations are shown. As a third comparison, the different results of applying a script to select semimuonic events to both samples are presented and discussed.

3.1 Cross Section

The cross section σ of a scattering process is a quantity which describes the probability of an interaction between two particles. In order to understand the meaning of this quantity, let us consider the easiest example which is the geometric cross section:

3.1.1 From the geometric cross section to the general cross section

Imagine a homogeneous and monoenergetic particle beam with point-like particles, a , hitting a target. We can assume this target is a volume, V , with a front area A and thickness d . Let us further assume that there are N_b scattering centers, each with the area σ_b in this volume. They have a homogeneous density n_b . The number of reactions per unit time, \dot{N} , is then given by

$$\dot{N} = \dot{N}_a \cdot p, \tag{4}$$

where \dot{N}_a is the number of incoming particles per unit time and p the probability that a particle a hits a target b . With the assumption that each particle of the beam has only one reaction and that the scattering centers are not overlapping each other, we can define the probability p in the following way:

$$p = \frac{\text{target area}}{\text{total area}} = \frac{N_b \sigma_b}{A}. \tag{5}$$

This can be put in equation (4) and transposed to

$$\sigma_b = \frac{\dot{N} \cdot A}{\dot{N}_a \cdot N_b}. \tag{6}$$

With the definition of the particle flux $\Phi_a = \frac{\dot{N}_a}{A} = n_a \cdot v_a$, we can find for a constant and homogeneous beam [Povh03]:

$$\sigma_b = \frac{\dot{N}}{\underbrace{\Phi_a \cdot N_b}_{\mathcal{L}}} = \frac{\text{number of reactions per unit time}}{\text{beam particles per unit time per unit area} \cdot \text{scattering centers}} \quad (7)$$

The denominator is called luminosity, \mathcal{L} . At the LHC we do not have a constant particle flux Φ_a , so we need to find another expression for \mathcal{L} . In the case where we have an accelerator with circulating beams, we can write \dot{N}_a , the number of incoming particles per unit time as

$$\dot{N}_a = \frac{N_{a,\text{total}}}{N_{\text{bunches}}} \cdot \frac{N_{\text{bunches}}}{t_{\text{orbit}}}, \quad (8)$$

where N_{bunches} is the number of bunches of particles a and b which are circulating in the accelerator and t_{orbit} is the time needed for one orbit. N_a is then just the number of particles in one bunch. The time needed for one orbit can be expressed with the use of the particle velocity v and the circumference u of the accelerator. The luminosity is then given by

$$\mathcal{L} = \Phi_a \cdot N_b = \frac{\dot{N}_a}{A} \cdot N_b = \frac{N_a \cdot N_{\text{bunches}} \cdot \frac{v}{u} \cdot N_b}{A}. \quad (9)$$

On the other hand, we can determine the cross section theoretically with the use of Fermi's golden rule. It states that the reaction rate per incoming particle and per target particle is given by

$$W = \frac{2\pi}{\hbar} |\mathcal{M}_{fi}|^2 \cdot \rho(E), \quad (10)$$

where \mathcal{M}_{fi} is the matrix element for the interaction, defined by

$$\mathcal{M}_{fi} = \langle \psi_f | \mathcal{H}_{\text{int}} | \psi_i \rangle = \int \psi_f^* \mathcal{H}_{\text{int}} \psi_i dV. \quad (11)$$

\mathcal{H}_{int} is the Hamiltonian for the interaction, ψ_i and ψ_f denote the initial and final state wave function, respectively. The function $\rho(E)$ in equation (10) is the density of final states in the energy region between E and $E + dE$. It can be derived in the following way. In phase space, each particle occupies a volume $h^3 = (2\pi\hbar)^3$, due to the uncertainty principle. A spinless particle scattered in the volume V and in the momentum area between p and $p + dp$ shall be considered. This area is in momentum space a spherical shell with inner radius p and outer radius $p + dp$. The corresponding volume in momentum space is given by $4\pi p^2 dp$. The number of final states available is then

$$dn(p) = \frac{V \cdot 4\pi p^2}{(2\pi\hbar)^3} dp. \quad (12)$$

We now want to express this relation with respect to the energy of the particle. For that, we need the energy-momentum relation

$$E = \sqrt{m_0^2 c^4 + p^2 c^2} = mc^2 \quad (13)$$

and its derivative

$$\begin{aligned} \frac{dE}{dp} &= \frac{1}{2\sqrt{m_0^2 c^4 + p^2 c^2}} \cdot 2pc^2 = \frac{pc^2}{mc^2} = \frac{p}{m} = v \\ \rightarrow dE &= v dp \end{aligned} \quad (14)$$

And from that, we can find the expression for the density of final states:

$$\rho(E) = \frac{dn(E)}{dE} = \frac{V \cdot 4\pi p^2}{v \cdot (2\pi\hbar)^3} \quad (15)$$

The reaction rate per incoming particle and per target particle (eq. 10) can be expressed using beam properties too.

$$W = \frac{\dot{N}}{N_a \cdot N_b} \quad (16)$$

With the use of equation 7 and the definition of the particle flux, we can find

$$W = \frac{\sigma \cdot \Phi_a}{N_a} = \frac{\sigma \cdot n_a \cdot v_a}{N_a} = \frac{\sigma \cdot v_a}{V}. \quad (17)$$

Altogether, we find the expression for the cross section:

$$\sigma = \frac{2\pi}{\hbar \cdot v_a} |\mathcal{M}_{fi}|^2 \cdot \rho(E) \cdot V \quad (18)$$

3.1.2 Cross section measurements at LHC

In order to determine the cross section of a given process, for example $t\bar{t}$ -production, in the experiment we need the luminosity and the number of events per time. The luminosity will be given by the accelerator and measured with the help of gauge processes. In these processes, the cross section is known and thus one only needs to determine the event rates. The TOTEM experiment is one example of a luminosity measurement at the LHC. In CMS, this will be done by the forward hadron calorimeter (HF) [CMS-TDR]. Typically, the time-integrated luminosity, L , will be determined. Then one only needs to count the number of events, of the process considered, in that period. To count events, they must be identified in the detector. In the first step, particles are reconstructed by hits in the tracker system and the calorimeters. In the second step, criteria are applied to the events, specific for the process to be selected, (see the selection algorithm in Chapter 3.3 for example), in order to suppress the background and enhance the signal to background ratio. The algorithms are, however, not identifying each event correctly. That is why we need to correct the number of events by a factor called efficiency.

The efficiency can be obtained by the use of Monte Carlo generators. Generated events can be read completely, and therefore we know exactly how many events of each kind we have. These events are passed through a detector simulation software. This software simulates the response of the detector to the events. The result can be treated like real data. We can now apply our selection algorithms on the reconstructed events and can compare the number of selected events with the number of events generated. That ratio is the above-mentioned efficiency.

3.1.3 Comparison of the obtained cross sections

The total cross sections for $t\bar{t}$ -production with up to two additional jets and all decay channels were obtained from the programs after the matching of parton showers with matrix elements. Since the generation process was divided in several jobs with different random seed numbers (in order to save computing time), the values given below are averaged over all jobs.

$$\sigma_{\text{SHERPA}} = 125.5 \pm 1.1 \text{ pb} \quad \sigma_{\text{MadGraph}} = 104.1 \pm 3.1 \text{ pb}$$

They are in the same order of magnitude and differ by about 20 pb ($\approx 20\%$). This difference can be explained in the following way. According to [Malt02], MadGraph generates the matrix elements for “all relevant subprocesses”. As nothing else was mentioned in the references, we assume that MadGraph and MadEvent use only tree level processes for the calculation and the generation. On the other hand, AMEGIC++, the matrix element generator of SHERPA, takes some next to leading order contributions, such as Coulomb corrections and running particle masses and widths, into account [Krau01]. This may lead to the differences found. The errors of the cross section obtained with the programs are similar. For SHERPA the error was calculated with using 5000 generated events and in MadGraph using about 800 events, in both cases after matching. Because it is a statistical error, it will drop with a larger number of events, hence we expect a larger error for MadGraph.

The total cross section is important for the total number of events at a given integrated luminosity, L . ($\rightarrow N = L \cdot \sigma$) The cross section is not important for the distributions of the kinematic variables (it does not change the shape, but only the height of them), therefore the further analysis in this work is done with the same cross sections for both programs ($\sigma = 110 \text{ pb}$). The different distributions shown in the next chapter are obtained for the same integrated luminosity $L = 20 \text{ pb}^{-1}$ if nothing else will be mentioned.

3.2 Distributions of Kinematical Variables

The distributions of the kinematical variables, such as p_T and η , are shown for different jet-finding algorithms. After the introduction of the algorithms, the distributions are shown for both generators and then they are compared.

3.2.1 Description of the different Jet algorithms

As indicated in Chapter 1.4, the final state particles are clustered together in jets. They need a clear definition that needs to be collinear- and infrared-safe. This means, that a jet should be splitted if the original parton splits into two collinear partons with smaller p_T or if it emits a soft parton. Another important property is the number of steps needed in an algorithm to find the jets. At the LHC, more than 100 particles can occur per event with many thousands of events per second. A jet algorithm with a large number of steps will require a lot of the computing resources.

The existing algorithms can be divided into two groups - clustering and cone algorithms. The main algorithms used in analysis are described in the following.

The k_{\perp} - or **kt algorithm** is a clustering algorithm. It defines a distance between two entities [Cheka02]³:

$$d_{ij} = \min(E_{\perp i}^2, E_{\perp j}^2) \cdot \frac{[(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2]}{R^2} \quad (19)$$

and a distance between the entity and the beam axis:

$$d_{iB} = E_{\perp i}^2 \quad (20)$$

In this equations, $E_{\perp i}$ denotes the transverse energy, η_i the pseudorapidity, and ϕ_i the azimuthal angle of the i th entity. An entity can be a particle or a pseudojet. R is a free parameter. The algorithm starts with a list of all particles and looks for the smallest value of d_{ij} and d_{iB} . If it is d_{ij} , the two entities (in the first step particles) are merged to a pseudojet. If d_{iB} is the smallest, which is not expected in the first steps, this entity will be removed from the list of available entities and is called a jet. These steps are done until no entities are left. In the algorithms used, kt4 and kt6, R is set to 0.4 or 0.6, respectively. The **anti- k_{\perp}** - or **anti-kt algorithm** [Cacc08] works similar to the kt-algorithm. The only difference is the definition of the distances. Instead of the squared energy, the inverse of the squared energy is used:

$$d_{ij} = \min(E_{\perp i}^{-2}, E_{\perp j}^{-2}) \frac{[(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2]}{R^2} \quad (21)$$

The distance between the particle and the beam axis is defined as:

$$d_{iB} = E_{\perp i}^{-2} \quad (22)$$

The steps of comparing d_{ij} and d_{iB} are the same as in the kt-algorithm. In the algorithms used here, antikt5 and antikt7, R is set to 0.5 or 0.7, respectively.

The **iterative cone algorithm** uses a cone with a fixed radius [Cheka02],

$$R = \sqrt{(\eta_i - \eta_{\text{seed}})^2 + (\phi_i - \phi_{\text{seed}})^2}, \quad (23)$$

around a seed cell to define the jet. Seed cells are deposits in the calorimeter above a certain energy threshold. The algorithm start with the most energetic seed. The jet direction is then defined by the energy-weighted sum of all particle directions inside the cone. As long as the jet direction differs from the direction of the seed, the procedure is repeated, where the current jet direction is the seed direction for the next iteration step. If the cone is stable, it is called a jet and all particles inside the cone are removed. Now the procedure is done for the seed with the highest energy among the remaining seeds. After all stable cones are found, a split and merge procedure is applied to avoid overlapping cones. This algorithm is not infrared safe. In the algorithm iterativeCone5, R is set to 0.5.

The **sisCone algorithm** is a seedless infrared-safe cone algorithm [Salam07]. It looks at each subset of particles enclosed by a circle in the (η, ϕ) -plane. Then, it checks whether the subset changes if the circle is drawn around the p_T -weighted center of the particles. If not, a stable cone, and therefore a jet, is

³This definition is longitudinal invariant. Several other, slightly different, definitions exist.

found. In the distributions shown later, R is set to 0.5 and 0.7, respectively, for sisCone5 and sisCone7. In the next chapters, the distributions of some kinematic variables are shown for both generators. As one criterion for comparison, the ratio $\frac{\sigma_{\text{MadGraph}}}{\sigma_{\text{SHERPA}}}$ was chosen. Each bin content of the MadGraph histogram was divided by the corresponding bin content of the SHERPA histogram. In case of $\frac{0}{0}$, the ratio will be set to zero. The shown error bars indicate statistical errors only. The errors for the ratio are obtained within the divide function for histograms in ROOT. If the distributions would be scaled to the different cross sections, as they were mentioned in Chapter 3.1.3, the ratio of them will be multiplied by $\frac{\sigma_{\text{MadGraph}}}{\sigma_{\text{SHERPA}}} = 0.83$. This factor can be considered as additional systematic uncertainty of $\approx 20\%$, but it is not shown in the histograms. This factor is not important for efficiency calculations. The number of selected events and the number of total events will be scaled with the same factor.

3.2.2 Jet p_T

The histograms of p_T of the jets show how the transverse momentum is distributed among the jets. The jets are defined using the algorithms introduced.

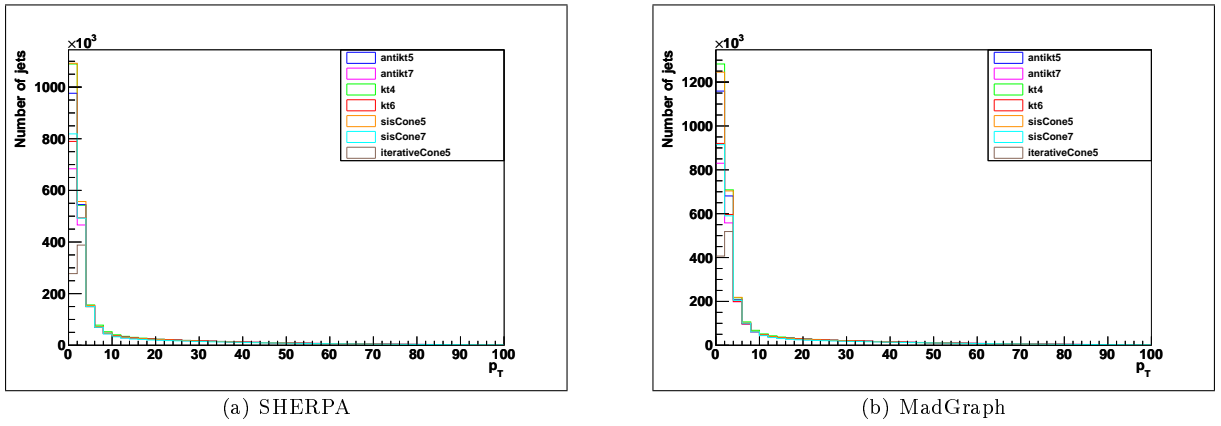


Figure 7: p_T [GeV] distribution of all jets for each generator

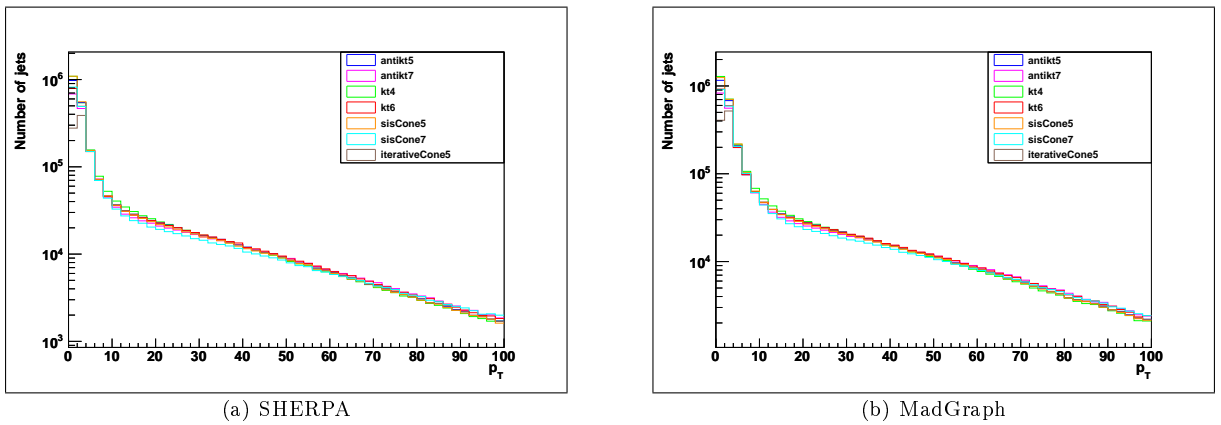


Figure 8: Logarithmic p_T [GeV] distribution of all jets for each generator

The distributions for both generators, shown in Figure 7, look very similar. Here, the number of events is not scaled to a certain luminosity, and all values of p_T are given in GeV. The distributions are dominated by soft jets with small p_T . Therefore, they are drawn in logarithmic scale again in Figure 8. Differences occurring from different jet algorithms are quite small, except that the iterative Cone algorithm finds less jets at low p_T than the other algorithms. A closer look is provided in Figures 9 - 12. There, the number

of events and, therefore the number of jets, is scaled to a luminosity of $L = 20 \text{ pb}^{-1}$, where the cross sections are set to $\sigma_{\text{MadGraph}} = \sigma_{\text{SHERPA}} = 110 \text{ pb}$.

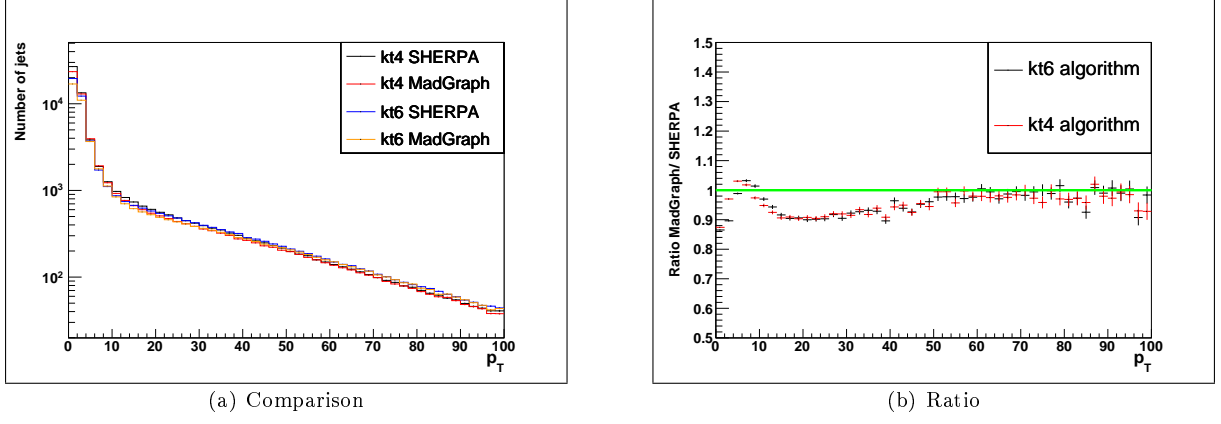


Figure 9: Comparison of the p_T [GeV] distribution of all jets for the kt algorithms

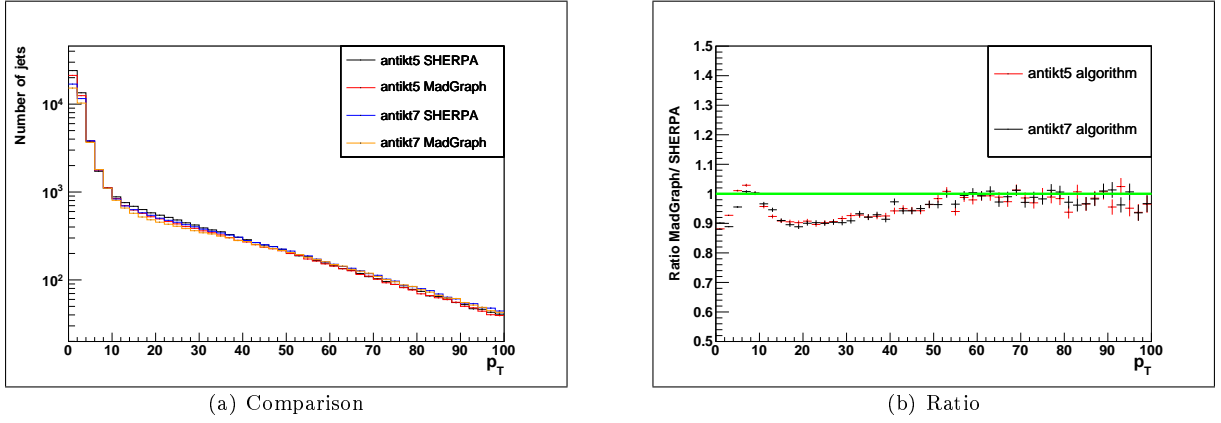


Figure 10: Comparison of the p_T [GeV] distribution of all jets for the antikt algorithms

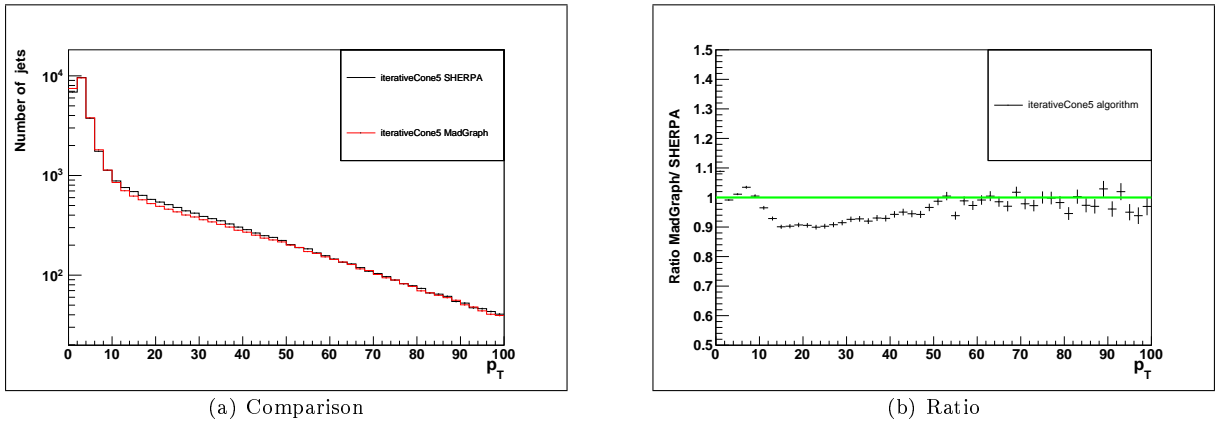


Figure 11: Comparison of the p_T [GeV] distribution of all jets for the iterative Cone algorithm

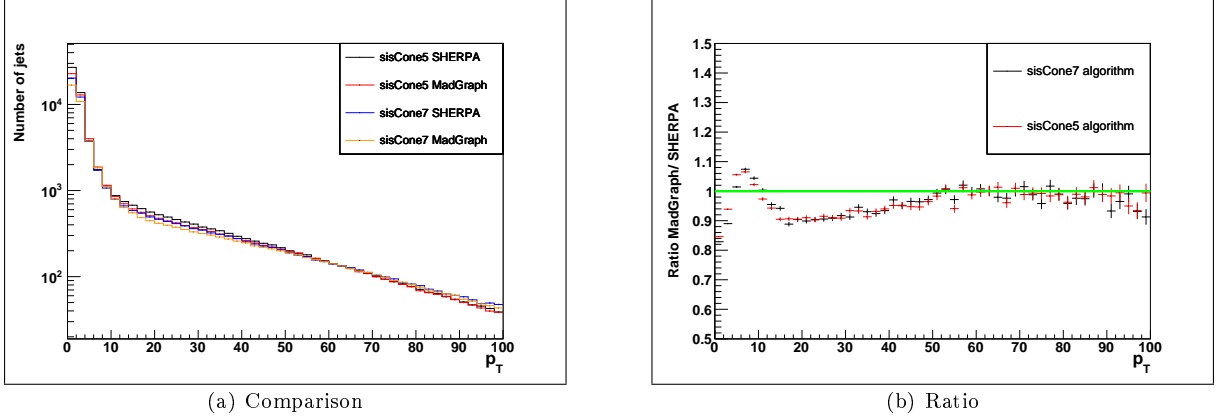


Figure 12: Comparison of the p_T [GeV] distribution of all jets for the sisCone algorithm

One clearly sees in the ratio $\frac{\text{MadGraph}}{\text{SHERPA}}$, shown in Figures 9(b) - 12 (b), that MadGraph has less jets with small p_T . There is a peak in the distribution, at $p_T \approx 10$ GeV, where the ratio is vice versa. Around $p_T = 25$ GeV MadGraph produces again by $\approx 10\%$ less jets. At higher p_T they seem to become equal.

3.2.3 Jet η

Figure 13 shows the pseudorapidity, η , distributions of the jets, defined with different jet algorithms.

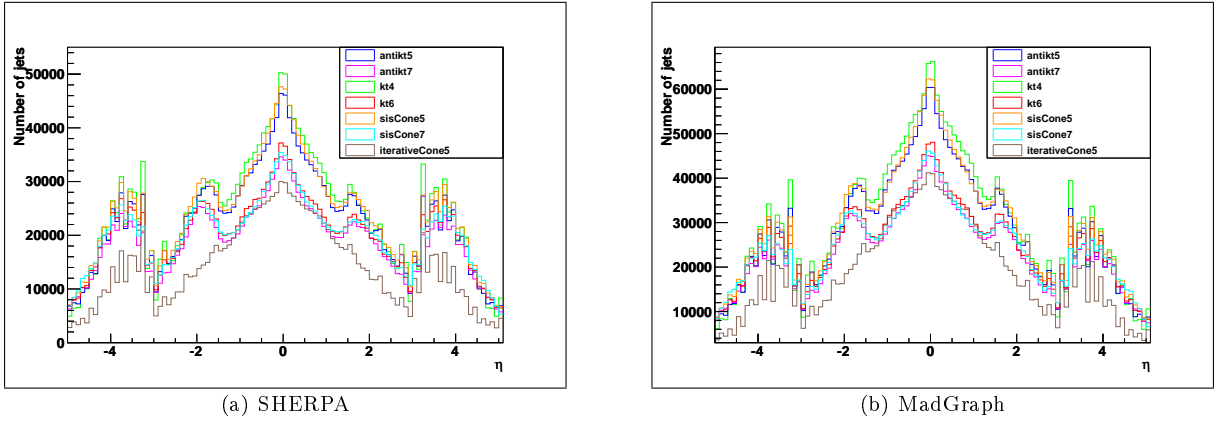
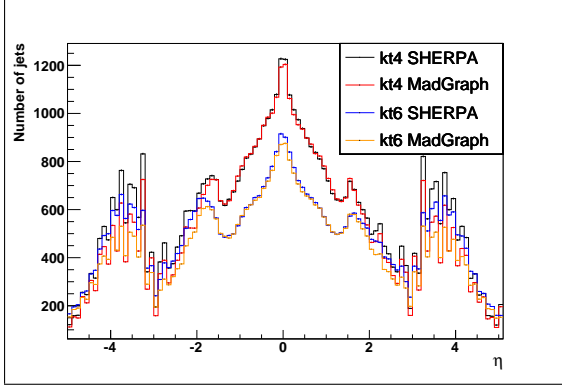
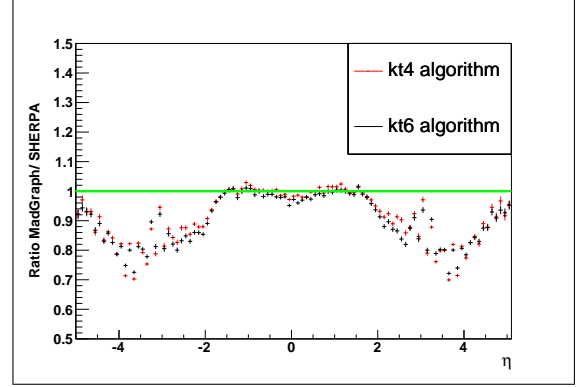


Figure 13: η distribution of all jets for each MC generator

Again, both figures look similar. One can clearly see that we have more jets from jet algorithms with smaller values of R (e.g. kt4 or sisCone5) than for jet algorithms with larger values of R (e.g. kt6 or sisCone7). Particles with a distance of $R = 0.6$ may be considered as two jets in sisCone5, while they could be considered as one jet with sisCone7. The dips around $|\eta| = 3$ and $|\eta| = 1.5$ are due to regions with worse resolution in the detector. This is, for example, the region where the endcap is connected to the barrel. A more detailed comparison is given in Figures 14 - 17.

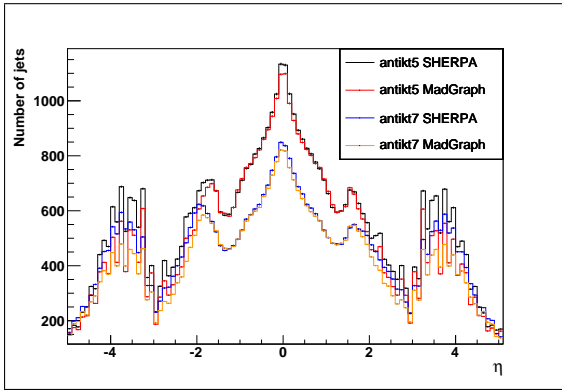


(a) Comparison

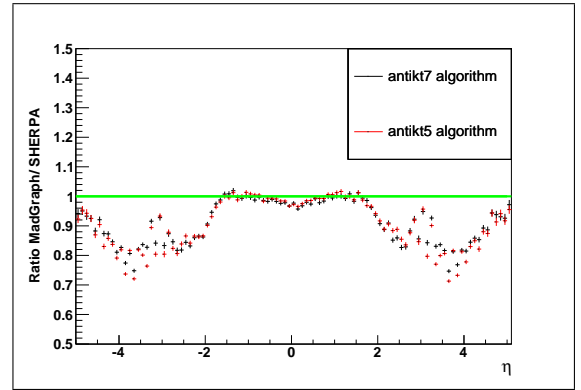


(b) Ratio

Figure 14: Comparison of the η distribution of all jets for the kt algorithms

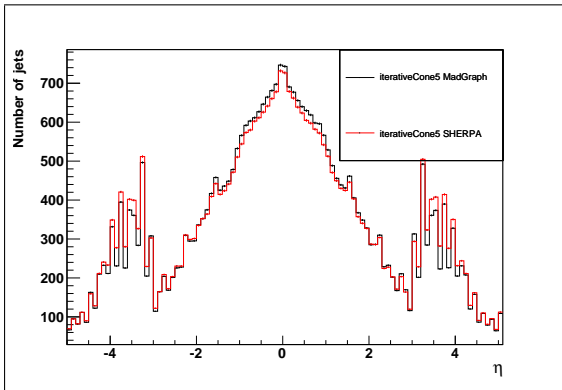


(a) Comparison

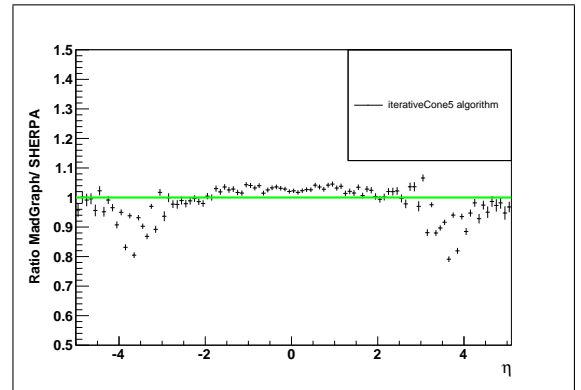


(b) Ratio

Figure 15: Comparison of the η distribution of all jets for the antikt algorithms



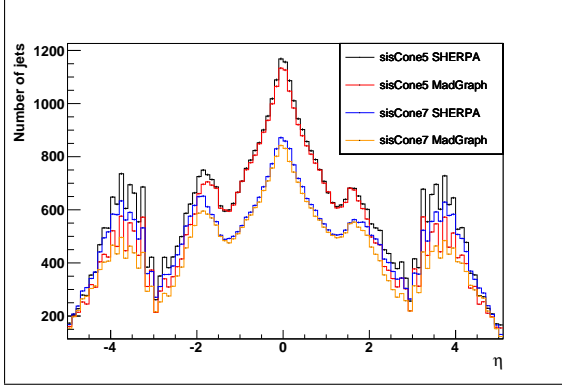
(a) Comparison



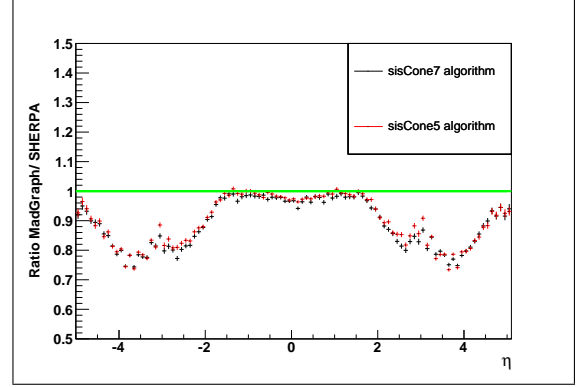
(b) Ratio

Figure 16: Comparison of the η distribution of all jets for the iterative Cone algorithm

In the ratio $\frac{\text{MadGraph}}{\text{SHERPA}}$ (Figures 14(b) - 17 (b)), one can see that in the central region, where $|\eta| < 1.5$, the generators give almost equal results for all algorithms. This pseudorapidity corresponds to an angle of $\varphi = 25.2^\circ$ with respect to the beam axis, which covers the major part of the detector. One exception is the iterative Cone algorithm. It yields a good ratio of about 1.03 in the range $|\eta| < 3$. The ratio between the two generators drops significantly in the outer regions. SHERPA produces 20 % more jets there. This is seen for all algorithms and could therefore be a systematic effect of the different generators.



(a) Comparison

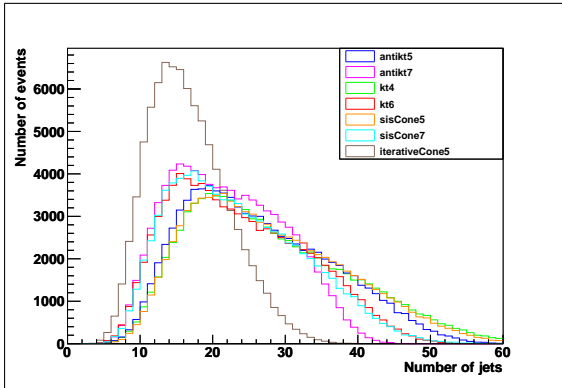


(b) Ratio

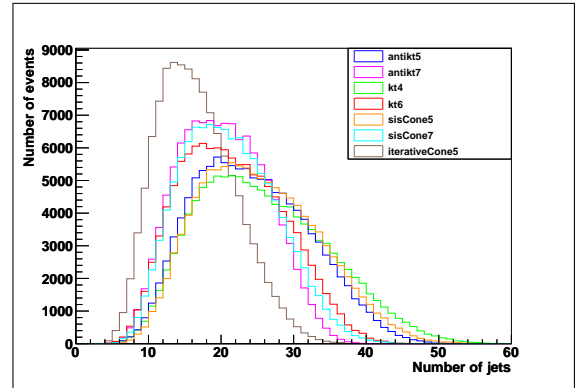
Figure 17: Comparison of the η distribution of all jets for the sisCone algorithm

3.2.4 Number of Jets

The distribution of the number of jets for the different algorithms is shown in Figure 18.



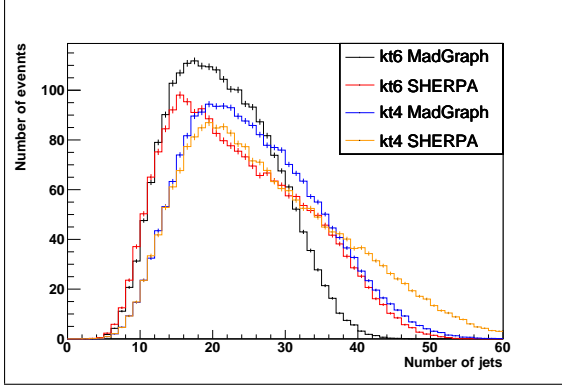
(a) SHERPA



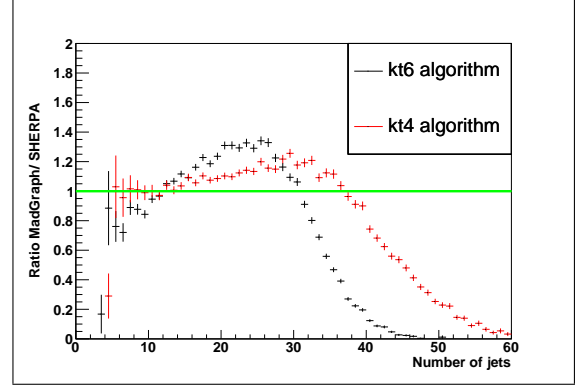
(b) MadGraph

Figure 18: Number of jets for both generators

In these distributions, one can clearly see the differences in the jet finding algorithms. All algorithms with small R find more jets than their corresponding partners with larger R . One exception is the iterative Cone algorithm which finds less jets than all the other algorithms. This could be because the iterative Cone algorithm is not infrared-safe [Cheka02]. The final state of $t\bar{t}$ production is characterized by a large number of particles. Due to their production, e.g. during PS, the iterative cone algorithm will not find as many stable jets as other algorithms. The distributions also indicate that SHERPA produced more jets than MadGraph. It will be more clear in the comparison plots.

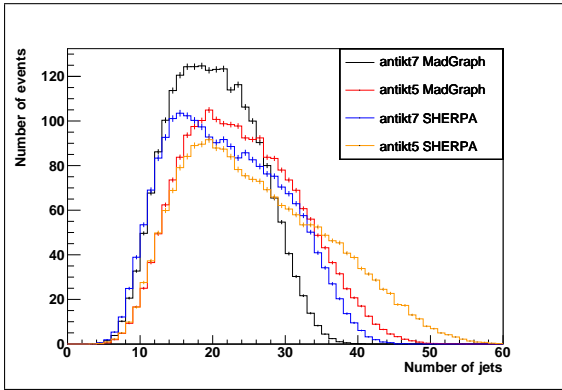


(a) Comparison

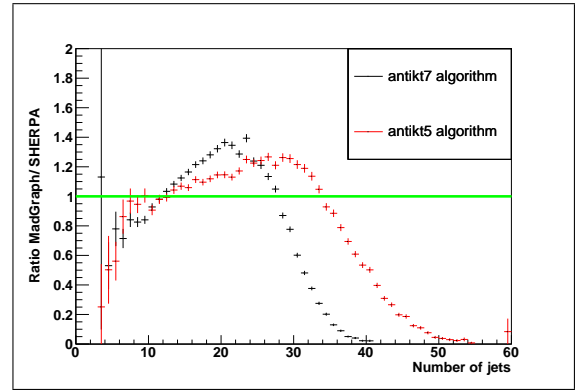


(b) Ratio

Figure 19: Comparison of the number of jets distribution for the kt algorithms

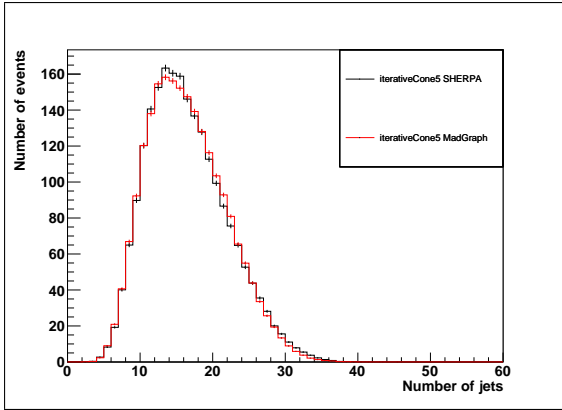


(a) Comparison

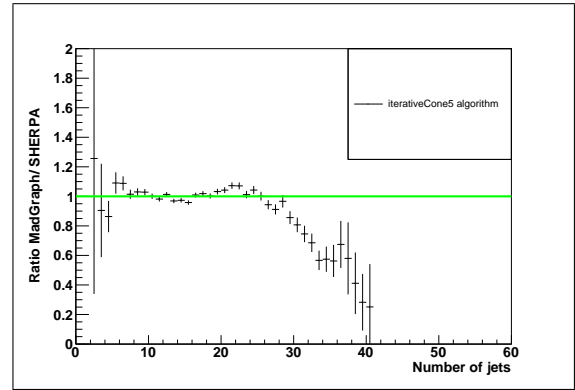


(b) Ratio

Figure 20: Comparison of the number of jets distribution for the antikt algorithms



(a) Comparison



(b) Ratio

Figure 21: Comparison of the number of jets distribution for the iterative Cone algorithm

Figures 19(b) - 22(b) show the ratio $\frac{\text{MadGraph}}{\text{SHERPA}}$ for the distribution of the number of jets per event for different jet algorithms. One can clearly see that the distribution for the number of jets differ significantly. For a small number of jets, MadGraph has less entries in the bins than SHERPA. For about ten to thirty jets it is vice versa until it changes again around 30 jets. SHERPA seems to produce more events with a higher number of jets. As it was shown in Chapter 3.2.2, many of them are soft jets with small p_T . This

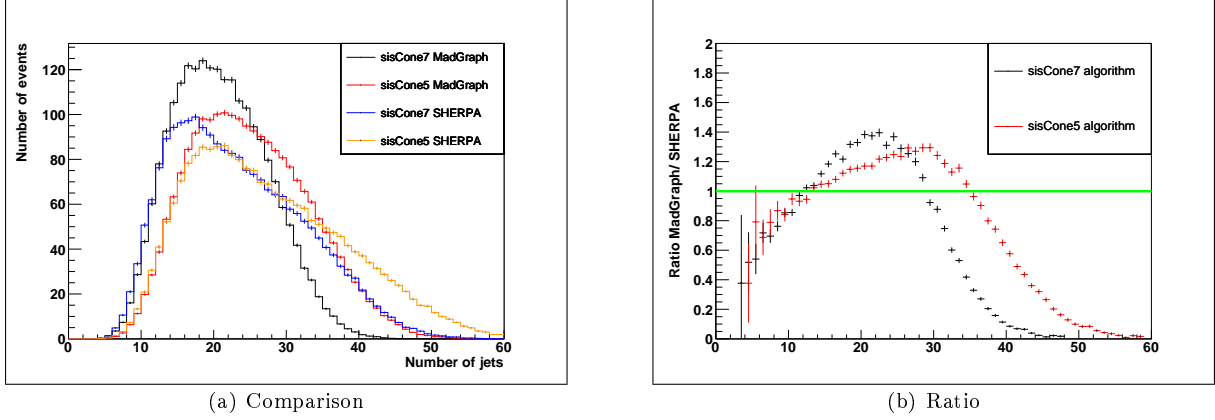


Figure 22: Comparison of the number of jets distribution for the sisCone algorithm

may be caused by the different matching algorithms. The PS in SHERPA produces more particles which will be converted to jets. Another reason could be a different handling of the underlying event⁴.

3.2.5 Sum of E_T for Jets

For each event, one can sum all transverse energies stored in jets. In Figure 23, one can see the distribution obtained for each generator. The comparison of the sum over E_T distributions for the different algorithms is given in Figures 24 - 27 .

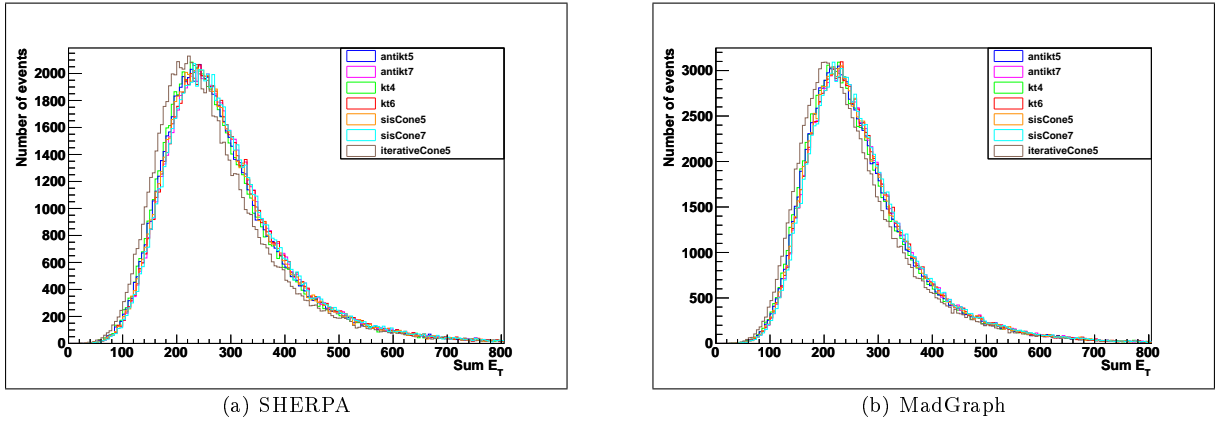
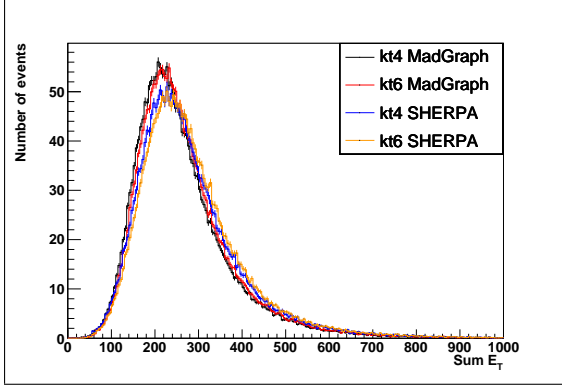
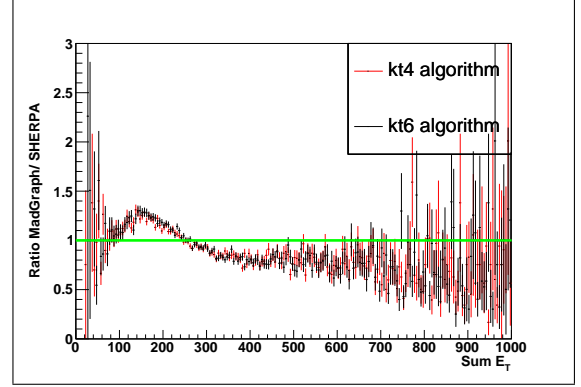


Figure 23: Sum E_T [GeV] for each MC generator

⁴The underlying event is the evolution of the beam remnants. This is a background process for each event

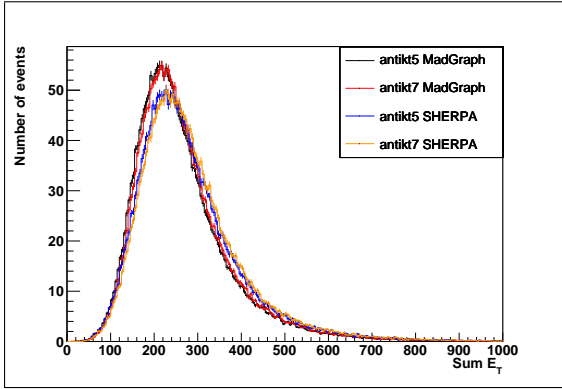


(a) Comparison

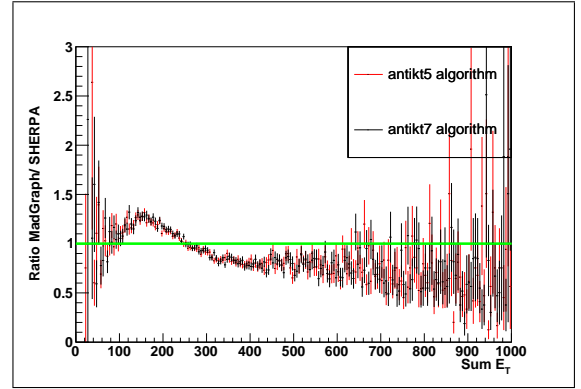


(b) Ratio

Figure 24: Comparison of the Sum E_T [GeV] distributions for the kt algorithms

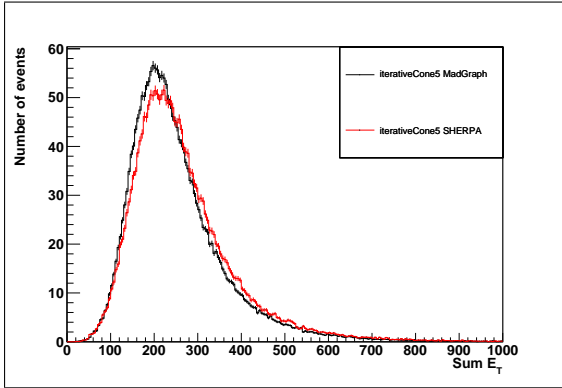


(a) Comparison

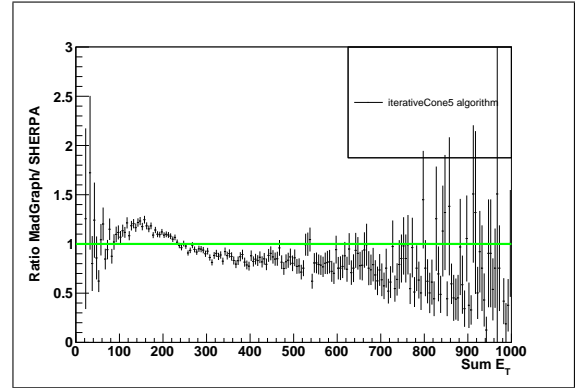


(b) Ratio

Figure 25: Comparison of the Sum E_T [GeV] distributions for the antikt algorithms



(a) Comparison



(b) Ratio

Figure 26: Comparison of the Sum E_T [GeV] distributions for the iterative Cone algorithm

Again, the shapes of the distributions look similar. For small values of Sum E_T , the uncertainties are quite large. For values in the range $100 \text{ GeV} < \text{Sum} E_T < 200 \text{ GeV}$ MadGraph gives more events, where in the range above $\text{Sum} E_T > 200 \text{ GeV}$ SHERPA has more events. This may again be caused by SHERPA finding more jets.

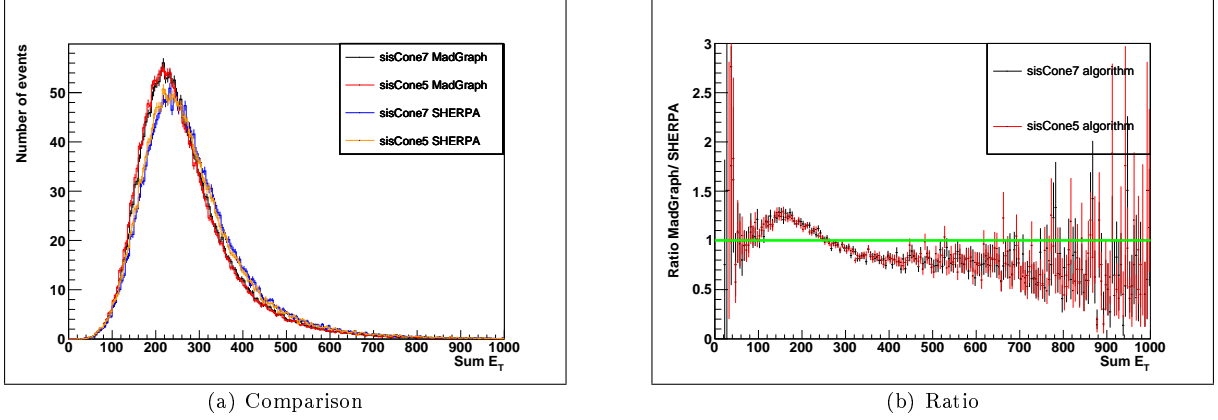


Figure 27: Comparison of the Sum E_T [GeV] distributions for the sisCone algorithm

3.2.6 Leading Jet p_T

The distribution of p_T was shown in Chapter 3.2.2 for all jets. In this section, we want to have a closer look at the p_T distribution of the leading jet. The leading jet of an event is the jet with the largest p_T . Figure 28 shows the histograms for SHERPA and MadGraph separately with the different jet algorithms.

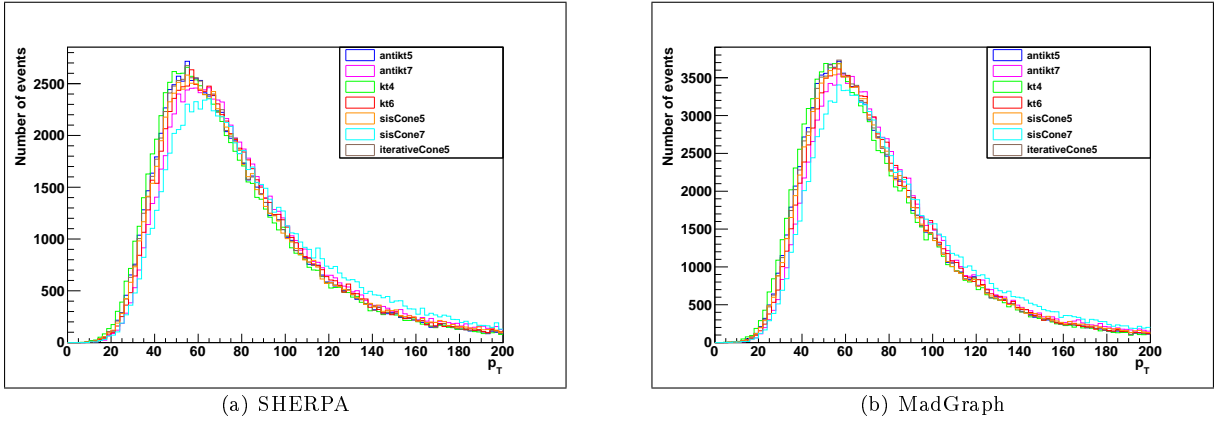
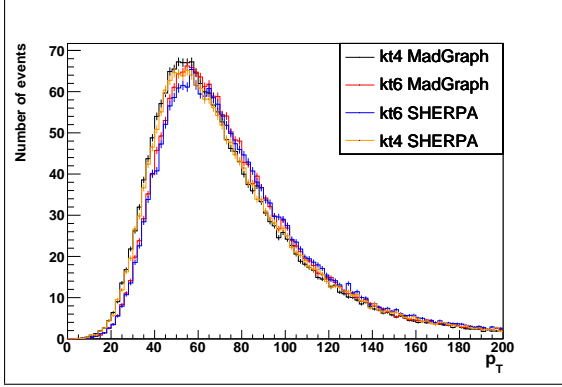
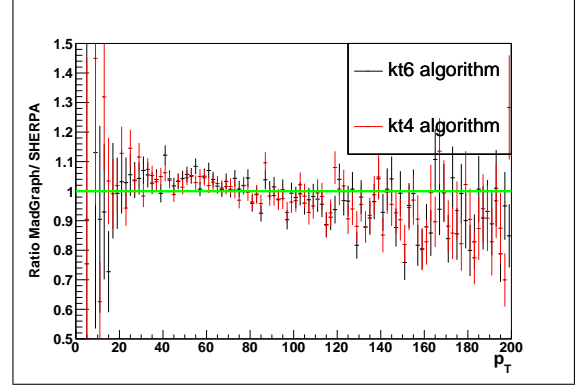


Figure 28: p_T [GeV] of the leading jet for both generators

The shapes look again similar. We can see a peak in both distributions at $p_T = 60$ GeV.

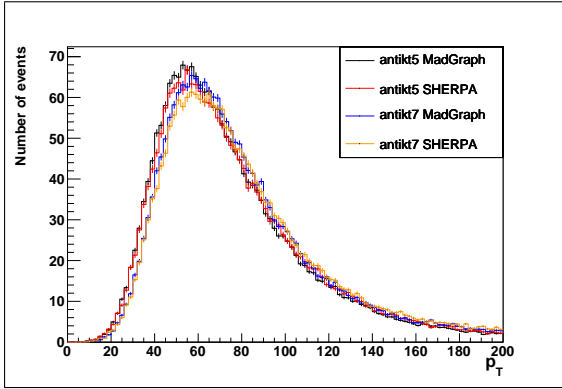


(a) Comparison

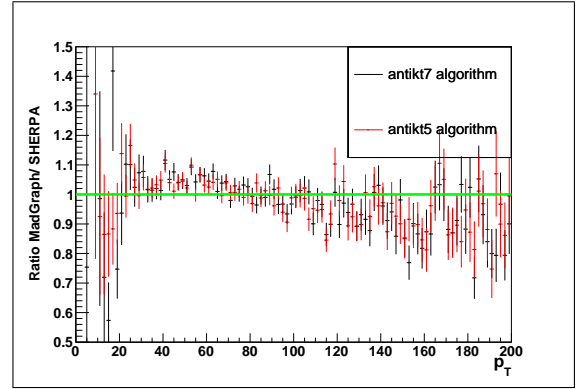


(b) Ratio

Figure 29: Comparison of the p_T [GeV] distribution of the leading jet for the kt algorithms

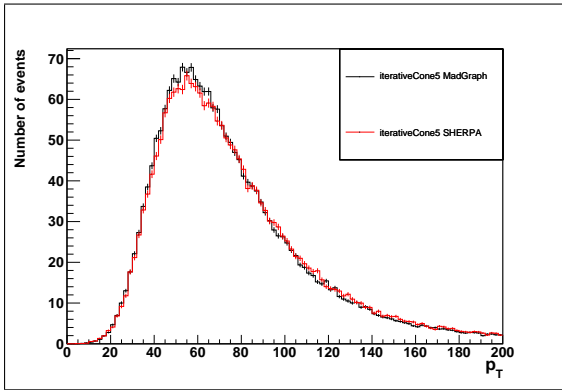


(a) Comparison

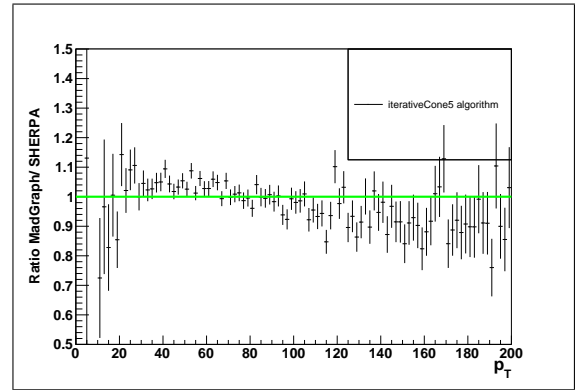


(b) Ratio

Figure 30: Comparison of the p_T [GeV] distribution of the leading jet for the antikt algorithms



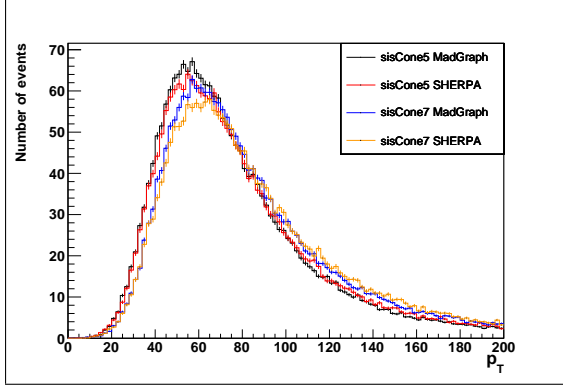
(a) Comparison



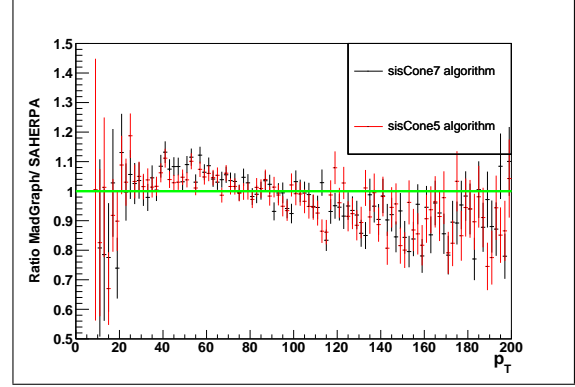
(b) Ratio

Figure 31: Comparison of the p_T [GeV] distribution of the leading jet for the iterative Cone algorithm

The ratio $\frac{\text{MadGraph}}{\text{SHERPA}}$ has a clear negative slope. At low p_T , MadGraph produces more leading jets than SHERPA. Around 100 GeV they become equal. SHERPA produces more leading jets with high p_T . This effect is also seen for all jet algorithms and, therefore, it might be a systematic effect caused by the different generators.



(a) Comparison

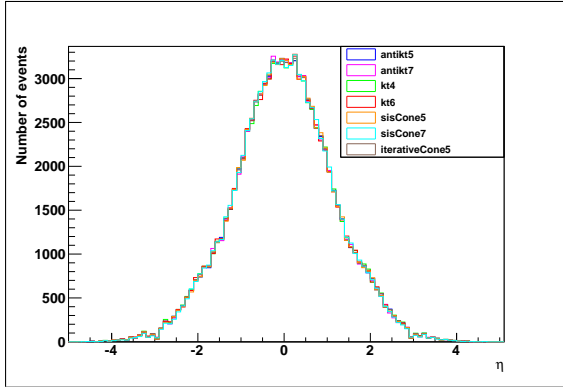


(b) Ratio

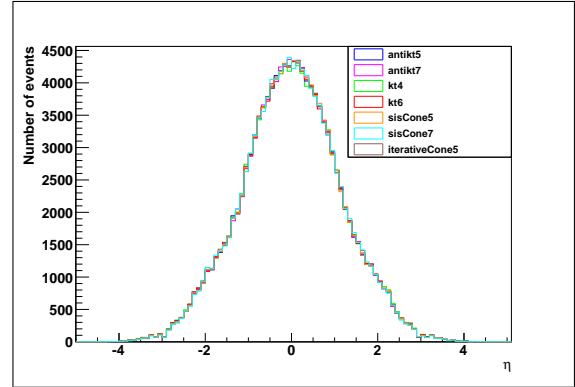
Figure 32: Comparison of the p_T [GeV] distribution of the leading jet for the sisCone algorithm

3.2.7 Leading Jet η

One can also have a look at the η distribution of the leading jet. It is shown in Figure 33.



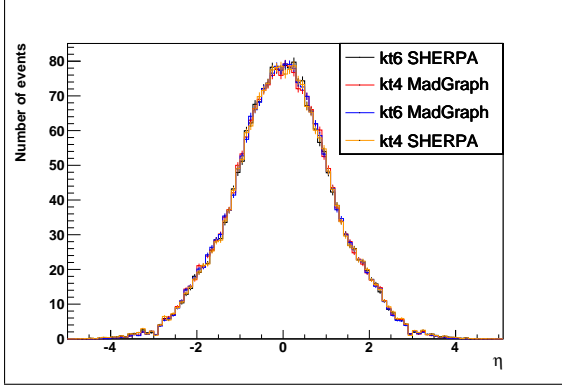
(a) SHERPA



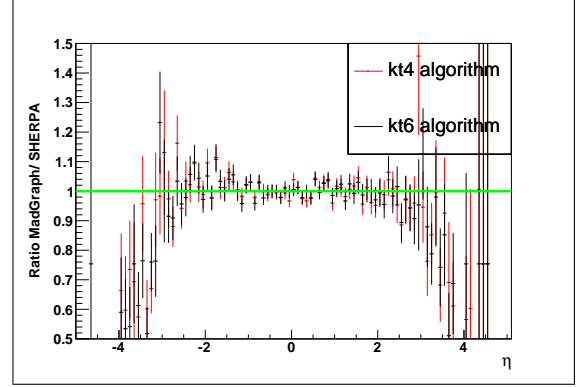
(b) MadGraph

Figure 33: η of the leading jet

We can see almost no difference between the different jet finding algorithms. The distribution has a peak at $\eta = 0$. This means that the leading jet is produced at large angles with up to $\varphi = 90^\circ$ with respect to the beam axis.

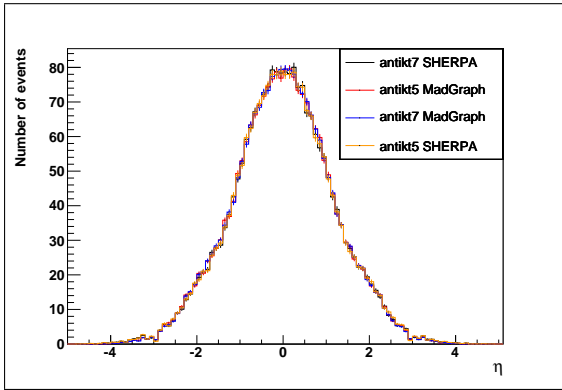


(a) Comparison

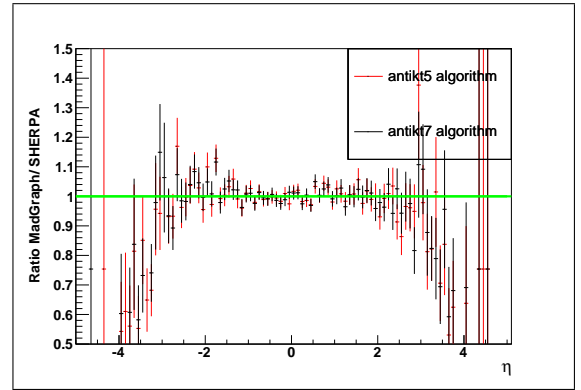


(b) Ratio

Figure 34: Comparison of the η distribution of the leading jet for the kt algorithms

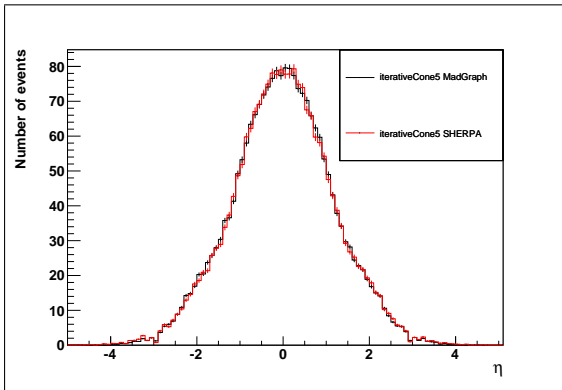


(a) Comparison

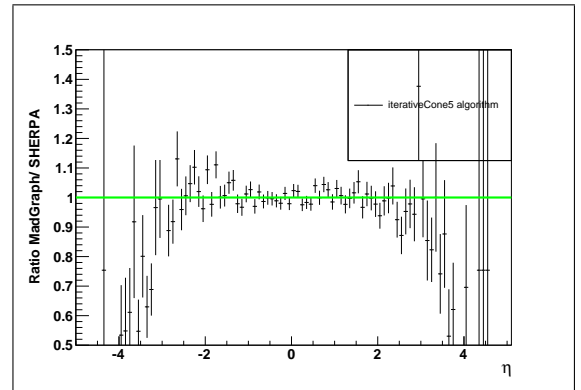


(b) Ratio

Figure 35: Comparison of the η distribution of the leading jet for the antikt algorithms



(a) Comparison



(b) Ratio

Figure 36: Comparison of the η distribution of the leading jet for the iterative Cone algorithm

Again, histograms (b) of Figures 34 - 37 show the ratio $\frac{\text{MadGraph}}{\text{SHERPA}}$. The ratio is almost one for all jet algorithms in the range $|\eta| < 3$. From this and from the results of the previous chapter (3.2.6), we can conclude that the impact on the analysis is not as pronounced as suggested by the other distributions because for the analysis mostly jets with high p_T from the central region $|\eta| < 2.5$ are chosen.

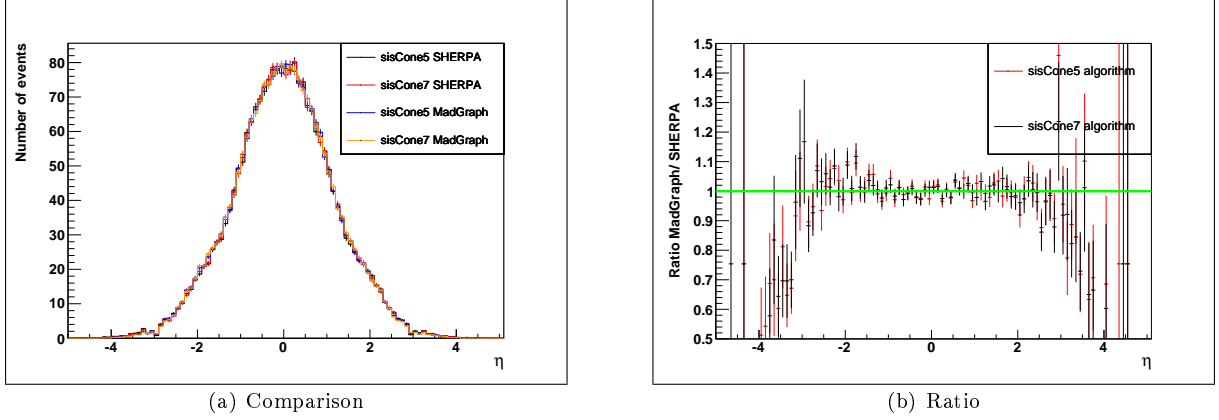


Figure 37: Comparison of the η distribution of the leading jet for the sisCone algorithm

3.2.8 Muon p_T

The kinematic variables p_T and η of the muon are also important. They are used to identify semimuonic $t\bar{t}$ events. This procedure is described in more detail in Chapter 3.3. Figure 38 shows the distribution of p_T for all muons in the event.

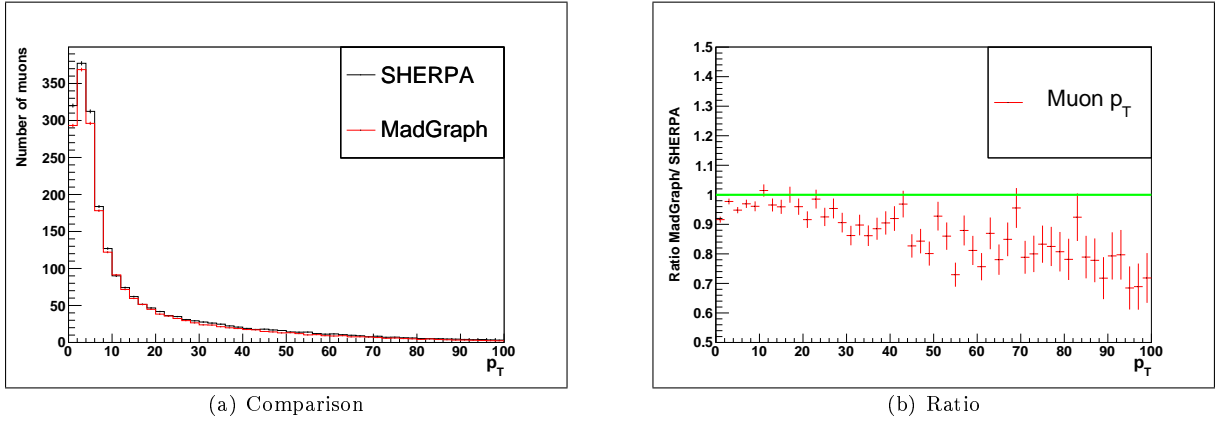


Figure 38: p_T [GeV] of the muon

The shapes of both distributions strongly resemble each other, but, from the ratio (Figure 38b), we see that MadGraph produces about 20 % less muons at high p_T . This will be important for the selection of semimuonic $t\bar{t}$ events in Chapter 3.3.

3.2.9 Muon η

The η distribution for the muons in the events is shown in Figure 39.

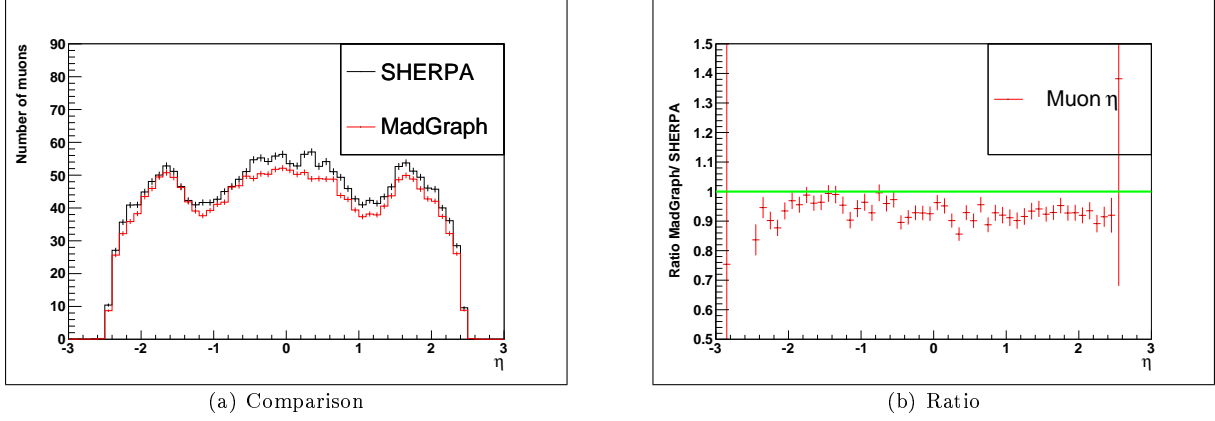


Figure 39: η of the muon

The conclusions from the η distribution for the muons are the same as for the p_T distribution. MadGraph produces less muons, even at equal cross sections. If we scale with the given cross sections, the difference would be even bigger.

3.2.10 Missing Energy in Transverse Direction

The determination of the missing energy is important as it allows conclusions about the kinematics of the neutrinos. They leave the detector without any interaction. Small values of missing energy, in the range 1 – 2 GeV, can also come from measurement uncertainties in the detector. The distribution of the missing energy is shown in Figure 40.

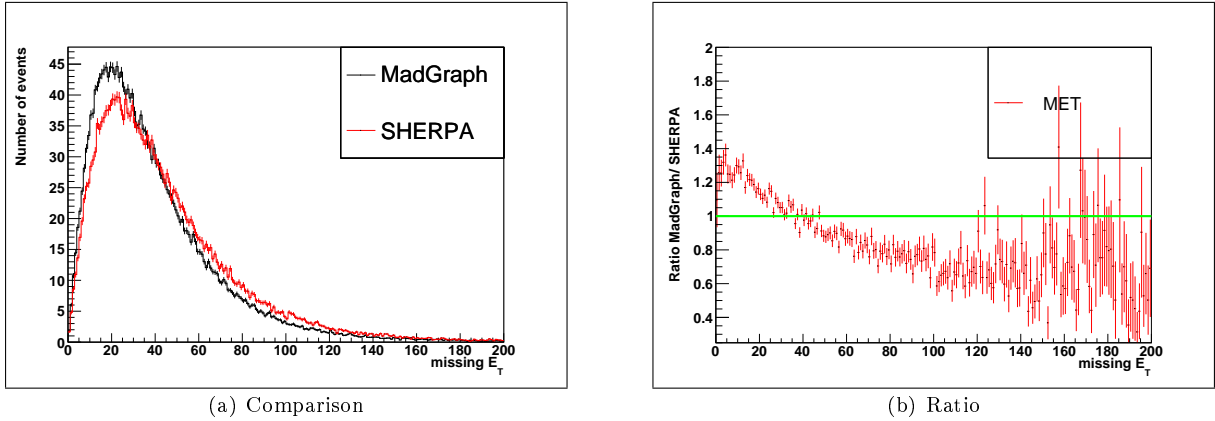


Figure 40: Missing E_T [GeV] for both generators

For values of the missing energy larger than 40 GeV, MadGraph produces less events. This would agree with the already obtained result that MadGraph yields less particles leading to less jets.

3.3 Selection Algorithm

An algorithm that can be used to select $t\bar{t}$ events with one muon in the final state, hereafter referred to as semimuonic channel, was described in [TOP-09-003]. In our analysis, the selection is performed and described according to that note. This channel is chosen because of its low trigger thresholds and clean signature. The events are selected according to the following steps.

1. The event contains at least one global muon.
2. These muons must have $p_T > 20$ GeV and $|\eta| < 2.1$.
3. The relative isolation of the muons, defined as $\text{RelIso} = \frac{E_{\text{CALO}}(\text{Iso}) + p_T(\text{Tracker, Iso})}{p_T}$, must be smaller than 0.05. The impact parameter of them, d_0 , must be smaller than 0.037 mm. For the whole track, $\frac{\chi^2}{\text{ndf}} < 10$ must be valid as well as $\text{EcalVeto} \leq 4$ and $\text{HcalVeto} \leq 6$.
4. The event contains at least four jets, defined with sisCone5 , that have $p_T > 30$ GeV and $|\eta| < 2.4$.
5. The number of muons with the given properties must be exactly one.
6. Events must not contain any electron with $p_T > 15$ GeV and $|\eta| < 2.5$.
- (7. The missing energy must be at least 30 GeV.)

3.3.1 Explanation of the Algorithm

A muon is called a global muon if a reconstructed track in the tracker matches a reconstructed track in the muon chambers. This must be valid for muons coming from the decay of a W -boson because such decays happen shortly after production at the IP. The p_T cut of the muons is required in order to suppress background originating from QCD. Muons from W^\pm have a high p_T because of the high mass of the W^\pm . The η cut of the muons is motivated by the acceptance of the muon trigger. The relative isolation, RelIso , is a combined criterion that is used to suppress the background coming from muons in jets. These muons are mostly produced in b/c -hadron decays. $E_{\text{CALO}}(\text{Iso})$ is defined by the sum of all energy deposits in the calorimeter within a range of $R < 0.3$ around the muon direction. The energy of the muon itself is excluded from the calculation. $p_T(\text{Tracker, Iso})$ is defined in a similar way. It is the sum over all transverse momenta inside a cone ($R = 0.3$) around the muon, excluding the muon momentum. The impact parameter, d_0 , gives the distance from the origin of the muon path to the IP. This value is set very small because the muons from W^\pm decays are produced almost instantly and, therefore, at the IP. The number of muons originating from hadron decays, such as π^\pm , is reduced. The quality of the reconstructed track is described by $\frac{\chi^2}{\text{ndf}}$. The value χ^2 is the sum of all quadratic differences of measured quantities (hits in the tracker) to the expected quantities (the reconstructed track), normalised to the expected value [Bohm10].

$$\chi^2 = \sum_i \frac{(x_{i,\text{measured}} - x_{\text{expected}})^2}{x_{\text{expected}}} \quad (24)$$

The denominator, ndf , labels the number of degrees of freedom for the track reconstruction. The calorimeter vetos, EcalVeto and HcalVeto , give a veto to $E_{\text{CALO}}(\text{Iso})$ because muons are expected to behave like minimum ionizing particles. Electrons coming from a W^\pm decay generate a shower in the ECal, depositing all their energy. Their contribution will be suppressed with this cut. The semimuonic decay channel of $t\bar{t}$ events contains at least four jets. Two of them originate from the decay of a W -boson, the other two come from the decay of b and \bar{b} . The event can contain additional jets from parton showering or processes of higher order. The η cut is motivated by the acceptance of the silicon tracker. The p_T cut is set to reduce the fraction of beam remnant jets. The sisCone5 algorithm is used to define the jets. The use of this algorithm is encouraged by the CMS collaboration [CMS-PAS]. The algorithm is collinear- and infrared-safe to all orders of perturbative QCD. Events with more than one muon are rejected in order to reduce the contributions from the dileptonic decay channel and from Z +jets. Events with an isolated electron are rejected as well, as the semielectronic decay channel is also a background process to the desired selection. In addition, a minimum missing energy in transverse direction of 30 GeV may be required in order to enrich events with a neutrino. However, measurements of missing energy are not used in the first months of the LHC operation because calorimeters must be calibrated well to get reasonable results. So step 7 is optional at present.

3.3.2 Cut Flow Tables

The selection steps, described in the previous chapter, were applied to the generated and reconstructed samples. In Table 5, one can see how many events survived each selection step if the cross sections are set to $\sigma_{t\bar{t}} = 110$ pb for both programs. These tables are called cut flow tables. In Table 6, one can see the cut flow scaled to the cross sections obtained from the generators. There are two different numbers listed for each MC generator. The first column specifies the number of events scaled to $L = 20 \text{ pb}^{-1}$. The second column states the number of events in the generated sample.

Step	SHERPA Events (20 pb ⁻¹)	Events	MadGraph Events (20 pb ⁻¹)	Events
0. before selection	2200	102000	2200	135305
1. Global Muons	1085.79 ± 4.84	50341	1017.62 ± 4.07	62586
2. $\eta < 2.1, p_T > 20$	538.96 ± 3.41	24988	469.79 ± 2.76	28893
3. muon-ID	405.58 ± 2.96	18804	344.95 ± 2.37	21215
3. relIso<0.05	305.39 ± 2.57	14159	259.93 ± 2.06	15986
4. N(jets)>=1	280.18 ± 2.46	12990	240.27 ± 1.98	14777
4. N(jets)>=2	197.76 ± 2.07	9169	167.96 ± 1.65	10330
4. N(jets)>=3	92.49 ± 1.41	4288	71.15 ± 1.08	4376
4. N(jets)>=4	27.24 ± 0.77	1263	17.46 ± 0.53	1074
5. N(μ)=1	26.92 ± 0.76	1248	17.32 ± 0.53	1065
6. el. veto	22.95 ± 0.70	1064	15.50 ± 0.50	953
7. MET > 30	19.41 ± 0.65	900	12.73 ± 0.45	783

Table 5: Cut flow table for the selection algorithm, $\sigma_{\text{SHERPA}} = \sigma_{\text{MadGraph}} = 110$ pb

Step	SHERPA Events (20 pb ⁻¹)	Events	MadGraph Events (20 pb ⁻¹)	Events
0. before selection	2510	102000	2082	135305
1. Global Muons	1238.78 ± 5.52	50341	963.04 ± 3.85	62586
2. $\eta < 2.1, p_T > 20$	614.90 ± 3.89	24988	444.59 ± 2.62	28893
3. muon-ID	462.73 ± 3.37	18804	326.44 ± 2.24	21215
3. relIso<0.05	348.42 ± 2.93	14159	245.984 ± 1.95	15986
4. N(jets)>=1	319.66 ± 2.80	12990	227.38 ± 1.87	14777
4. N(jets)>=2	225.63 ± 2.36	9169	158.95 ± 1.56	10330
4. N(jets)>=3	105.52 ± 1.61	4288	67.34 ± 1.02	4376
4. N(jets)>=4	31.08 ± 0.87	1263	16.53 ± 0.50	1074
5. N(μ)=1	30.71 ± 0.87	1248	16.39 ± 0.50	1065
6. el. veto	26.18 ± 0.80	1064	14.66 ± 0.48	953
7. MET > 30	22.15 ± 0.74	900	12.05 ± 0.43	783

Table 6: Cut flow table for the selection algorithm, $\sigma_{\text{SHERPA}} = 125.5$ pb, $\sigma_{\text{MadGraph}} = 104.1$ pb

The efficiency of the selection is then given by $\epsilon = \frac{\text{survived events}}{\text{all events}}$. In Table 7, one can see the obtained efficiencies for SHERPA and MadGraph. The efficiency is independent of the chosen cross sections because the number of survived events and the total number of events will be scaled by the same factor.

	SHERPA	MadGraph	rel. Difference ($\frac{\text{SHERPA}-\text{MadGraph}}{\text{SHERPA}}$)
ϵ before MET cut	1.04 %	0.70 %	32.5 %
ϵ after MET cut	0.88 %	0.58 %	34.4 %

Table 7: Calculated efficiencies of the selection algorithm

The efficiencies differ by about 33 %. The reason could be the lower number of particles produced by MadGraph. Therefore, we get less events with one muon and four jets above the thresholds. Additionally, one can obtain the fraction of survived events per step. They are listed in Table 8.

Step	SHERPA	MadGraph	Difference
1. Global Muons	49.4%	46.3 %	3.1 %
2. $\eta < 2.1, p_T > 20$	49.6 %	46.2 %	3.4 %
3. muon-ID	75.3 %	73.4 %	1.9 %
3. relIso<0.05	75.3 %	75.4 %	-0.1 %
4. N(jets)>=1	91.7 %	92.4 %	-0.7 %
4. N(jets)>=2	70.6 %	69.9 %	0.7 %
4. N(jets)>=3	46.8 %	42.4 %	4.4 %
4. N(jets)>=4	29.5 %	24.5 %	5.0 %
5. N(μ)=1	98.8 %	99.2 %	-0.4 %
6. el. veto	85.3 %	89.5 %	-4.2 %
7. MET > 30	84.6 %	82.2 %	2.4 %

Table 8: Fraction of surviving events per selection step

In this fractions, we can see that in most of the steps a little less events from MadGraph pass. The largest difference between MadGraph and SHERPA is in the kinematics cut of the muon (step 2) and in the number-of-jets cuts (step 4). This is a direct consequence of the distributions of these variables, as it was shown in the previous chapter. Another interesting fact is that due to the lower number of generated particles in the MadGraph sample, we have more events with exactly one muon and more events that pass the electron veto step than in SHERPA.

3.3.3 Control plots

In this chapter, a few kinematic distributions after the selection algorithm are shown. In Figure 41, we can see the number of jets for events that survived the selection step 1.

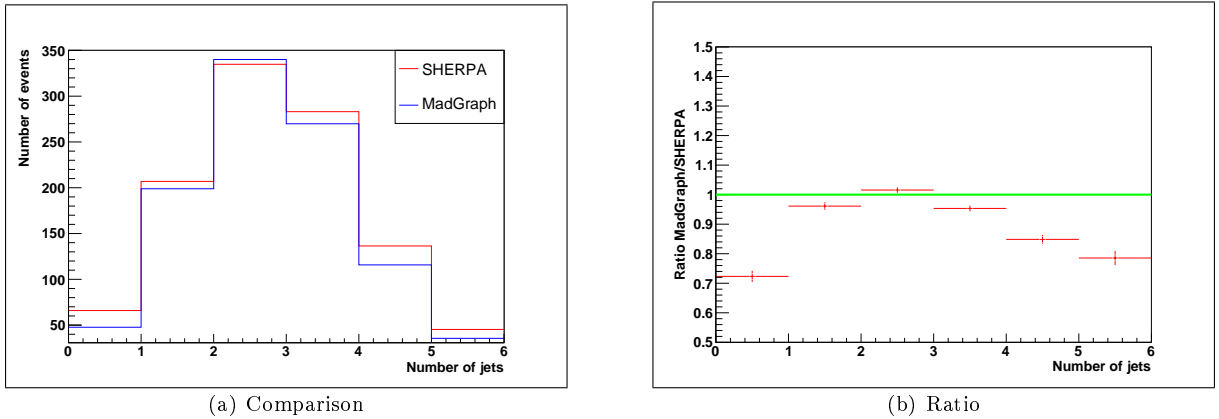


Figure 41: Number of jets after selection step 1

As one can see, only a small fraction of the sample has at least four jets and will therefore pass selection step 4.

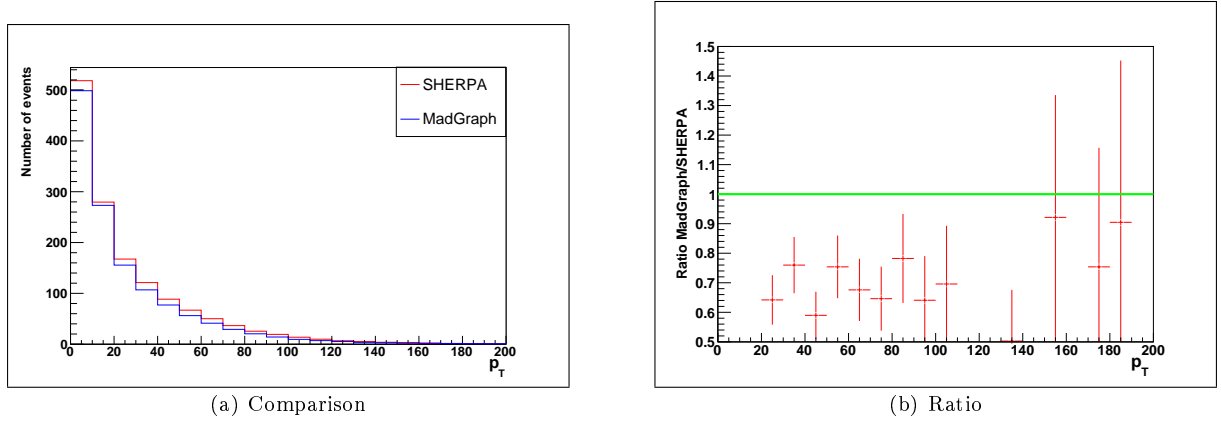


Figure 42: p_T [GeV] of the muon after step 1

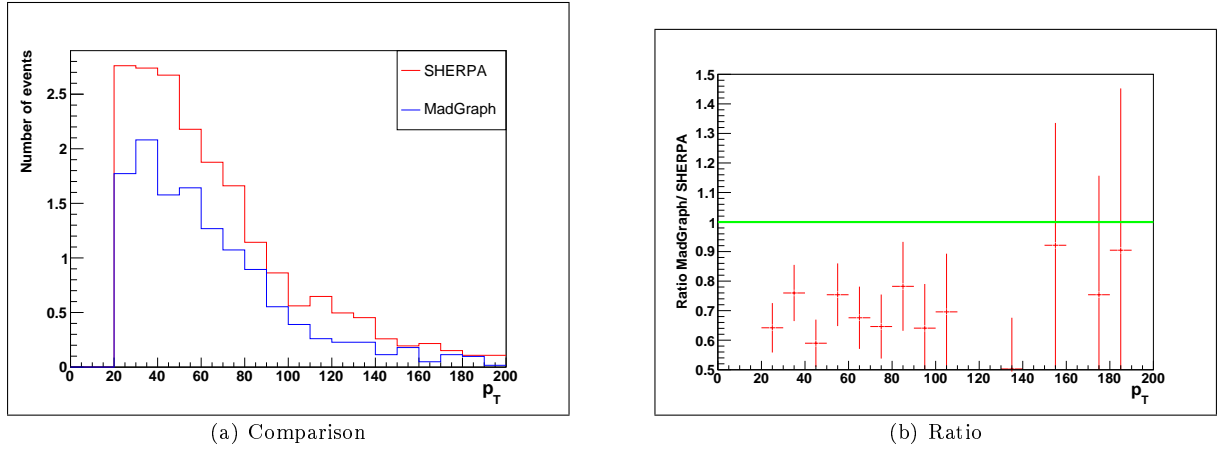


Figure 43: p_T [GeV] of the muon after step 7

The Figures 42 and 43 show the distribution of p_T for the muons after step 1 and after all steps, including the MET cut. The control plots show exactly what we saw in the cut flow tables. In MadGraph less events will survive the algorithm.

4 Conclusion

Using the two Monte Carlo event generators SHERPA and MadGraph the events for the process

$$pp \rightarrow t\bar{t} + X \text{ jets}, \quad X \leq 2$$

are generated in all decay channels and processed with the FastSimulation of the CMS detector. Using almost identical settings in the generators, we found that SHERPA produces more particles than MadGraph. This leads to a higher multiplicity of the events. This could be caused by the different approaches to match parton showering and matrix element calculations.

The obtained cross sections of both generators differ by about 20 pb. While SHERPA gives

$$\sigma_{\text{SHERPA}} = 125.5 \text{ pb},$$

we obtain from MadGraph

$$\sigma_{\text{MadGraph}} = 104.1 \text{ pb}.$$

This difference of about 20 % can be explained with SHERPA using some next to leading order corrections while MadGraph uses only leading order calculations.

The agreement of the kinematic distributions between SHERPA and MadGraph is relatively good in the central region of the detector. There, they differ by about 10 % or less. Good examples are the p_T and η distributions of the leading jet.

As an additional comparison, the performance of different jet finding algorithms was shown. The shapes for all distributions look comparable for the different algorithms. One exception was the iterative cone algorithm. Sometimes it showed a different shape than the other algorithms and the accordance between SHERPA and MadGraph was best for it. Due to its infrared- and collinear unsafety it is not used for analysis. Due to the similar shapes of the distributions, we can conclude that the obtained differences are not caused by the jet algorithms.

In the selection algorithm for semimuonic events, we found that each selection step contributes a small error, leading to a relative large error in the total efficiency ϵ . In this thesis, it resulted in an relative shift of 34 % after the seven steps of the selection of semimuonic $t\bar{t}$ events. This difference could have a large impact on the cross section measurements and especially in the obtained uncertainty. A solution to that could be, to simulate the whole event including all background processes with both programs. The obtained kinematic distributions could be compared to real data and the MC generator that fits best could be chosen for further analysis of the selection algorithms. The increasing number of events that are obtained from the LHC in the next years will help to tune the MC generators to an even better performance. In the future more and more next to leading order effects will be included in the programs as well. Therefore the difference between the programs should decrease in the future.

5 Acknowledgement

I want to thank everyone who supported me during the last three years and especially during the last month. Thank you, my heavenly father!

A special thanks goes to my supervisors Prof. Dr. W. Lohmann and Dr. D. Merkt who made this work possible, especially in the tight schedule.

I want to thank my family and all my friends in Zittau as well as in Cottbus, for supporting me in the last three years. I really appreciate that. I also want to thank the Roland-Berger-Foundation for giving me many great opportunities during the last two years. I also want to thank everyone from Cottbus University who made this possible, especially Prof. Dr. G. Seibold.

Many thanks goes to all the people at DESY-Zeuthen. I want to thank Ringo Schmidt, who was always there if I had a question (and there were so many of them). Thank you for reading my thesis and giving such good remarks, especially in the final hours!

An exceptional thanks goes to Igor Marfin, whos scripts made my work much easier or sometimes even possible. Огромное спасибо Игорь! Thank you very much Vladimir Makarenko, for generating all my wishes with MadGraph.

Many thanks goes to Jaimie Keppel who helped me fighting with "Physics English" and "English English".

Thank you very much M. W.

6 Declaration of Academic Honesty - Selbstständigkeitserklärung

I, Claudius Krause, declare that I did this work by myself only with the use of the below mentioned references.

Zeuthen, September 8, 2010

Ich, Claudius Krause, erkläre, dass ich die hier vorliegende Arbeit selbstständig und nur unter Nutzung der unten genannten Quellen angefertigt habe.

Zeuthen, September 8, 2010

References

- [Alwall08] J. Alwall et al. *Comparative study of various algorithms for the merging of parton showers and matrix elements in hadronic collisions*, Eur. Phys. J. C53 (2008), (arXiv:0706.2569)
- [Bigi86] I.I.Y. Bigi et al. *Production and Decay Properties of Ultra-heavy Quarks*, Phys. Lett. B181, 157, 1986
- [Bohm10] G. Bohm, G. Zech *Introduction to Statistics and Data Analysis for Physicists*, Verlag Deutsches Elektronen-Synchrotron, 2010
- [Boos01] E. Boos et. al., *Generic User Process Interface for Event Generators*, September 2001, (arXiv:hep-ph/0109068v1)
- [Cacc08] M. Cacciari, G. P. Salam, G. Soyez *The anti- k_t jet clustering algorithm*, April 2008, (arXiv:0802.1189v2)
- [Cata91] S. Catani, Y.L. Dokshitzer, M. Olsson, G. Turnock, B. R. Webber, *Phys. Lett. B* 269, 1991, p. 432
- [Cata92] S. Catani, Y.L. Dokshitzer, B. R. Webber, *Phys. Lett. B* 285, 1992, p. 291
- [Cata93] S. Catani, Y.L. Dokshitzer, B. R. Webber, *Nucl. Phys. B* 406, 1993, p. 187
- [Cheka02] S. V. Chekanov, *Jet algorithms: a minireview*, 2002, (arXiv:hep-ph/0211298v1)
- [CKKW] S. Catani, F. Krauss, R. Kuhn, B. R. Webber, *QCD Matrix Elements + Parton Showers*, (arXiv:hep-ph/0109231v1)
- [CMS1] found on 6th August,
<http://cms.web.cern.ch/cms/Detector/Designed/index.html>
- [CMS2] found on 6th August,
<http://www.weltmaschine.de/experimente/cms/>
- [CMS-CR] S.R. Slabospitsky *Event generators for top quark production and decays*, CMS Conference Report, CMS CR 2006/016, March 2006, (arXiv:hep-ph/0603124v1)
- [CMS-PAS] The CMS Collaboration, *CMS Physics Analysis Summary – Performance of Jet Algorithms in CMS*, CMS PAS JME-07-003, found on 15th September 2010 on
<http://cdsweb.cern.ch/record/1198227/files/JME-07-003-pas.pdf>
- [CMS-TDR] *CMS Physics - Technical Design Report*, found on 6th August,
<http://cdsweb.cern.ch/record/922757/files/lhcc-2006-001.pdf>
- [Denn07] , A. Denner, *Techniques for the calculation of electroweak radiative corrections at the one-loop level and results for W-physics at LEP200*, September 2007, (arXiv:0709.1075v1)
- [DESY] A. Floßdorf, B. Hegner, J. Mnich, C. Rosemann *CMS top physics at DESY*, internal presentation, CMS group meeting 12 July 2007
- [Brout64] F. Englert, R. Brout, *Phys. Rev. Lett.* 13, 1964, p.321
- [Erik07] David Eriksson, *CKKW and MLM matching*, presentation at a QCD course at Uppsala University, found on 9th August 2010 on
http://www.isv.uu.se/thep/courses/QCD/QCD_presentation_David.pdf
- [Garc10] A. García-Bellido, *Single top quark production at the Tevatron*, August 2010, (arXiv:1008.0169v1)
- [Glei03] T. Gleisberg et al. *SHERPA 1.α, a proof-of-concept version*, November 2003, (arXiv:hep-ph/0311263v1)
- [Glei08] T. Gleisberg et al. *Event generation with Sherpa 1.1*, JHEP 02 (2009) 007 (arXiv:0811.4622)

- [Higgs64] P. W. Higgs, *Phys. Rev. Lett.* **13**, 1964, p.508
- [Höche06] S. Höche et. al., *Matching Parton Showers and Matrix Elements*, February 2006, (arXiv:hep-ph/0602031v1)
- [Krau01] F. Krauss, R. Kuhn, G. Soff, *AMEGIC++ 1.0 A Matrix Element Generator In C++*, September 2001, (arXiv:hep-ph/0109036v1)
- [Krau02] F. Krauss, *Matrix Elements and Parton Showers in Hadronic Interactions*, 24 May 2002, (arXiv:hep-ph/0205283v1)
- [Kass] R. D. Kass, notes from lectures, found on 15th August 2010 on http://www.physics.ohio-state.edu/~kass/P880_P20_m&t06-1.ppt
- [Lohm09] W. Lohmann, *Script to the lecture "Allgemeine Physik III - Kerne und Teilchen"*, Brandenburg University of Technology, SS09
- [MadGraph] *MadEvent Minimal User Guide*, MadGraph version: V4.1, found on 5th August 2010, <http://cp3wks05.fynu.ucl.ac.be/Manual/Manual-March-2007.pdf>
- [Malt02] F. Maltoni, T. Stelzer, *MadEvent: Automatic event generation with MadGraph*, 15 August 2002, (arXiv:hep-ph/0208156v1)
- [MLM1] F. Caravaglios, M. L. Mangano, M. Moretti, R. Pittau, *A new approach to multi-jet calculations in hadron collisions*, Nucl. Phys. B539, p. 215, 1999, (arXiv:hep-ph/9807570v1)
- [MLM2] M. L. Mangano, M. Moretti, R. Pittau, *Multijet matrix elements and shower evolution in hadronic collisions: $Wb\bar{b} + n$ jets as a case study*, Nucl. Phys. B632, p. 343, 2002, (arXiv:hep-ph/0108069v1)
- [MCDB] Monte Carlo Database, <http://mcdb.cern.ch/>
- [PDG08] particle data group *PARTICLE PHYSICS BOOKLET*, July 2008
- [Povh03] B.Povh, K. Rith, C. Scholz, F. Zetsche *Particles and Nuclei*, Fourth Edition, 2003
- [PYTHIA] T. Sjöstrand, S. Mrenna, P. Skands, *PYTHIA 6.4 - Physics and Manual*, found on 16th August, <http://home.thep.lu.se/~torbjorn/pythia/lutp0613man2.pdf>
- [ROOT] R. Brun, F. Rademakers, *ROOT - An Object Oriented Data Analysis Framework*, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also <http://root.cern.ch/>
- [Salam07] G. P. Salam, G. Soyez, *A practical Seedless Infrared-Safe Cone jet algorithm*, April 2007, (arXiv:0704.0292v2)
- [SHERPA1] *SHERPA 1.1.1 manual*, found on 5th August 2010, <http://www.hepforge.org/archive/sherpa/howto-1.1.1.pdf>
- [SHERPA2] *SHERPA 1.2.0 internet manual*, found on 5th August 2010, http://projects.hepforge.org/sherpa/doc/SHERPA-MC-1.2.0/Sherpa_1.html
- [SHERPA3] *Homepage of the Sherpa Event Generator*, found on 6th August 2010, <http://projects.hepforge.org/sherpa/dokuwiki/doku.php>
- [Stelz94] T. Stelzer, W. F. Long, *Automatic Generation of Tree Level Helicity Amplitudes*, January 1994, (arXiv:hep-ph/9401258v1)
- [TOP-09-003] *Prospects for the first Measurement of the $t\bar{t}$ Cross Section in the Muon-plus-Jets Channel at $\sqrt{s} = 10$ TeV with the CMS Detector*, CMS Physics Analysis Summary, TOP-09-003, found on 17th August 2010, <http://cdsweb.cern.ch/record/1194519/files/TOP-09-003-pas.pdf>

- [1] found on 3rd August 2010, http://w3.iihe.ac.be/cms/data/Pictures/CMS_all_sketch.jpg
- [2] found on 5th August 2010,
http://cms.web.cern.ch/cms/Resources/Website/Media/Videos/Animations/files/CMS_Slice.gif
- [3] found on 6th August 2010,
http://projects.hepforge.org/sherpa/dokuwiki/_media/sketch.gif?cache=

The references to arXiv can be found at <http://arxiv.org/> and the corresponding article ID.

A Settings for SHERPA and MadGraph/MadEvent

A.1 SHERPA: Run.dat

```
(beam){
  BEAM_1          = 2212          ! possible beam particles:  P+, P-, e+, e-
  BEAM_ENERGY_1   = 3500.         ! in GeV
  BEAM_SPECTRUM_1 = Monochromatic ! Monochromatic,Laser_Backscattering,Simple_Compton
  K_PERP_MEAN_1   = 0.33
  K_PERP_SIGMA_1  = 0.8
  BEAM_2          = 2212          ! possible beam particles:  P+, P-, e+, e-
  BEAM_ENERGY_2   = 3500.         ! in GeV
  BEAM_SPECTRUM_2 = Monochromatic ! Monochromatic,Laser_Backscattering,Simple_Compton
  K_PERP_MEAN_2   = 0.33
  K_PERP_SIGMA_2  = 0.8
}(beam)
(isr){
  BUNCH_1         = 2212          ! possible beam particles:  P+, P-, e+, e-
  ISR_1           = 0n           ! On/Off
  BUNCH_2         = 2212          ! possible beam particles:  P+, P-, e+, e-
  ISR_2           = 0n           ! On/Off
  ISR_SMIN        = 1.e-10       ! Minimal fraction of nominal s for parton after ISR
  ISR_SMAX        = 1.0          ! Maximal fraction of nominal s for parton after ISR
  PDF_SET         = cteq6l1      !
  PDF_SET_VERSION = 1
  PDF_GRID_PATH   = CTEQ6Grid    ! CTEQ6Grid,PDFsets,MRST99Grid
}(isr)
(model){
  MODEL           = SM           ! Model
  EW_SCHEME       = 3            ! which parameters define the ew sector.
  M_W             = 80.419
  M_Z             = 91.188
  M_H             = 120
  GF              = 0.0000116639
  CKMORDER        = 1
  CABIBBO         = 0.2236
  MASS[6]         = 172.5        ! top mass, default: 175
  WIDTH[6]        = 1.45412849   !top width, default: 1.5
  WIDTH[23]       = 2.4412334    !Z width, default: 2.49
  WIDTH[24]       = 2.04759951   !W width, default: 2.06
  WIDTH[25]       = 0.00575165778 !H width, default: 0.0034
  MASSIVE[5]=1          ! consider massive bottom and top quarks
  MASSIVE[6]=1
}(model)
(me){
  ME_SIGNAL_GENERATOR = Amegic    ! Internal or Amegic or Comix(new)
  EVENT_GENERATION_MODE = Unweighted
}(me)
(integration){
  ERROR = 1.0e-2    ! Error by calculating matrix-elements
  FINISH_OPTIMIZATION = Off ! Integrate until optimization is completed
  SCALES = QCD
  COUPLINGS = QCD
  VEGAS = 0n
}(integration)
(mi){
```

```

    MI_HANDLER = Amisic                      ! Amisic / None
}(mi)
(shower){
SHOWER_GENERATOR = CSS
}(shower)
(fragmentation){
    FRAGMENTATION = Lund          ! Off, Lund (Pythia string fragmentation) or Ahadic
    DECAYMODEL     = Lund          ! Lund or Hadrons
}(fragmentation)
(processes){
!!ttbar with up to 2 jets in all channels
Process 93 93 -> 6[a] -6[b]
    Decay 6[a] -> 5 24[c]
    Decay -6[b] -> -5 -24[d]
    Decay 24[c] -> 90 91
    Decay -24[d] -> 93 93
CKKW sqr(30/E_CMS)
Integration_Error 0.02
    End process;
Process 93 93 -> 6[a] -6[b] 93
    Decay 6[a] -> 5 24[c]
    Decay -6[b] -> -5 -24[d]
    Decay 24[c] -> 90 91
    Decay -24[d] -> 93 93
CKKW sqr(30/E_CMS)
Integration_Error 0.02
    End process;
Process 93 93 -> 6[a] -6[b] 93 93
    Decay 6[a] -> 5 24[c]
    Decay -6[b] -> -5 -24[d]
    Decay 24[c] -> 90 91
    Decay -24[d] -> 93 93
CKKW sqr(30/E_CMS)
Integration_Error 0.05
    End process;

Process 93 93 -> 6[a] -6[b]
    Decay 6[a] -> 5 24[c]
    Decay -6[b] -> -5 -24[d]
    Decay 24[c] -> 93 93
    Decay -24[d] -> 90 91
CKKW sqr(30/E_CMS)
Integration_Error 0.02
    End process;
Process 93 93 -> 6[a] -6[b] 93
    Decay 6[a] -> 5 24[c]
    Decay -6[b] -> -5 -24[d]
    Decay 24[c] -> 93 93
    Decay -24[d] -> 90 91
CKKW sqr(30/E_CMS)
Integration_Error 0.02
    End process;
Process 93 93 -> 6[a] -6[b] 93 93
    Decay 6[a] -> 5 24[c]
    Decay -6[b] -> -5 -24[d]
    Decay 24[c] -> 93 93

```



```

    Decay -24[d] -> 90 91
CKKW sqr(30/E_CMS)
Integration_Error 0.05
End process;

Process 93 93 -> 6[a] -6[b]
    Decay 6[a] -> 5 24[c]
    Decay -6[b] -> -5 -24[d]
        Decay 24[c] -> 90 91
        Decay -24[d] -> 90 91
CKKW sqr(30/E_CMS)
Integration_Error 0.02
End process;
Process 93 93 -> 6[a] -6[b] 93
    Decay 6[a] -> 5 24[c]
    Decay -6[b] -> -5 -24[d]
        Decay 24[c] -> 90 91
        Decay -24[d] -> 90 91
CKKW sqr(30/E_CMS)
Integration_Error 0.02
End process;
Process 93 93 -> 6[a] -6[b] 93 93
    Decay 6[a] -> 5 24[c]
    Decay -6[b] -> -5 -24[d]
        Decay 24[c] -> 90 91
        Decay -24[d] -> 90 91
CKKW sqr(30/E_CMS)
Integration_Error 0.05
End process;

Process 93 93 -> 6[a] -6[b]
    Decay 6[a] -> 5 24[c]
    Decay -6[b] -> -5 -24[d]
        Decay 24[c] -> 93 93
        Decay -24[d] -> 93 93
CKKW sqr(30/E_CMS)
Integration_Error 0.02
End process;
Process 93 93 -> 6[a] -6[b] 93
    Decay 6[a] -> 5 24[c]
    Decay -6[b] -> -5 -24[d]
        Decay 24[c] -> 93 93
        Decay -24[d] -> 93 93
CKKW sqr(30/E_CMS)
Integration_Error 0.02
End process;
Process 93 93 -> 6[a] -6[b] 93 93
    Decay 6[a] -> 5 24[c]
    Decay -6[b] -> -5 -24[d]
        Decay 24[c] -> 93 93
        Decay -24[d] -> 93 93
CKKW sqr(30/E_CMS)
Integration_Error 0.05
End process;
}(processes)
(selector){

```

```

JetFinder sqr(20/E_CMS) 1.
PT 93 20. 100000.
PT 5 20. 100000.
PT 22 0. 100000.
PT 90 0. 100000.
PT 91 0. 100000.
Energy 93 0. 100000.
Energy 5 0. 100000.
Energy 22 0. 100000.
Energy 90 0. 100000.
Rapidity 93 -5. 5.
Rapidity 5 -5. 5.
Rapidity 22 -200000. 200000.
Rapidity 90 -200000. 200000.
DeltaR 93 93 0.001 100
DeltaR 5 5 0.001 100
DeltaR 90 90 0 100
DeltaR 22 22 0 100
DeltaR 93 5 0.001 100
DeltaR 93 22 0 100
DeltaR 93 90 0 100
DeltaR 22 5 0 100
DeltaR 90 5 0 100
DeltaR 22 90 0 100
Mass 93 93 0 100000
Mass 5 5 0 100000
Mass 22 22 0 100000
Mass 90 90 0 100000
}(selector)
(run){
RANDOM_SEED1 = 28893 ! will be changed by script
RANDOM_SEED2 = 1068 ! will be changed by script
EVENTS = 1000 ! can be changed in inputline
ANALYSIS = 0 ! not installed
OUTPUT =2 ! Information Events (if 3) and Errors (if 2)
CDXS_ITMAX 5000;
}(run)

```

A.2 MadGraph: Header of .lhe file

```

<LesHouchesEvents version="1.0">
<header>
<!--
*****
#                                                                 *
#                               MadGraph/MadEvent                  *
#                                                                 *
#                               The new web generation              *
#                                                                 *
#                               http://madgraph.hep.uiuc.edu        *
#                               http://madgraph.phys.ucl.ac.be      *
#                               http://madgraph.roma2.infn.it       *
#                                                                 *
#                               CP3 team                            Tim Stelzer *
#                               Universite de Louvain                University of Illinois *
#                                                                 *
#.....*
#                                                                 *
# This file contains all the information necessary to reproduce   *
# the events generated:                                           *
#                                                                 *
# 1. software version                                             *
# 2. proc_card.dat        : code generation info including model *
# 3. param_card.dat       : model primary parameters in the LH format *
# 4. run_card.dat         : running parameters (collider and cuts) *
# 5. pythia_card.dat      : present only in the pythia event file  *
# 6. pgs_card.dat        : present only in the pgs event file     *
#                                                                 *
#                                                                 *
*****
-->
<MGVersion>
# MG/ME version      : 4.4.44
# madgraph version   : 4.4.4
# template version   : 2.4.23
# helas version      : 3.8
# model version      : sm_1.2.1
</MGVersion>
<MGProcCard>
*****
#                               MadGraph/MadEvent                  *
#                               http://madgraph.hep.uiuc.edu        *
#                                                                 *
#                               proc_card.dat                       *
#                                                                 *
# This file is used to generate the code for a specific process. *
# Some notation/conventions:                                       *
#                                                                 *
# 0. Do not modify the TAGS and their order.                     *
# 1. hash/pound is a comment.                                     *
# 2. The number after the @ is used as an identifier for the class *
#    of processes. It can be any positive integer.               *
# 3. The number of lines for the max couplings depends on how many *
#    different classes of couplings are present in the model      *

```

```

# In the SM these are just two: QED (which include EW) and QCD *
# 4. Write "end_coup" after the couplings list, *
# to tell MG that the couplings input is over. *
# 5. Write "done" after the proc list to *
# to tell MG that the proc input is over. *
# 6. Some model names available at present are: *
# sm = Standard Model *
# smckm = Standard Model with Cabibbo matrix *
# mssm = Minimal Supersymmetric Standard Model *
# 2hdm = Generic Two Higgs Doublet model *
# heft = Higgs EFT (+Standard Model) *
# usrmod = User Model *
# 7. Don't leave spaces between the particles name in the *
# definition of the multiparticles. *
*****
*****
# Process(es) requested : mg2 input *
*****
# Begin PROCESS # This is TAG. Do not modify this line
pp>tt~ @0 # Process
QCD=2 # max qcd order
QED=0 # max qed order
end_coup # the coupling list is over
pp>tt~j @1 # Process
QCD=3 # max qcd order
QED=0 # max qed order
end_coup # the coupling list is over
pp>tt~jj @2 # Process
QCD=4 # max qcd order
QED=0 # max qed order
end_coup # the coupling list is over
done # the process list is over
# End PROCESS # This is TAG. Do not modify this line
*****
# Model information *
*****
# Begin MODEL # This is TAG. Do not modify this line
sm
# End MODEL # This is TAG. Do not modify this line
*****
# Start multiparticle definitions *
*****
# Begin MULTIPARTICLES # This is TAG. Do not modify this line
P uu~cc~dd~ss~g
J uu~cc~dd~ss~g
L+ e+mu+
L- e-mu-
vl vevm
vl~ ve~vm~
# End MULTIPARTICLES # This is TAG. Do not modify this line
</MGProcCard>
<slha>
*****
# MadGraph/MadEvent *
*****
# Les Houches friendly file for the SM parameters of MadGraph *

```

```

#           Spectrum and decay widths produced by SMCalc           *
#*****
#*Please note the following IMPORTANT issues:                        *
#                                                                    *
#0. REFRAIN from editing this file by hand! Some of the parame-   *
#   ters are not independent                                       *
#   (such as G_Fermi, alpha_em, sin(theta_W),MZ,MW) and serious   *
#   problems might be encountered (such as violation of unitarity *
#   or gauge invariance). Always use a calculator.                *
#                                                                    *
#1. alpha_S(MZ) has been used in the calculation of the parameters*
#   but, for consistency, it will be reset by madgraph to the     *
#   value expected IF the pdfs for collisions with hadrons are    *
#   used. This value is KEPT by madgraph when no pdf are used     *
#   lpp(i)=0 .                                                    *
#                                                                    *
#2. Values of the charm and bottom kinematic (pole) masses are   *
#   those used in the matrix elements and phase space UNLESS they *
#   are set to ZERO from the start in the model (particles.dat)   *
#   This happens, for example, when using 5-flavor QCD where      *
#   charm and bottom are treated as partons in the initial state  *
#   and a zero mass might be hardwired in the model definition.   *
#                                                                    *
#*****
Block SMINPUTS           # Standard Model inputs
    1      1.32506980E+02  # alpha_em(MZ)(-1) SM MSbar
    2      1.16639000E-05  # G_Fermi
    3      1.18000000E-01  # alpha_s(MZ) SM MSbar
    4      9.11880000E+01  # Z mass (as input parameter)
Block MGYUKAWA           # Yukawa masses m/v=y/sqrt(2)
#   PDG      YMASS
    5      4.20000000E+00  # mbottom for the Yukawa y_b
    4      1.42000000E+00  # mcharm for the Yukawa y_c
    6      1.64500000E+02  # mtop for the Yukawa y_t
   15      1.77700000E+00  # mtau for the Yukawa y_ta
Block MGCKM              # CKM elements for MadGraph
    1  1      9.75000000E-01  # Vud for Cabibbo matrix
Block MASS               # Mass spectrum (kinematic masses)
#   PDG      Mass
    5      4.80000000E+00  # bottom pole mass
    6      1.72500000E+02  # top pole mass
   15      1.77700000E+00  # tau mass
   23      9.11880000E+01  # Z mass
   24      8.04190000E+01  # W mass
   25      1.20000000E+02  # H mass
#   PDG      Width
DECAY      6      1.45412849E+00  # top width
DECAY     23      2.44123340E+00  # Z width
DECAY     24      2.04759951E+00  # W width
DECAY     25      5.75165778E-03  # H width
#   BR      NDA      ID1      ID2
    8.27656837E-02    2      4      -4  # BR( H -> c cbar )
    7.17702310E-01    2      5      -5  # BR( H -> b bbar )
    0.00000000E+00    2      6      -6  # BR( H -> t tbar )
    4.31827533E-02    2     15     -15  # BR( H -> tau- tau+)
    6.90768858E-03    2     23     23  # BR( H -> Z Z^(*))

```

```

7.46091936E-02      2      24      -24      # BR( H -> W      W^(*) )
3.02217699E-02      2      21      21      # BR( H -> g      g      )
1.42784807E-03      2      22      22      # BR( H -> A      A      )
</slha>
<MGRunCard>
*****
#                               MadGraph/MadEvent                               *
#                               http://madgraph.hep.uiuc.edu                       *
#                                                                           *
#                               run_card.dat                                     *
#                                                                           *
# This file is used to set the parameters of the run.                         *
#                                                                           *
# Some notation/conventions:                                                 *
#                                                                           *
# Lines starting with a '#' are info or comments                           *
#                                                                           *
# mind the format:  value      = variable      ! comment                     *
*****
#
*****
# Running parameters
*****
#
*****
# Tag name for the run (one word)                                           *
*****
'  tt'      = run_tag ! name of the run
*****
# Run to generate the grid pack                                             *
*****
.false.     = gridpack !True = setting up the grid pack
*****
# Number of events and rnd seed                                             *
*****
100000      = nevents ! Number of unweighted events requested
53 = iseed   ! rnd seed (0=assigned automatically=default))
*****
# Collider type and energy                                                 *
*****
1      = lpp1  ! beam 1 type (0=NO PDF)
1      = lpp2  ! beam 2 type (0=NO PDF)
3500   = ebeam1 ! beam 1 energy in GeV
3500   = ebeam2 ! beam 2 energy in GeV
*****
# Beam polarization from -100 (left-handed) to 100 (right-handed)         *
*****
0      = polbeam1 ! beam polarization for beam 1
0      = polbeam2 ! beam polarization for beam 2
*****
# PDF CHOICE: this automatically fixes also alpha_s and its evol.         *
*****
'cteq6l1' = pdlabel ! PDF set
*****
# Renormalization and factorization scales                                 *
*****

```

```

F          = fixed_ren_scale ! if .true. use fixed ren scale
F          = fixed_fac_scale ! if .true. use fixed fac scale
91.1880    = scale           ! fixed ren scale
91.1880    = dsqrt_q2fact1    ! fixed fact scale for pdf1
91.1880    = dsqrt_q2fact2    ! fixed fact scale for pdf2
1          = scalefact       ! scale factor for event-by-event scales
*****
# Matching - Warning! ickkw > 0 is still beta
*****
1          = ickkw           ! 0 no matching, 1 MLM, 2 CKKW matching
*****
#
*****
# BW cutoff (M+/-bwcutoff*Gamma)
*****
10000     = bwcutoff
*****
# Standard Cuts
*****
#
*****
# Minimum and maximum pt's                                     *
*****
20 = ptj          ! minimum pt for the jets
20 = ptb          ! minimum pt for the b
0  = pta          ! minimum pt for the photons
0  = ptl          ! minimum pt for the charged leptons
0  = misset       ! minimum missing Et (sum of neutrino's momenta)
0  = ptheavy      ! minimum pt for one heavy final state
1d5 = ptjmax      ! maximum pt for the jets
1d5 = ptbmax      ! maximum pt for the b
1d5 = ptamax      ! maximum pt for the photons
1d5 = ptlmax      ! maximum pt for the charged leptons
1d5 = missetMax   ! maximum missing Et (sum of neutrino's momenta)
*****
# Minimum and maximum E's (in the lab frame)                   *
*****
0  = ej          ! minimum E for the jets
0  = eb          ! minimum E for the b
0  = ea          ! minimum E for the photons
0  = el          ! minimum E for the charged leptons
1d5 = ejmax      ! maximum E for the jets
1d5 = ebmax      ! maximum E for the b
1d5 = eamax      ! maximum E for the photons
1d5 = elmax      ! maximum E for the charged leptons
*****
# Maximum and minimum rapidity                                 *
*****
5d0 = etaj       ! max rap for the jets
5d0 = etab       ! max rap for the b
2d5 = etaa       ! max rap for the photons
2d5 = etal       ! max rap for the charged leptons
0d0 = etajmin    ! min rap for the jets
0d0 = etabmin    ! min rap for the b
0d0 = etaamin    ! min rap for the photons
0d0 = etalmin    ! min rap for the charged leptons

```

```

*****
# Minimum and maximum DeltaR distance                                     *
*****
0.001 = drjj      ! min distance between jets
0.001  = drbb      ! min distance between b's
0      = drll      ! min distance between leptons
0      = draa      ! min distance between gammas
0.001  = drbj      ! min distance between b and jet
0      = draj      ! min distance between gamma and jet
0      = drjl      ! min distance between jet and lepton
0      = drab      ! min distance between gamma and b
0      = drbl      ! min distance between b and lepton
0      = dral      ! min distance between gamma and lepton
1d2 = drjjmax ! max distance between jets
1d2 = drbbmax ! max distance between b's
1d2 = drllmax ! max distance between leptons
1d2 = draamax ! max distance between gammas
1d2 = drbjmax ! max distance between b and jet
1d2 = drajmax ! max distance between gamma and jet
1d2 = drjlmax ! max distance between jet and lepton
1d2 = drabmax ! max distance between gamma and b
1d2 = drblmax ! max distance between b and lepton
1d2 = dralmax ! max distance between gamma and lepton
*****
# Minimum and maximum invariant mass for pairs                           *
*****
0      = mmjj      ! min invariant mass of a jet pair
0      = mmbb      ! min invariant mass of a b pair
0      = mmaa      ! min invariant mass of gamma gamma pair
0      = mml      ! min invariant mass of l+l- (same flavour) lepton pair
1d5 = mmjjmax ! max invariant mass of a jet pair
1d5 = mmbbmax ! max invariant mass of a b pair
1d5 = mmaamax ! max invariant mass of gamma gamma pair
1d5 = mmlmax ! max invariant mass of l+l- (same flavour) lepton pair
*****
# Minimum and maximum invariant mass for all letpons                     *
*****
0      = mmnl      ! min invariant mass for all letpons (l+- and vl)
1d5 = mmnlmax ! max invariant mass for all letpons (l+- and vl)
*****
# Inclusive cuts                                                         *
*****
0      = xptj ! minimum pt for at least one jet
0      = xptb ! minimum pt for at least one b
0      = xpta ! minimum pt for at least one photon
0      = xptl ! minimum pt for at least one charged lepton
*****
# Control the pt's of the jets sorted by pt                             *
*****
0      = ptj1min ! minimum pt for the leading jet in pt
0      = ptj2min ! minimum pt for the second jet in pt
0      = ptj3min ! minimum pt for the third jet in pt
0      = ptj4min ! minimum pt for the fourth jet in pt
1d5 = ptj1max ! maximum pt for the leading jet in pt
1d5 = ptj2max ! maximum pt for the second jet in pt
1d5 = ptj3max ! maximum pt for the third jet in pt

```



```

1d5 = ptj4max ! maximum pt for the fourth jet in pt
0   = cutuse  ! reject event if fails any (0) / all (1) jet pt cuts
*****
# Control the Ht(k)=Sum of k leading jets                                     *
*****
0   = htjmin ! minimum jet HT=Sum(jet pt)
1d5 = htjmax ! maximum jet HT=Sum(jet pt)
0   = ht2min ! minimum Ht for the two leading jets
0   = ht3min ! minimum Ht for the three leading jets
0   = ht4min ! minimum Ht for the four leading jets
1d5 = ht2max ! maximum Ht for the two leading jets
1d5 = ht3max ! maximum Ht for the three leading jets
1d5 = ht4max ! maximum Ht for the four leading jets
*****
# WBF cuts                                                                    *
*****
0   = xetamin ! minimum rapidity for two jets in the WBF case
0   = deltaeta ! minimum rapidity for two jets in the WBF case
*****
# maximal pdg code for quark to be considered as a jet                       *
# otherwise b cuts are applied                                                *
*****
5   = maxjetflavor
*****
# Jet measure cuts                                                            *
*****
20  = xqcut   ! minimum kt jet measure between partons
*****
</MGRunCard>
<MGGenerationInfo>
# Number of Events      :      100000
# Integrated weight (pb) : .27669E+03
# Truncated wgt (pb)    : .39469E+00
# Unit wgt              : .27669E-02
</MGGenerationInfo>
<MGDecayInfo>
# particle name      : t
# decay mode         : t -> b anything (e+mu+ta+ud+cs)
# MC partial width: .13935E+01
# Rnd seed           :      -1
# Number of Events   :      100000
# Integrated weight (pb): .27669E+03
# Max wgt            : *****
# Average wgt        : .27669E-02
</MGDecayInfo>
<MGDecayInfo>
# particle name      : t~
# decay mode         : t~ -> b~ anything (e+mu+ta+ud+cs)
# MC partial width: .13979E+01
# Rnd seed           :      -2
# Number of Events   :      100000
# Integrated weight (pb): .27669E+03
# Max wgt            : *****
# Average wgt        : .27669E-02
</MGDecayInfo>
</header>

```

```

<init>
  2212  2212  0.350000000000E+04  0.350000000000E+04  0 0 10042 10042 3 3
  0.10723674330E+03  0.11890260030E+00  0.27669000000E-02  0
  0.95759642100E+02  0.10617695111E+00  0.27669000000E-02  1
  0.73693614600E+02  0.81710448600E-01  0.27669000000E-02  2
</init>
<event>
...
</event>
</LesHouchesEvents>

```

A.3 Configuration file of FastSimulation

```

# Auto generated configuration file
# using:
# Revision: 1.155
# Source: /cvs_server/repositories/CMSSW/CMSSW/Configuration/PyReleaseValidation/python/ConfigBuilder
# with command line options: TTbar_Tauola_cfi.py -s GEN:ProductionFilterSequence,FASTSIM --pileup=No

import FWCore.ParameterSet.Config as cms

process = cms.Process('HLT')

# import of standard configurations
process.load('Configuration.StandardSequences.Services_cff')
process.load('FWCore.MessageService.MessageLogger_cfi')
process.load('FastSimulation.Configuration.RandomServiceInitialization_cff')
process.load('FastSimulation.PileUpProducer.PileUpSimulator7TeV_cfi')
process.load('FastSimulation.Configuration.FamosSequences_cff')
process.load('Configuration.StandardSequences.MagneticField_38T_cff')
process.load('Configuration.StandardSequences.Generator_cff')
process.load('FastSimulation.Configuration.FamosSequences_cff')
process.load('FastSimulation.Configuration.HLT_8E29_cff')
process.load('IOMC.EventVertexGenerators.VtxSmearedParameters_cfi')
process.load('FastSimulation.Configuration.CommonInputs_cff')
process.load('FastSimulation.Configuration.EventContent_cff')

process.MessageLogger.cerr.threshold = 'INFO'

#process.MessageLogger.categories.append('HepMCFileReader')
#process.MessageLogger.cerr.HepMCFileReader= cms.untracked.PSet(
#    limit = cms.untracked.int32(-1)
#)

#process.MessageLogger.categories.append('MCFileSource')
#process.MessageLogger.cerr.MCFileSource= cms.untracked.PSet(
#    limit = cms.untracked.int32(-1)
#)

process.VtxSmeared.initialSeed = cms.untracked.uint32(98765432)
process.famosPileUp.initialSeed = cms.untracked.uint32(918273)
process.famosSimHits.initialSeed = cms.untracked.uint32(13579)
process.siTrackerGaussianSmearingRecHits.initialSeed = cms.untracked.uint32(24680)
process.ecalRecHit.initialSeed = cms.untracked.uint32(654321)
process.ecalPreshowerRecHit.initialSeed = cms.untracked.uint32(6541321)

```

```

process.hbhereco.initialSeed = cms.untracked.uint32(541321)
process.horeco.initialSeed = cms.untracked.uint32(541321)
process.hfreco.initialSeed = cms.untracked.uint32(541321)
process.paramMuons.initialSeed = cms.untracked.uint32(54525)
process.MuonSimHits.initialSeed = cms.untracked.uint32(987346)
process.simMuonRPCDigis.initialSeed = cms.untracked.uint32(524964)
process.simMuonCSCDigis.initialSeed = cms.untracked.uint32(525432)
process.simMuonDTDigis.initialSeed = cms.untracked.uint32(67673876)

process.configurationMetadata = cms.untracked.PSet(
    version = cms.untracked.string('$Revision: 1.155 $'),
    annotation = cms.untracked.string('TTbar_Tauola_cfi.py nevts:10'),
    name = cms.untracked.string('PyReleaseValidation')
)
process.maxEvents = cms.untracked.PSet(
    input = cms.untracked.int32(-1)
)
process.options = cms.untracked.PSet(
    Rethrow = cms.untracked.vstring('OtherCMS',
        'StdException',
        'Unknown',
        'BadAlloc',
        'BadExceptionType',
        'ProductNotFound',
        'DictionaryNotFound',
        'InsertFailure',
        'Configuration',
        'LogicError',
        'UnimplementedFeature',
        'InvalidReference',
        'NullPointerError',
        'NoProductSpecified',
        'EventTimeout',
        'EventCorruption',
        'ScheduleExecutionFailure',
        'EventProcessorFailure',
        'FileInPathError',
        'FileOpenError',
        'FileReadError',
        'FatalRootError',
        'MismatchedInputFiles',
        'ProductDoesNotSupportViews',
        'ProductDoesNotSupportPtr',
        'NotFound'),
    wantSummary = cms.untracked.bool(True)
)
# Input source

process.source = cms.Source("LHESource",
    fileNames = cms.untracked.vstring(
#       'file:../../GeneratorInterface/Pythia6Interface/test/ttbar_5flavours_xqcut20_10TeV.lhe'

#insert files here

```

```

)
)

from Configuration.Generator.PythiaUESettings_cfi import *
#pythiaUESettingsBlock.pythiaUESettings+=[ 'MSTJ(21)=0    ! To prohibit all decays' ]

process.generator = cms.EDFilter("Pythia6HadronizerFilter",
    pythiaHepMCVerbosity = cms.untracked.bool(False),
    maxEventsToPrint = cms.untracked.int32(0),
    pythiaPylistVerbosity = cms.untracked.int32(0),
    filterEfficiency = cms.untracked.double(0.254),
    comEnergy = cms.double(7000.0),
    PythiaParameters = cms.PSet(
        pythiaUESettingsBlock,
        processParameters = cms.vstring('MSEL=0          ! User defined processes',
            'PMAS(5,1)=4.8    ! b quark mass',
            'PMAS(6,1)=172.5 ! t quark mass',
            'MSTJ(1)=1        ! Fragmentation/hadronization on or off',
            'MSTP(61)=1       ! Parton showering on or off'),
        # This is a vector of ParameterSet names to be read, in this order
        parameterSets = cms.vstring('pythiaUESettings',
            'processParameters')
    ),
    jetMatching = cms.untracked.PSet(
        scheme = cms.string("Madgraph"),
        mode = cms.string("auto"),      # soup, or "inclusive" / "exclusive"
        MEMAIN_etaclmax = cms.double(5.0),
        MEMAIN_qcut = cms.double(30.0),
        MEMAIN_minjets = cms.int32(0),
        MEMAIN_maxjets = cms.int32(2),
    )
)

# Output definition
process.output = cms.OutputModule("PoolOutputModule",
    splitLevel = cms.untracked.int32(0),
    outputCommands = process.FEVTDEBUGHLTEventContent.outputCommands,
    #   fileName = cms.untracked.string('My_cfi_py_GEN_FASTSIM.root')
    fileName = cms.untracked.string('INSERTFILEHERE'),
    dataset = cms.untracked.PSet(
        dataTier = cms.untracked.string('GEN-SIM-DIGI-RECO'),
        filterName = cms.untracked.string('')
    ),
    SelectEvents = cms.untracked.PSet(
        SelectEvents = cms.vstring('generation_step_semimuon')
    )
)

process.output.outputCommands += ['keep recoCaloJets_*_*_*']
process.output.outputCommands += ['keep recoGenJets_*_*_*']
process.output.outputCommands += ['keep *_mms2_*_*']
process.output.outputCommands += ['keep *_genParticles2_*_*']

```

```

process.output.outputCommands += ['keep *_wbosons_*_*']
process.output.outputCommands += ['keep *_tops_*_*']

# Output definition
process.output2 = cms.OutputModule("PoolOutputModule",
    splitLevel = cms.untracked.int32(0),
    outputCommands = process.FEVTDEBUGHLEventContent.outputCommands,
#    fileName = cms.untracked.string('My_cfi_py_GEN_FASTSIM.root')
    fileName = cms.untracked.string('CHANGEFILEHERE'),
    dataset = cms.untracked.PSet(
        dataTier = cms.untracked.string('GEN-SIM-DIGI-RECO'),
        filterName = cms.untracked.string('')
    ),
    SelectEvents = cms.untracked.PSet(
        SelectEvents = cms.vstring('!generation_step_semimuon')
    )
)
process.output2.outputCommands += ['keep recoCaloJets_*_*_*']
process.output2.outputCommands += ['keep recoGenJets_*_*_*']
process.output2.outputCommands += ['keep *_mms2_*_*']
process.output2.outputCommands += ['keep *_genParticles2_*_*_*']
process.output2.outputCommands += ['keep *_wbosons_*_*']
process.output2.outputCommands += ['keep *_tops_*_*']

# Additional output definition

# Other statements
process.famosPileUp.PileUpSimulator = process.PileUpSimulatorBlock.PileUpSimulator
process.famosPileUp.PileUpSimulator.averageNumber = 0
process.famosSimHits.SimulateCalorimetry = True
process.famosSimHits.SimulateTracking = True
#process.famosSimHits.ActivateDecays.comEnergy = 7000
process.simulation = cms.Sequence(process.simulationWithFamos)
process.HLTEndSequence = cms.Sequence(process.reconstructionWithFamos)

process.famosSimHits.SourceLabel= cms.InputTag("generator")
process.genParticles.src=cms.InputTag("generator")
process.genParticles.abortOnUnknownPDGCode=cms.untracked.bool(False)

# set correct vertex smearing
process.Early7TeVCollisionVtxSmearingParameters.type = cms.string("BetaFunc")
process.famosSimHits.VertexGenerator = process.Early7TeVCollisionVtxSmearingParameters
process.famosPileUp.VertexGenerator = process.Early7TeVCollisionVtxSmearingParameters
# Apply Tracker and Muon misalignment
process.famosSimHits.ApplyAlignment = True
process.misalignedTrackerGeometry.applyAlignment = True

process.misalignedDTGeometry.applyAlignment = True
process.misalignedCSCGeometry.applyAlignment = True

#process.GlobalTag.globaltag = "STARTUP31X_V8::All"

```

```

# Get frontier conditions - not applied in the HCAL, see below
from Configuration.PyReleaseValidation.autoCond import autoCond
process.GlobalTag.globaltag = autoCond['startup']

process.mycaloJetMet = cms.Sequence(
    process.recoAllJets+
    process.recoJetIds+
    process.metreco
)

## extend an existing sequence by otherLabels
#def mkseq(process, firstlabel, *otherlabels):
#    seq = getattr(process, firstlabel)
#    for x in otherlabels: seq += getattr(process, x)
#    return cms.Sequence(seq)

process.mySeq = getattr(process, 'reconstructionWithFamos')
process.mySeq.replace(process.caloJetMet, process.mycaloJetMet)
#process.reconstructionWithFamos += process.recoAllJets

process.ProductionFilterSequence = cms.Sequence(process.generator)

#####Semimuonic#####

process.genParticles2 = cms.EDProducer("GenParticleProducer",
    saveBarCodes = cms.untracked.bool(True),
    src = cms.InputTag("generator"),
    abortOnUnknownPDGCode = cms.untracked.bool(False)
)
#####Some Candidate selection

process.mms = cms.EDFilter("CandViewRefSelector",
    src = cms.InputTag("genParticles2"),
    cut = cms.string("abs(pdgId) = 13 & status = 1"),
    filter = cms.bool(True)
)

process.mms2 = cms.EDFilter("CandViewSelector",
    src = cms.InputTag("genParticles2"),
    cut = cms.string("abs(pdgId) = 13 & status = 1")
)

process.nns = cms.EDFilter("CandViewRefSelector",
    src = cms.InputTag("genParticles2"),
    cut = cms.string("abs(pdgId) = 14 & status = 1"),
    filter = cms.bool(True)
)

process.bbs = cms.EDFilter("CandViewRefSelector",
    src = cms.InputTag("genParticles2"),
    cut = cms.string("abs(pdgId) = 5 & status = 2"),

```

```

filter = cms.bool(True)
)
#####Some algos to reconstruct top-quraks and W-bosons
process.wbosonsPlus=cms.EDProducer('CandViewCombiner',
decay = cms.string("mms@+ nns"),
cut = cms.string("75 < mass < 90")
)
process.wbosonsMinus=cms.EDProducer('CandViewCombiner',
decay = cms.string("mms@- nns"),
cut = cms.string("75 < mass < 90")
)
process.wbosons=cms.EDProducer('CandViewMerger',
src = cms.VInputTag( "wbosonsPlus", "wbosonsMinus" )
)
process.wbosonsNumberFiltered=cms.EDFilter("CandViewCountFilter",
src = cms.InputTag("wbosons"),
minNumber=cms.uint32(1)
)
process.topsPlus=cms.EDProducer('CandViewShallowCloneCombiner',
decay = cms.string("bbs@- mms@+ nns"),
cut = cms.string("160 < mass < 180")
)
process.topsMinus=cms.EDProducer('CandViewShallowCloneCombiner',
decay = cms.string("bbs@+ mms@- nns"),
cut = cms.string("160 < mass < 180")
)
process.tops=cms.EDProducer('CandViewMerger',
src = cms.VInputTag( "topsPlus", "topsMinus" )
)
process.topsNumberFiltered=cms.EDFilter("CandViewCountFilter",
src = cms.InputTag("tops"),
minNumber=cms.uint32(1)
)
##### My sequence #####
process.mysemi = cms.Sequence(
process.genParticles2*
(process.mms2+process.mms)*
process.nns*
process.bbs*
(process.wbosonsPlus + process.wbosonsMinus)*
process.wbosons*
process.wbosonsNumberFiltered*
(process.topsPlus + process.topsMinus)*
process.tops*
process.topsNumberFiltered
)

#####End of semimuonic#####

# Path and EndPath definitions
process.generation_step_semimuon = cms.Path(cms.SequencePlaceholder("randomEngineStateProducer")+pro

```

```

process.reconstruction = cms.Path(process.reconstructionWithFamos)
process.out_step = cms.EndPath(process.output+process.output2)

# Schedule definition
process.schedule = cms.Schedule(process.generation_step_semimuon)

process.schedule.extend(process.HLTSchedule)
process.schedule.extend([process.reconstruction,process.out_step])
# special treatment in case of production filter sequence
for path in process.paths:
    getattr(process,path)._seq = process.ProductionFilterSequence*getattr(process,path)._seq

```