Validation of the Herwig++-Matchbox Framework for Event Generation at the CMS Experiment

Validierung des Herwig++-Matchbox Frameworks zur Eventgenerierung am CMS Experiment

Bachelor Thesis

Julia Hunt

at the Department of Physics
Institut für Experimentelle Kernphysik (IEKP)

Reviewer: Prof. Dr. Ulrich Husemann
Second reviewer: Dr. Christian Reuschle
Advisor: Marco Harrendorf

Karlsruhe, 14 October 2015
This thesis was typeset using \LaTeX, which was originally developed by Leslie Lamport and is based on \TeX created by Donald Knuth. Diagrams were drawn using the \LaTeX TikZ package [FT15]. Histograms were produced with Rivet [Riv15 B15].
I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 14 October 2015

(Julia Hunt)
Contents

1. Introduction 1

2. Theoretical and phenomenological principles 3
   2.1. Particle physics 3
      2.1.1. The Standard Model of particle physics 3
      2.1.2. Analysis principle 6
      2.1.3. Event generation – stages of a collision 7
      2.1.4. Hadronic cross sections in perturbation theory 8
   2.2. Monte Carlo event generation 10
      2.2.1. The Monte Carlo method 10
      2.2.2. Monte Carlo simulation 11
      2.2.3. Monte Carlo integration 11

3. Validation of the Herwig++-Matchbox framework 13
   3.1. Programs 13
      3.1.1. Herwig++ 13
      3.1.2. ThePEG 14
      3.1.3. MadGraph5_aMC@NLO 14
      3.1.4. OpenLoops 15
      3.1.5. Rivet 15
      3.1.6. Combinations of event generators 15
   3.2. Fixed order cross sections and results 17
      3.2.1. Comparison to Sherpa plus OpenLoops 24
      3.2.2. Fixed order Rivet analysis 25
   3.3. Particle level 26
      3.3.1. Construction of the Rivet ttH analysis for particle level samples 26
      3.3.2. Results of the Herwig++-Matchbox framework 30
      3.3.3. Comparison to MadGraph5_aMC@NLO and Herwig++-Matchbox 33
      3.3.4. Comparison to MadGraph5_aMC@NLO and Herwig++ 2.7.1 35
      3.3.5. Comparison to the LO matrix elements of Herwig++ 2.7.1 37

4. Summary 39

Bibliography 41

5. Appendix 47

Appendix 49
   A. Cross sections 49
   B. Input files for Herwig++ 52
   C. MadGraph5_aMC@NLO cards 59
   D. Rivet analyses 64
1. Introduction

The field of particle physics is concerned with gaining a better understanding of the fundamental laws of nature by studying the smallest components of our universe. Major tools on the experimental side are particle colliders in which particles are collided at ultrarelativistic kinetic energies. The collision products are studied in order to analyse the underlying fundamental interactions. For the intent of comparing a given physical model with measurement data of particle detectors, the means of choice are nowadays Monte Carlo simulations which strive to predict the outcome of an experiment in its entirety.

Since high energy collider experiments reach deep into the world of quantum mechanics where physics can only be described on probabilistic grounds, theoretical predictions are based on the simulation of large numbers of events. Simultaneously experiments must collect a tremendous amount of data.

The best-known and most powerful particle accelerator today is the Large Hadron Collider (LHC) near Geneva. In this ring accelerator, proton-proton and heavy ion collisions are studied at centre-of-mass energies of up to $\sqrt{s} = 14$ TeV. As protons are composite particles, one has to consider the theory describing the interactions of the corresponding constituent partons (quarks and gluons).

The commonly accepted theory to describe hadronic interactions at high energies is quantum chromodynamics (QCD), the quantum theory of the strong interaction. Under most circumstances, it is difficult to make theoretical predictions for hadronic interactions using QCD. Perturbative expansions in the coupling only apply at high energies, as the coupling strength between partons decreases for high momentum transfers. Until recently, it was sufficient for these approximate theoretical predictions to consider leading order (LO) terms only, since experimental uncertainties were prominent.

However, there have been tremendous efforts to reduce uncertainties in particle detectors like the ones situated at the LHC. Eventually a point has been reached where, in order to be able to draw further information from the comparison to theory predictions, it is necessary to reduce the uncertainties of the corresponding simulations as well. Therefore, the next-to-leading order (NLO) terms have to be considered.

1 The coupling strength decreases due to the negative running of the strong coupling constant, which leads to a phenomenon known as asymptotic freedom.
This thesis aims first and foremost at contributing to the validation of the new version of Herwig++, a computer program with the purpose of simulating such particle collisions. It provides exact next-to-leading order accuracy at the perturbative fixed order level for many processes. For this purpose, external next-to-leading order libraries are interfaced to the parton shower algorithms of Herwig++ to approximate all-order results and physical particle cascades further on.

The upcoming chapter 2 lays the theoretical and terminological foundations from a particle physics point of view. The Standard Model of elementary particle physics is briefly introduced and the concepts needed for particle physics event simulations and data analysis are explained.

In chapter 3, the necessary high energy physics software tools are introduced. Results of the new version of Herwig++, centred around the Matchbox framework, are presented and discussed for fixed order as well as particle level samples. Further, the results are compared with the output of several different combinations of other event generators.

The thesis at hand is written in the context of the recent kick-off of the second run of the LHC, where after an upgrade break of two years centre-of-mass energies of $\sqrt{s} = 13$ and 14 TeV are finally going to be reached.
2. Theoretical and phenomenological principles

2.1. Particle physics

The following chapter gives a brief overview of the particle physics background of the thesis. An outline to basics like the Standard Model is given, and some fundamental concepts of event generation like the analysis principle or QCD cross section handling are explained. En route, the commonly used terminology is introduced.

2.1.1. The Standard Model of particle physics

The current theory to describe the physics of fundamental particle interactions is called the Standard Model of particle physics. It describes a limited set of elementary particles and is based on three quantum field theories that describe the three fundamental interactions at nuclear scale and below. Gravity is not described by the Standard Model, since there is no satisfying quantum gravity theory to this day. However, assuming (with good reason) that effects of gravity are negligible at such small distances is a good approximation.

The three relativistic quantum field theories of the Standard Model describe three fundamental forces: Quantum electrodynamics (QED) \[\text{[Tom46, Sch48, Fey49]}\] to describe electromagnetic processes, the Glashow-Weinberg-Salam (GWS) theory \[\text{[BH82]}\], which treats the electromagnetic and weak interaction in a unified theory, and quantum chromodynamics (QCD) as the theory describing the strong interaction.

QED, GWS theory, and QCD are the theories that arise under local $SU(3) \times SU(2) \times U(1)$ invariance. These symmetries give rise to vector fields that are in turn associated with the three described fundamental forces. Quantizing these fields yields the mediators of the fields, the gauge bosons listed in table 2.1. Their coupling to other fundamental particles is described by the Standard Model Lagrangian, see e.g. \[\text{[CG07, Gr10]}\]. It can be split into three parts: a QCD sector, an electroweak sector, and a Higgs sector.

Fundamental particles are particles that are not composite. Speaking about the Standard Model, this is attributed to the gauge and Higgs bosons, quarks, and leptons – the latter two being fermionic, meaning they have half-integer spin. Figure 2.1 lists all elementary particles of the Standard Model and also gives an overview as to which particles take part in which interaction.
Table 2.1: Mediators of fundamental interactions. The Standard Model force carriers are listed along with their common abbreviation symbols, the corresponding fundamental force and charge, and their masses \cite{O+14a}.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Mediator</th>
<th>Force Acting on charge</th>
<th>Mediator mass</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma$</td>
<td>photon</td>
<td>electromagnetic</td>
<td>electric</td>
</tr>
<tr>
<td>$W^\pm, Z$</td>
<td>W and Z bosons</td>
<td>weak</td>
<td>weak isospin and hypercharge</td>
</tr>
<tr>
<td>$g$</td>
<td>gluon</td>
<td>strong</td>
<td>colour</td>
</tr>
</tbody>
</table>

Quarks exist in three families. In total, there are six types of quarks called flavours: up, down, strange, charm, bottom, and top. Leptons come in six types as well. All of the quarks and the electron, muon, and tau lepton carry electric charge. Therefore, they interact electromagnetically. The weak gauge bosons couple to all quarks and leptons as well as the Higgs boson. Meanwhile, only partons, i.e. quarks and gluons, carry colour charge, meaning they participate in the strong interaction.

There are three kinds of colour charges, commonly referred to as blue, green and red. As a consequence, gluons come in eight varieties. Furthermore, for each of the fermions there exists an antiparticle with oppositely signed charge-like quantum numbers.

All known matter is made up from these elementary particles. Quark bound states only occur in nature with a total colour charge of zero and are termed hadrons. Two sub-categories of hadrons are baryons and mesons. Our stable everyday surroundings feature mainly electrons and baryons which consist of three valence quarks. Another possibility to arrange quarks in a colourless fashion is a meson. This is the term for a quark-antiquark bound state.

Relevant for this thesis are especially the QCD properties of protons (which, comprising two up and one down quark, classify as baryons), top quarks, top antiquarks, and also the Higgs boson.

The Higgs boson is the quantum excitation of the Higgs field. It gives rise to the masses of the gauge bosons and is the newest discovered and also final piece in the Standard Model. Decades after its prediction by Robert Brout and François Englert \cite{EB64}, Peter Higgs \cite{Hig64}, and Gerald Guralnik, Carl Hagen and Tom Kibble \cite{GHK64}, its discovery \cite{CMS12, ATL12} was only achieved in July 2012 by the LHC collaborations ATLAS and CMS \cite{CMS08}.

The theoretical artifice leading to the prediction of the Higgs boson also provides a necessary mechanism to explain the masses of the gauge bosons and fermions; adding a straightforward mass term to the Lagrangian density of a theory with local gauge invariance like the Standard Model would break this symmetry and is, therefore, not possible; see e.g. \cite{Wol15}.

However, the concept of spontaneous symmetry breaking induced by the Higgs mechanism, an ingredient to the Standard Model, can serve as an explanation which is consistent with experimental evidence of massive W and Z bosons, and also massive fermions. In spontaneous symmetry breaking the Lagrangian density and, therefore, the derived equations of motion keep the local gauge symmetry. However, the system possesses a non-zero degenerate vacuum expectation value in its ground state and thus, by choosing a particular one of the previously equivalent ground states, the symmetry is spontaneously broken \cite{Wol15}. 

A frequently called upon example to illustrate this is the mexican hat potential, in which the potential itself obeys rotational symmetry, as does for example a marble on top of it. Nevertheless, if the marble falls into the brim in a specific direction, the symmetry is broken; see e.g. [Gri08].

This spontaneous breaking of local symmetries gives rise to the prediction of an additional scalar field, to which massive gauge bosons and fermions couple. Through this coupling the acquisition of mass can be explained. In the case of the fermions this coupling takes the form of a Yukawa coupling. This scalar field is called the Higgs field; it also features triple and quartic Higgs boson self-couplings.

The Standard Model has survived countless experimental tests and performs very well at predicting the outcome of such experiments. Therefore, it is viewed as an established theory. Everything yet to come will – but for a very profound uprooting of our physical understanding – have to include the Standard Model as a limiting case. Still, the Standard Model has its restrictions; for example a missing dark matter candidate or the inability to explain matter-antimatter asymmetry. These observations are not explained within the Standard Model, and many believe that physics beyond the Standard Model (BSM) must exist.

Figure 2.1.: Elementary Particles of the Standard Model and their interactions. Red-shaded particles participate in the strong interaction, grey-shaded particles in the electromagnetic interaction, and green-shaded particles in the weak interaction. Taken from [Har14, Mis14].
2.1.2. Analysis principle

A new physical model, which should be compatible with the so-far established theories, can be tested by comparison to experimental data. In the case of collider physics, the underlying procedure of the necessary analysis chain can be sectioned into the steps shown in figure 2.2.

![Figure 2.2: Illustration of the steps in a particle physics analysis. Based on [Har14] and [PG13].](image-url)

- **Acquisition of data**
  First of all an experiment has to be designed and built that is suited to validate or falsify the models in question. Ensuingly, to obtain significant statistical data samples, the same experimental setup has to be gone through again and again. In the case of the LHC, a design luminosity of \(10^{34} \text{ cm}^{-2}\text{s}^{-1}\) and an inelastic scattering cross section of 100 mb yield approximately \(10^9\) inelastic interactions per second. This corresponds to a tremendous amount of data, and it proves difficult to store all of it in such quick succession [Sta07].

- **Selection**
  With regard to reducing the storage requirement, several layers of automated triggers and filters are connected ahead. They are designed to extract events which involve physics of current research interest. Only about 200 events per second pass these filters, and are stored on the Worldwide LHC Computing Grid (WLCG) [LHC]. This will still correspond to a data amount of roughly 30 petabytes per year during LHC Run 2 [Cha]. In the end, after further selection only 1 in approximately \(10^{12}\) events is kept [Lan13].
The information about an event is limited to specific types of particles that can be observed by the detector. Those are not necessarily the original products of the underlying hard scattering, but mostly objects after a long chain of decays. Lastly, even though major efforts are made to hermetically instrument all directions with detecting devices, it cannot be avoided that information is lost owing to detector geometry and finite resolution.

- **Theory input and event simulation**
  In order to obtain simulated data from a given theoretical model, event generators are employed. Such programs, like Herwig++, are capable of simulating specified physics processes and their subsequent decays starting from the matrix elements \( \text{ME} \). While obtaining the exact matrix elements for a specified process from the theory’s Lagrangian is the basis for the simulation, there is still a long way to go to actual simulated data on particle level. A more detailed description of event generation can be found in the following section 2.1.3. Obtaining simulated data is likewise a question of statistics and consequently many events are required for a reliable statement.

- **Detector simulation**
  All simulated data on what is called the hadron or particle level must undergo a detector simulation in order to receive information from theory simulations on the same level as accessible from the experimental side. This includes simulating the detector response, reconstructing the simulated and observed particles from plain detector level data, and applying the same trigger and selection cuts.

- **Comparison**
  Eventually the two data sets can be compared. If they agree, this encourages confidence in the theory. If a significant difference occurs and other causes can be excluded, this might be a sign of new physics.

### 2.1.3. Event generation – stages of a collision

Hadrons consist of quarks and gluons, whose interaction is described by QCD. The principal quantum numbers are thereby determined through the valence quarks.

It lies within the nature of QCD that the coupling of quarks and gluons is strong in the low energy regime. One consequence is colour confinement; no object carrying colour charge can ever be observed. On the other hand, the coupling decreases as the energy increases. In high energy physics experiments the coupling gets so weak that it is appropriate to describe the partons as quasi-free particles. This behaviour is called asymptotic freedom and allows theoretical physicists to apply perturbation theory.

**Hard subprocess**

The central quantity to look for in collider scattering experiments are cross sections \( \sigma \). They are a measure of the probability that an interaction of two particles takes place. Determining cross sections for hadronic processes is not straightforward, and will be addressed in the next section 2.1.4. One element however, the hard subprocess, can be described using perturbation theory.

This stage of the event generation yields data on a level called fixed order. In this thesis, the term “parton level” is applied synonymously for fixed order, although strictly speaking simulated data after applying the parton shower step still describes partons.

---

1 In this context, matrix element always refers to the scattering matrix element of the underlying hard scattering, which is the projection of the time-evolved initial state of a partonic scattering process on the final state.
Parton shower
Often the sets of incoming and outgoing particles include coloured partons, which in turn will radiate other partons (gluons or quark-antiquark pairs) with some probability. This eventually leads to a cascade of partons, called parton shower. QCD radiation is enhanced in the soft and collinear regimes and thus, partons are often observed in collimated bunches called jets. At each branching, the energy per parton decreases until it is of the order of the hadronisation scale, which is roughly 1 GeV \[^{+11b}\]. At this energy the coupling gets too strong, and the partons begin to form hadrons.

If the parton shower step is applied, the phase space division between the hard subprocess and the parton shower must be considered properly to avoid double counting. This is called matching.

Hadronisation
From this point on, another – well-tested, but barely understood – process takes place, called hadronisation. It is so far only described by phenomenological models. Without hadronisation, the previous step would leave us with coloured particles which are known not to exist freely and thus cannot be observed in nature. The idea is to rearrange the partons in packages of colour neutral baryons and mesons that can finally be observed by a real particle detector. Naturally many of them are still unstable particles and are most likely to decay before leaving the detector, see e.g. \[^{+11b}\].

Underlying event
If referring to inelastic hadron-hadron collisions, there are also some spectator quarks or remnants to be considered which do not necessarily take part in the hard subprocess. Nonetheless, they influence the process as they undergo further, mostly soft, scattering and also interact with partons stemming from the hard subprocess. This is designated as the underlying event.

If several partonic interactions take place in the same hadron-hadron collision, this is called multiple parton interaction (MPI).

All of the above has to be considered in a simulation for hadron scattering and every event generator basically follows these steps as outlined.

2.1.4. Hadronic cross sections in perturbation theory
Hadronic cross sections contain contributions from long-distance (soft regime) and short-distance (hard regime) physics. Only the short-distance physics can be described in perturbation theory, whereas long-distance physics cannot. Coping with this can be achieved by factorising the original cross section into a part for the long-distance physics, which absorbs all problematic soft and collinear terms\(^2\) and a hard-scattering or partonic cross section for the subprocess, which can be computed in perturbation theory \[^{CSS04,M+07}\].

For the long-distance physics part the distribution of partons inside hadrons (parton distribution function or parton density function, PDF) have to be extracted from experimental measurements like for example at the H1 and ZEUS experiment at HERA or D0/CDF at the Tevatron, or at the LHC \[^{PG13}\]. Several PDF sets exist, for instance the CT \[^{CT1}\] or CTEQ \[^{The}\] sets, or the NNPDF sets \[^{b}\].

Once the distribution of partons has been measured at a certain energy, the PDF can be applied universally, as the scale evolution of PDFs can be shown to be governed by a set of coupled integral-differential equations called the DGLAP equations \[^{GL72,Dok77,AP77}\].

\(^2\)Particles radiated from other particles are considered soft if they possess a small energy fraction of the initial particle. Collinear means that the angle between the initial particle’s axis and the particle’s own path is vanishing.
The hadronic cross section $\sigma_X$ for $X$ final state particles can be written as the convolution of the PDFs $f_{a/b}(x_1/2, \mu_F^2)$ and the partonic cross section $\sigma_{ab \rightarrow X}(x_1, x_2, \alpha_s(\mu_R^2), \frac{Q^2}{\mu_F^2}, \frac{Q^2}{\mu_R^2})$ [Sch10]:

$$\sigma_X = \sum_{a,b} \int_0^1 \! dx_1 dx_2 \, f_a(x_1, \mu_F^2) \, f_b(x_2, \mu_F^2) \, \sigma_{ab \rightarrow X}(x_1, x_2, \alpha_s(\mu_R^2), \frac{Q^2}{\mu_F^2}, \frac{Q^2}{\mu_R^2}) + O\left(\frac{1}{\mu_F^2}\right).$$

$f_a(x_1, \mu_F^2)$ denotes the parton density function for finding parton $a$ with longitudinal momentum fraction $x_1$ in hadron 1. The Bjorken variable $x_1$ also gives the energy or distance at which the hadron is probed. The factorisation scale $\mu_F$ corresponds to the resolution scale separating the long-distance physics and short-distance physics [M+07, McE07]. Furthermore, the partonic cross section $\sigma_{ab \rightarrow X}$ is proportional to the squared scattering amplitude $M$:

$$\sigma_{ab \rightarrow X}(x_1, x_2, \alpha_s(\mu_R^2), \frac{Q^2}{\mu_F^2}, \frac{Q^2}{\mu_R^2}) \equiv \int \left| M_{ab \rightarrow X}(x_1, x_2, \alpha_s(\mu_R^2), \frac{Q^2}{\mu_F^2}, \frac{Q^2}{\mu_R^2}) \right|^2 d\Phi_X,$$

where the Lorentz-invariant measure $Q^2$ denotes the momentum transfer in the process, and $\alpha_s$ denotes the strong coupling constant which is in fact not a constant in this picture but a running coupling depending on the renormalisation scale $\mu_R$. The renormalisation scale is usually chosen equal to $\mu_F$. Finally, $d\Phi_X$ represents the phase space element for $X$ final state particles.

It is advantageous to expand the partonic cross-section in the strong coupling constant $\alpha_s$ such that

$$\sigma \approx \sigma^{LO} + \sigma^{NLO} + \text{higher order terms.}$$

The LO part contains those diagrams that give rise to the desired process at the lowest possible order in $\alpha_s$. This is also referred to as the Born level. The NLO part contains terms for the real emission $\int_{n+1} d\sigma^R$, where an additional parton is radiated in the hard subprocess (also called leg), and the virtual contribution $\int_n d\sigma^V$, one-loop Feynman diagrams. Taking these into account, the NLO cross section for an $n$-particle final state is given by [B+10]

$$\sigma^{NLO} = \sigma^{LO} + \int_{n+1} d\sigma^R + \int_n d\sigma^V + \int_n d\sigma^C,$$

$$\sigma^{LO} = \int_n d\sigma^B.$$

While ultraviolet divergences can be dealt with through the application of a renormalisation scheme, the infrared divergences need different handling; owing to the circumstance that infrared divergences appear separately in the virtual and real emission term, they cannot simply be renormalised. A common way to deal with the divergences in numerical programs is known as the subtraction method.

---

3Originally, the virtual contributions contain ultra-violet divergences. To get rid of them, a renormalisation procedure must be applied. Afterwards, $\alpha_s$ is dependent on the scale that defines the kinematics of the renormalisation, see e.g. [Ran14].
2. Theoretical and phenomenological principles

The collinear counterterm $\int d^2 g \sigma^C$ ensures analytical cancellation of additional collinear divergences in the initial state radiation, which are not taken care of by the subtraction method, see e.g. [Rau14].

Figure 2.3 shows exemplary LO diagrams of the $t\bar{t}H$ process.

Figure 2.3.: The leftmost and central Feynman diagrams represent gluon fusion processes leading to the final state $t\bar{t}H$. The right-hand Feynman diagram illustrates quark-antiquark annihilation with the same final state. Adapted from [Dit11].

LO event generation is often (but not always) enough to determine the shape of distributions of observables, whereas it does not give the normalization correctly. This lies in the nature of the factorisation scale $\mu_F$ which in itself is an unphysical quantity, as the full cross section does not depend on it. However, considering that only finite orders can be computed in perturbative QCD, the results do depend on $\mu_F$. Therefore, large corrections are usually to be expected for LO. The empirical correction that must be applied in order to scale LO to NLO is usually referred to as NLO K-factor $K$:

$$K = \frac{\sigma^{NLO}}{\sigma^{LO}}.$$  

2.2. Monte Carlo event generation

This chapter introduces the concept of the numerical Monte Carlo method and highlights the uses and applications of Monte Carlo integration and simulation in high energy particle physics, especially in the process of event generation.

2.2.1. The Monte Carlo method

The Monte Carlo method is a central tool in high energy particle physics analysis. In the course of comparing between theory and experiment various situations arise in which it is necessary to resort to this numerical technique.

In the context of the Monte Carlo method, probabilities are approximated using random numbers. For the sake of feasibility pseudo-random numbers from a deterministic sequence must suffice. Today there are sophisticated procedures to produce pseudo-random numbers with minimized correlation. Commonly used algorithms are the Mersenne-Twister [MN97] or the WELL “Well Equidistributed Long-period Linear” generators [PLM06].

A good statement about the quantity that is to be approximated with the Monte Carlo method can be made thanks to the (Strong) Law of Large Numbers. It states that for a sequence of independent and identically distributed random numbers, which follow some probability distribution, the mean of this sample will almost surely converge to the expectation value of the probability distribution. The discussion in this section follows [Gra13, B+11b].

In data analysis the Monte Carlo method is employed in Monte Carlo simulation and Monte Carlo integration. With an increasing number of dimensions to integrate over, analytical methods eventually get too complex and one has to use such stochastic numerical methods.
2.2.2. Monte Carlo simulation

Whenever a system is very intricate, for instance involving many particles or very complicated analytical descriptions, Monte Carlo simulation is useful. With Monte Carlo simulation it is also possible to estimate parameters that are not easily determined experimentally if a system description can be modelled. Prior to analysing experimental or simulated data, it is a handy tool for the simulation of expected background behaviour and for choosing eligible variables or regions in phase space where background and data can be distinguished with less effort. Monte Carlo simulation is not limited to background estimates – it is in fact well-adapted to simulating whole data sets of expected experimental results.

Moreover, Monte Carlo simulation has its uses for experimental physicists: Underlying physics of an experiment may be simulated in advance and a well-adjusted detector design can be developed accordingly. Eventually, simulation of the detector’s inaccuracies and its acceptance can help with the reconstruction of the actual events, see e.g. [Rau14, Sch10].

2.2.3. Monte Carlo integration

Monte Carlo integration relies on the fact that a $d$-dimensional integral $I$ over a function $g(\vec{x})$ can be interpreted in terms of the average of the function $\langle g \rangle$. Hence the result can be numerically approximated by a random number sample of size $N$ in the following way [San13]:

$$I := \int_D d^d x g(\vec{x}) = V \cdot \langle g \rangle,$$

$$\bar{g} = \frac{V}{N} \sum_{i=1}^{N} g(\vec{x}_i) \rightarrow V \cdot \langle g \rangle \text{ for } N \rightarrow \infty,$$

where $D$ is a domain in the $d$-dimensional integration region and $V$ the volume of $D$, see e.g. [Rau14, Gra13]. The integration points $\vec{x}_i$ can be composed of random numbers. In contrast to other numerical integration methods like Gaussian integration or Simpson’s rule, whose complexity or accuracy gets worse for higher dimensions, the accuracy of a Monte Carlo integration does not depend on the number of dimensions. It increases with an increased sample size, as the variance scales with $1/N$.

Consequently, Monte Carlo integration is the only choice for calculating the phase space integrals occurring in the cross section computation of event generation. There, the dimension of the integral is $d = 3n - 4$ for an $n$-particle final state.

The problem of the above straight Monte Carlo integration is that it is very inefficient in terms of convergence and also inaccurate if the distribution involves large fluctuations. The fluctuations will be rarely hit by random numbers, but will influence the mean value significantly. In order to overcome this problem, a variance reduction technique called importance sampling can be employed. A transformation using a weight function $w(\vec{x})$ is applied to increase the sampling in important areas. $w(\vec{x})$ is simply the Jacobian determinant $\left| \frac{\partial(\vec{y})}{\partial(\vec{x})} \right|$ of the variable transformation with the underlying mapping $\vec{x} \rightarrow \vec{y}(\vec{x})$:

$$I = \int_D d^d x g(\vec{x}) = \int_{y(D)} d^d y \frac{g(\vec{y}(\vec{x}))}{w(\vec{y}(\vec{x}))}.$$
3. Validation of the Herwig++-Matchbox framework

The Matchbox framework of Herwig++ offers interfaces to several other matrix element providers or event generators. The chapter at hand gives a short introduction to the programs that were used in the course of this bachelor thesis.

3.1. Programs

Herwig++ is a computer program for event generation.

Several different approaches to the setup of an event generator exist: there are matrix element generators that only provide matrix elements (ME) of the hard subprocess at LO or NLO. Others perform the phase space integration and parton shower (PS) matching as well; they are referred to as LO+PS or NLO+PS, respectively.

General-purpose Monte Carlo (GPMC) event generators (often abbreviated to event generators) on the other hand simulate all necessary steps from the hard subprocess up to hadronisation, see e.g. [Har14] for further details. Some GPMC event generators use their own ME generator, or otherwise have a few ME hard coded. More often, external ME providers are interfaced nowadays.

3.1.1. Herwig++

Herwig++ is a general purpose Monte Carlo event generator. The project was initiated in the prospect of the LHC and is the successor of the now obsolete event generator Herwig. For the benefit of a more flexible structure and newer possibilities in coding, it was decided to start from scratch again, using the experience won from Herwig. Herwig++ is written in C++ programming language instead of Fortran and it comprises many improvements in the physics implementations.

The current release version Herwig++ 2.7.1 features, amongst others, the following elements:

- Several hard coded LO and NLO MEs
- Angular ordered parton shower
- Dipole shower [GP09] [GP11]
• Many hard coded NLO MEs include the corrections for NLO matched cross sections
• For some processes, the parton showers are matrix element corrected
• Decays (including spin correlations)
• Cluster hadronisation
• Multiple parton interactions (MPI)
• Various Beyond the Standard Model (BSM) hard subprocesses (hard coded or through specific model file input)

For a more detailed description, the reader is referred to the manual of the version 2.3 [B+08] and the release notes that have been added since then [Her14]. This version has been thoroughly validated during the past years by a broad community. It also found its way into the CMSSW software framework used by the CMS collaboration [CMS15].

The follow-up Herwig++-Matchbox version is going to contain several new features. At the time of writing, the new version is not yet officially released and only available to the developers. An integral part of the new version is the Matchbox framework; NLO matching now happens fully automated and parton shower algorithms have been improved in this context as well. The Matchbox framework also provides an environment to make use of different matrix element providers like MadGraph5_aMC@NLO (section 3.1.3), OpenLoops (section 3.1.4), GoSam [C+11, C+14], VBFNLO or NJet [B+13, B+11a, EZ08, vO91, B+a] (via the BLHA interface standard of Herwig++ or other forms of runtime interfaces). The possibility to separately generate the integration grids (Monte Carlo integration, see section 3.2.3) of single hard subprocesses and thus parallelise this step is also implemented. Furthermore, a drastic improvement of the efficiency of the numerical integration and phase space sampling has been achieved.

A useful tool to run Herwig++-Matchbox parallelised on a computer cluster is Herwig-Parallel [Rau14]. It is currently only compatible with the slightly older Rivet [Riv15] version 1.9.0; a short introduction of Rivet is given in section 3.1.5. As the Herwig-Parallel extension was used to create all fixed order and some of the particle level samples here, said Rivet version was used throughout the thesis for consistency reasons.

3.1.2. ThePEG

Herwig++ is built on ThePEG, the Toolkit for High Energy Physics Event Generation [L+15]. ThePEG provides all necessary infrastructure for the event generation procedure and is constructed to allow users to plug in physics implementations. The benefit of this is that it is not necessary to recompile the program every time the physics implementations change, since ThePEG overrides its defaults at run time with the plugin’s parameters.

ThePEG also provides access to the commonly used PDF sets of the LHAPDF interpolator [LHA15, B+14]; it can handle Les Houche Accord event format (LHE) files, a standardised interface to facilitate the interaction of matrix element providers and event generators; it includes a HepMC [D+10] output module; and it offers an interface to Rivet and the jet algorithm library FastJet [MCS12].

3.1.3. MadGraph5_aMC@NLO

MadGraph5_aMC@NLO [A+14] is the result of a junction of MadGraph5 and aMC@NLO. While the original MadGraph5 event generator – being a ME+PS event generator – could

\footnote{Particle physicists have agreed on the usage of a file format called HepMC to simplify the exchange of particle level data computed by different event generators.}
only deal with processes at LO, the NLO+PS event generator aMC@NLO was specifically designed to match NLO matrix elements with a parton shower by using the MC@NLO matching in an automated way. The combination of the two projects yields an event generator within whose framework the user can simulate many Standard Model and BSM processes at LO and NLO with optional matching to the parton shower of a GPMC event generator [Mad15]. MadGraph5-aMC@NLO can also be interfaced directly to other generators as a provider for tree level and one-loop matrix elements, as e.g. implemented in the Herwig++-Matchbox framework.

3.1.4. OpenLoops

OpenLoops [CMP11, OPP07, vH10] is based on a high-performance algorithm that can calculate tree level and one-loop matrix elements for any SM process. Most Standard Model QCD processes are currently implemented in OpenLoops 1.1.1 at NLO precision.

OpenLoops 1.1.1 can be used as a one-loop matrix element provider for Herwig++ in the scope of the Matchbox framework or similarly as a plugin to Sherpa [G+09], as in section 3.2.1.

3.1.5. Rivet

Rivet [Riv15] stands for Robust Independent Validation of Experiment and Theory. It is a tool for validating an event generator, for the reason that it does not allow to pass information except for strictly physical observables after hadronisation – essentially operating on the same level of information as can be gained from experiment. The program is written in C++. Its analyses are based on constructs called projections, which allow for the filtering of user-defined final states from huge data sets. Rivet comes with a collection of analyses that can be applied to experimental and Monte Carlo data alike, as they operate on HepMC input files; therefore, it is a very useful interface between theory and experiment. Experimental collaborations sometimes provide a Rivet analysis along with a publication which can be used by event generator developers to double-check their own results through comparison. These analysis routines also enable theorists to study e.g. new (BSM) models using measurement data.

Rivet further offers the infrastructure to add more user-defined analyses, as will be further elaborated in chapters 3.2.2 and 3.3.1.

3.1.6. Combinations of event generators

In order to make a contribution to the validation of the Herwig++-Matchbox framework, the results of several combinations of event generators were compared. The centre piece is the first combination in which the new Herwig++ version was directly steering the whole event generation procedure.

All four combinations yielded output files in the HepMC format which were further analysed with Rivet. For the parton level studies, only the first combination was used. Particle level studies were conducted with all four combinations.

Combination 1

The setup illustrated in figure 3.1 was used: The Matchbox framework of Herwig++ accessed the tree-level matrix elements via its MadGraph5-aMC@NLO interface and took the virtual contributions from OpenLoops.

As matching procedure, the aMC@NLO procedure was used; for the parton shower the default option was chosen, which is the angular ordered parton shower.
3. Validation of the Herwig++-Matchbox framework

Figure 3.1: The Herwig++-Matchbox framework steered the event generation in the first combination.

Figure 3.2: The second and third combination used for event generation in this thesis were based on MadGraph5_aMC@NLO. MadGraph5_aMC@NLO is used to calculate everything up to parton level, then the Herwig++ 2.7.1 interface is used to have the further steps in the event generation carried out. In order to pass the same parton level sample to the Herwig++-Matchbox version, the input file generated by MadGraph5_aMC@NLO had to be slightly adjusted.

With this combination the parton level studies were conducted in a preliminary step, and eventually particle level events were generated in order to compare those to the combinations below.

Combination 2 and 3

What will be addressed as combinations 2 and 3 in the following are two variations of the same composition, the only difference between the two being the versions of Herwig++.

In this setup, sketched in figure 3.2, MadGraph5_aMC@NLO took the lead. Born, real, and virtual matrix element contributions were all taken from its own resources. MadGraph5_aMC@NLO has an implemented option to hand over to parton shower and hadronisation procedures of Herwig++ 2.7.1, which involves writing an LHE file (with event data at parton level) and tailoring an input file for Herwig++ 2.7.1. At this stage, the ME information given by MadGraph5_aMC@NLO is already on the level of a matched cross section. The input file was subsequently invoked by Herwig++ 2.7.1 to perform the parton shower. This setup is going to be referred to as Combination 3.

Since the LHE file and the input file were easily accessible afterwards, it was possible to adjust the input file to the new Herwig++-Matchbox version’s input parameters and manually feed Herwig++-Matchbox with the data from MadGraph5_aMC@NLO to have it invoke the parton shower step on the identical LHE file again. This setup will be called Combination 2.
3.2. Fixed order cross sections and results

Figure 3.3.: The old standard, the built-in matrix elements of Herwig++, was used in a fourth set of samples for the purpose of having a reliable comparison basis.

Combination 4

The fourth comparison counterpart, see figure 3.3, were the built-in (LO) Herwig++ matrix elements. They have been thoroughly validated during the past years and are supposed to be a trustworthy standard for the outcome.

3.2. Fixed order cross sections and results

This chapter discusses the parton level results obtained with the new preliminary Herwig++ version featuring the Matchbox framework for NLO calculations and matching (abbreviated to Herwig++-Matchbox henceforth) – more precisely the first combination described in the previous section 3.1.6.

In the context of this thesis, the term “parton level” is used synonymously for the term “fixed order”, which means it is applied to results before the parton shower step. Strictly speaking, results after the application of the parton shower but before hadronisation would still be called “parton level” results, but not “fixed order” anymore. However, this thesis does not discuss any samples at this intermediate level.

Later on, the term particle level is applied to samples at a level on which physical events (in the sense of particles that can be observed) are produced. It includes hadronisation and further steps beyond the parton shower step.

Detectors can never give direct insight to the processes prior to hadronisation, and owing to this, simulated parton level data is unphysical; parton level samples are never going to have an experimental counterpart. Nevertheless, there are good reasons for inspecting this first simulation step before proceeding to particle level:

- For one thing, the cross section can already be calculated at parton level (which is easier to compute than for the particle level) and can be compared to the results of other event generators at this level.
- A useful feature of parton level samples is the availability of generator information about the actual simulated partons.
- A test analysis is easier to accomplish at fixed order, compared to later event generation stages when one has to try and retrace everything starting from particle level information.
- Moreover, it is always sensible to start from the beginning and scrutinise a system step by step. Any peculiarities occuring at parton level would most likely be handed down to particle level.

In order to thoroughly test this new preliminary version of Herwig++, two processes (top pair production and associated Higgs production with top quarks) were studied at different energies for the LHC and with different PDF sets. The simulation of a proton-proton collision resulting in the creation of a top quark and top antiquark pair (commonly
abbreviated to $t\bar{t}$) is well-studied, and the availability of comparison material makes this an ideal starting point. As a second process the associated Higgs production with top quarks, which features a top quark, top antiquark, and Higgs boson in the final state (commonly abbreviated to $t\bar{t}H$), was considered. This is especially interesting in the context of the discovery of the Higgs boson, and works as a suitable extension to the first process.

Regarding the centre-of-mass energies at which the proton collisions were simulated, the four values that are also being examined at the LHC were chosen: 7 and 8 TeV as in Run 1, and 13 and 14 TeV as in the now launched Run 2.

Further choices were made concerning the PDF sets to be applied. As different empirical sets have been computed with different interpolation algorithms, taking varying measurement data into account, this decision might well influence the outcome of the simulation. In the past, widely employed PDF sets were the CTEQ6 PDFs and their (NLO) successors, the CT10 series. Another, newer option is the NNPDF30 set which was extracted from combined measurement data obtained at HERA-II, ATLAS, LHCb, and CMS.

While the CTEQ and CT PDFs have been obtained through a fitting process of a pre-parametrised form, the NNPDF is the result of a fitting process using a neural network in combination with experimental data.

As this thesis also aims at the comparison of LO and NLO results, the NLO edition of a PDF set was employed where available: The LHAPDF $B^{\pm \alpha}$ PDF sets in table 3.1 were applied with the noted corresponding $\alpha_s$ values at values for the Z boson mass $M_Z$, which was also explicitly set for the simulations.

<table>
<thead>
<tr>
<th>PDF name</th>
<th>$\alpha_s, M_Z$</th>
<th>$M_Z$ in GeV</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTEQ6L1</td>
<td>0.129783</td>
<td>91.1876</td>
</tr>
<tr>
<td>CT10</td>
<td>0.117982</td>
<td>91.1876</td>
</tr>
<tr>
<td>CT10NLO</td>
<td>0.118001</td>
<td>91.1876</td>
</tr>
<tr>
<td>NNPDF30_nlo_as_0118</td>
<td>0.1180000</td>
<td>91.199997</td>
</tr>
</tbody>
</table>

In the scope of this examination of the parton level results, for all combinations of the above mentioned parameters LO and NLO samples with 2 million events per sample were simulated. The choice of a fixed scale $\mu$ for the factorisation and renormalisation scale for the $t\bar{t}$ process corresponds to the top quark mass $\mu = m_t = 173.5$ GeV; while for the $t\bar{t}H$ process, the top quark mass plus half of the Higgs boson mass $\mu = m_t + \frac{m_H}{2} = 235$ GeV was used. Generating the $t\bar{t}$ process requires two QCD vertices in the corresponding Feynman diagrams; the $t\bar{t}H$ process registers with two QCD vertices and one electroweak vertex in Herwig++. Orders of $\alpha_s$ and $\alpha$ were set accordingly in the input files $^2$.

All event generation was conducted using the five flavour scheme (5FS). The alternative, a four flavour scheme (4FS), would have meant neglecting the bottom quark as a constituent of the initial state protons. Consequently, calculations with a 4FS are expected to be slightly less close to experimental results than with a 5FS owing to the neglect of the bottom quark mass in the initial state. Surely this simplification is sometimes unavoidable, as specific processes cannot yet be calculated in the 5FS. However, it is not necessary to use the 4FS for simulating the two processes under study.

$^2$Electroweak vertices are named QED in the input card of Herwig++.
As a first check, a look at the cross section results is illuminating.

Figures 3.4 to 3.7 show the cross section results for t¯t and t¯tH for the different PDFs at all selected energies. It is common practice to give the uncertainties by varying the factorisation and renormalisation scales simultaneously, once by a factor of 0.5 and once by a factor of 2.0. This was done, as a reference, for the NNPDF30 samples. The error bars span a relatively large range at LO. All cross sections, regardless of the PDF they were calculated with, lie well within these bounds. Evidently, the results vary less with the scales for the NLO calculations, as was to be expected. All cross section results are still within the bounds. Even the sample generated with CTEQ6L1, which is the only tested non-NLO PDF, behaves agreeably.

A rough estimate of the ratio between the NLO and LO cross sections (NLO K-factor) lies between 1.46 and 1.5 for the t¯t process, and in the range of 1.18 and 1.25 for the t¯tH process for the CT10, CT10nlo and the NNPDF30 PDF respectively. Variations in the K-factor depend on the energy and the PDF. The K-factors of cross sections computed with the LO PDF CTEQ6L1 are noticeably larger for both the t¯t process (around 1.8) and the t¯tH process (around 1.4). All obtained K-factors are listed in tables A.5 and A.6 in the appendix.

The cross section results can further be compared to the ATLAS-CMS recommended predictions [CM13] which are listed in table 3.2 along with the with Herwig++-Matchbox.
obtained values. Owing to the fact that the The CT10 as well as the CT10nlo PDF was tested. Since they barely differ, only one of them is listed here. All of the Herwig++-Matchbox LO and NLO cross sections as well as their uncertainties can be found in tables A.1 to A.4 in the appendix.

Table 3.2.: Obtained $t\bar{t}$ NLO cross section results compared to the CMS-ATLAS recommended NNLO values $^{[CM13]}$. As NLO and NNLO cannot be directly compared, differences are expected. The obtained NLO cross sections are given with statistical uncertainties.

<table>
<thead>
<tr>
<th>Centre-of-mass energy (TeV)</th>
<th>Recommen. NNLO value $\pm$ scale variation</th>
<th>CTEQ6L1 $\sigma_{t\bar{t}} \pm \Delta_{\text{stat}}$</th>
<th>CT10nlo $\sigma_{t\bar{t}} \pm \Delta_{\text{stat}}$</th>
<th>NNPDF30 $\sigma_{t\bar{t}} \pm \Delta_{\text{stat}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sqrt{s}$ in pb</td>
<td>in pb</td>
<td>in pb</td>
<td>in pb</td>
<td>in pb</td>
</tr>
<tr>
<td>7</td>
<td>$173.60^{+4.46}_{-5.85} +8.85 +5.32$</td>
<td>$149.88^{+0.15}_{-0.15}$</td>
<td>$147.49^{+0.15}_{-0.15}$</td>
<td>$155.25^{+0.16}_{-0.16}$</td>
</tr>
<tr>
<td>8</td>
<td>$247.74^{+6.26}_{-8.45} +11.47 +7.41$</td>
<td>$217.92^{+0.22}_{-0.22}$</td>
<td>$211.16^{+0.21}_{-0.21}$</td>
<td>$222.71^{+0.22}_{-0.22}$</td>
</tr>
<tr>
<td>13</td>
<td>$815.96^{+19.37}_{-28.61} +34.38 +22.67$</td>
<td>$763.95^{+0.81}_{-0.81}$</td>
<td>$698.66^{+0.73}_{-0.73}$</td>
<td>$732.76^{+0.75}_{-0.75}$</td>
</tr>
<tr>
<td>14</td>
<td>$966.01^{+22.68}_{-33.89} +40.52 +26.54$</td>
<td>$913.78^{+0.98}_{-0.98}$</td>
<td>$827.28^{+0.85}_{-0.85}$</td>
<td>$867.96^{+0.90}_{-0.90}$</td>
</tr>
</tbody>
</table>

The official values for the $t\bar{t}$ cross sections are only available at NNLO, while the calculations with Herwig++ are at NLO precision. The recommended values are thus not really comparable, however they can serve as a landmark. A further cross section comparison is done in section 3.2.1 to results obtained with Sherpa.

The $t\bar{t}H$ recommended values, however, are provided in NLO accuracy and are listed in table 3.3 along with the fixed order Herwig++-Matchbox results.

Table 3.3.: Obtained $t\bar{t}H$ NLO cross section results compared to the LHC Higgs Cross Section Working Group recommended NLO values $^{[LHC13]}$. The applied particle masses differ slightly, a possible explanation for small deviations. The obtained NLO cross sections are given with statistical uncertainties.

<table>
<thead>
<tr>
<th>Centre-of-mass energy (TeV)</th>
<th>Recommen. NLO value $\pm$ scale variation</th>
<th>CTEQ6L1 $\sigma_{t\bar{t}H} \pm \Delta_{\text{stat}}$</th>
<th>CT10nlo $\sigma_{t\bar{t}H} \pm \Delta_{\text{stat}}$</th>
<th>NNPDF30 $\sigma_{t\bar{t}H} \pm \Delta_{\text{stat}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sqrt{s}$ in pb</td>
<td>in pb</td>
<td>in pb</td>
<td>in pb</td>
<td>in pb</td>
</tr>
<tr>
<td>7</td>
<td>$0.0843^{+0.0027}_{-0.0078} +0.0071$</td>
<td>$0.0820^{+0.0001}_{-0.0001}$</td>
<td>$0.0832^{+0.0001}_{-0.0001}$</td>
<td>$0.0869^{+0.0001}_{-0.0001}$</td>
</tr>
<tr>
<td>8</td>
<td>$0.1262^{+0.0048}_{-0.0017} +0.0102$</td>
<td>$0.1227^{+0.0002}_{-0.0002}$</td>
<td>$0.1252^{+0.0002}_{-0.0002}$</td>
<td>$0.1303^{+0.0002}_{-0.0002}$</td>
</tr>
<tr>
<td>13</td>
<td>$0.4966^{+0.0281}_{-0.0462} +0.0437$</td>
<td>$0.4896^{+0.0007}_{-0.0007}$</td>
<td>$0.4814^{+0.0007}_{-0.0007}$</td>
<td>$0.5045^{+0.0007}_{-0.0007}$</td>
</tr>
<tr>
<td>14</td>
<td>$0.5969^{+0.0352}_{-0.0555} +0.0531$</td>
<td>$0.5975^{+0.0009}_{-0.0009}$</td>
<td>$0.5818^{+0.0008}_{-0.0008}$</td>
<td>$0.6090^{+0.0009}_{-0.0009}$</td>
</tr>
</tbody>
</table>

A difference would have been plausible to arise between the recommended and obtained values from the difference in chosen top quark and Higgs boson masses. For this bachelor
3.2. Fixed order cross sections and results

thesis, a top quark mass of 173.5 GeV and a Higgs boson mass of 125.9 GeV were consistently used, whereas the CMS-ATLAS values were calculated with a top quark mass of 173.2 GeV in case of the $t\bar{t}$ cross sections respectively a top quark mass of 172.5 GeV and a Higgs boson mass of 126.0 GeV in case of the $t\bar{t}H$ process. Naturally, as the programs and algorithms are not the same, many other parameters do not coincide. However, this does not seem to have a great impact: the obtained values match the recommended values and they are without exception well inside the uncertainty bands.

Applying the fixed order Rivet analysis described in the following section 3.2.2 yields plots of the distributions of various observables of the final state particles. Some selected examples shall be given here.

The 1.9.0 version of Rivet is known not to be able to calculate the statistical uncertainties of NLO samples correctly \cite{B15}. Therefore, as the ratio plot has no validity, it will be skipped in all following NLO plots.

Figures 3.8 and 3.9 show the resulting top quark transverse momentum $p_T$ distribution of $t\bar{t}$ events for collisions at 8 TeV. The plot on the left-hand side is the result at LO and the right-hand side plot was produced with identical parameters at NLO precision. The top antiquark plots behave accordingly as expected within statistics. The results obtained with different PDFs are in good agreement with each other – both at LO and NLO – in regard to the top quark transverse momentum $p_{T,t}$ distribution.

Figures 3.10 and 3.11 display the Higgs boson rapidity $y$ distribution resulting from $t\bar{t}H$ events at 14 TeV centre-of-mass energy.

Evidently the LO plots produce the anticipated shape in this case. Owing to the normalisation to the same cross section, the agreement of the LO and NLO distributions obtained with different PDFs is apparent.

Again similar results for different PDFs can be observed.

A feature of the NLO plots is the slight fluctuation occurring especially at higher energies and for few bin entries in many of the plots. This has its root in the negative event weights.
Validation of the Herwig++-Matchbox framework

Figure 3.10.: LO $t\bar{t}H$ result for the Higgs boson rapidity distribution at 14 TeV for different PDFs using Herwig++-Matchbox.

Figure 3.11.: NLO $t\bar{t}H$ result for the Higgs boson rapidity distribution at 14 TeV for different PDFs using Herwig++-Matchbox.

that arise in NLO calculations, in contrast to the LO calculations. The comparison between the weight distributions at LO and NLO shown in figures 3.12 and 3.13 reveals the reason as to why NLO plots need more events for the same statistical significance. While the LO weights are positive and contribute constructively to the number of events, the NLO weights display a shape with a large amount of negative weights. These partly counteract the positive weights, resulting in a much lower effective number of events for NLO. The examples are taken from the 8 TeV $t\bar{t}$ samples.

Secondly, every now and then positive weights get sorted into one bin, while the corresponding negative weights from subtraction terms end up in the neighbouring bin. These mis-binning effects occur irrespective of the bin width and can not be altogether
3.2. Fixed order cross sections and results

avoided when using Rivet. Currently the program cannot handle these misbinning issues in the necessary fashion. Some built-in Herwig analyses are designed for this purpose, however, a t\bar{t} or t\bar{t}H analysis is not existent. Implementing the required steps into an own analysis would have exceeded the time frame of this thesis.

However, the source of the fluctuations is inherent to and fully expected for the way that Herwig++-Matchbox operates, namely calculating NLO subtraction terms with the Catani-Seymour Dipole Subtraction Method [CS96, CS]. In fact, such behaviour is expected for a typical NLO Monte Carlo simulation without proper bin-smearing techniques. Therefore, these fluctuations are not to be viewed as an error, but merely as the expression of a not yet fully concerted interplay of the independent projects Herwig++ and Rivet.

Further, there are no unexpected differences in the shapes of the plots for varying the energy from 7 TeV up to 14 TeV. Increasing the centre-of-mass energy results in wider rapidity distributions and generally higher transverse momenta. Two examples are given in figures 3.14 and 3.15.

Thus, concerning the t\bar{t} and t\bar{t}H processes at fixed order, the Herwig++-Matchbox combination works without displaying obvious flaws and overall according to expectation. The output demonstrated reliability in comparison to recommended results.

With another look at the cross section comparison plots 3.4 to 3.7, the difference attributed to the choice of one of these PDFs is negligible, since it lies within the uncertainties given by scale variation, and the shapes of all tested observables is identical for all applied PDF sets. It is consequently justified to proceed with only one PDF at particle level. The NNPDF30 was chosen for this. Furthermore, investigating at 8 TeV and 13 TeV has thereby turned out to be sufficient.
3. Validation of the Herwig++-Matchbox framework

3.2.1. Comparison to Sherpa plus OpenLoops

Another thesis, which was written at this institute around the same time, used the event generator Sherpa \cite{G+09, CMP11, vH10, OPP07, GH08, GK08, SK08, KKS} to study the same processes ($t\bar{t}$ and $t\bar{t}H$) with similar parameter choices – such as the PDF, energy and LO/NLO precision. Sherpa was used with an interface to OpenLoops, which served as a provider for one-loop matrix elements. A selection of the thus obtained parton level results makes up the counterpart for another cross check with the Herwig++-Matchbox framework in this section.

The cross section values are listed in tables A.7 to A.10 in the appendix and illustrated in figures 3.16 to 3.19 fully analogously to the cross section figures 3.4 to 3.7 in the previous parton level section 3.2.

As a reference point, the NNPDF30 cross section which was obtained with Herwig++-Matchbox is again given, including the error bars obtained from scale variation (straight
For the purpose of comparison, the NNPDF30 cross section obtained with Sherpa+OpenLoops including its scale variation error bars is also given (V-shaped error bar ends). Except for the t\bar{t}H NLO cross sections in figure 3.19, all central values always lie within both given error bars. In the t\bar{t}H NLO comparison plot, the error bands still overlap; therefore, all cross section results of Sherpa and Herwig++ for the studied PDFs are in agreement.

### 3.2.2. Fixed order Rivet analysis

Owing to the much more basic information level at fixed order before the parton shower step, the analyses provided with Rivet cannot usually be applied to parton level events; they operate at particle level following experimental needs.

The fixed order Rivet t\bar{t} analysis utilised in the previous section was provided by Daniel Rauch [Rau14], and has been slightly adjusted in terms of the output. Additionally, a t\bar{t}H Rivet analysis has been derived from it. Both can be found in the appendices D.6 and D.7. A short description of the functionality of these fixed order Rivet analyses is given here.

Analysing Monte Carlo simulated data at fixed order is comparatively straightforward, especially if handling exclusive samples: Final state particles requested during the event generation (e.g. top quark and top antiquark for the process pp \rightarrow t\bar{t}) will unconditionally and exclusively appear in the Rivet final state projection, as only these are produced by the matrix element event generator. They can be identified via their Particle Data Group (PDG) ID [G04] which is stored in the data set.

The Rivet namespace provides easy access to various attributes of those identified particles. Starting from the four-momentum \( p \), the transverse momentum \( p_T \), rapidity \( y \), invariant mass \( m \), or other quantities may be extracted and filled into histograms as the program loops over all events.

Instead of writing out the huge amount of information of all events produced by Herwig++-Matchbox to a HepMC file – in this case containing the parton level results – the Rivet interface implemented in Herwig++ was used to analyse the events on-the-fly. This leads to an output in form of an AIDA file\(^3\) containing all Rivet histogram information rather than a disk space-consuming HepMC file with all the “raw” event generator parton level data.

---

\(^3\)AIDA is a format used by Rivet for histogramming in order to avoid dependencies on more commonly used but extensive frameworks like Root. The AIDA framework is superseded by the YODA framework in newer Rivet 2.X versions, but is nevertheless very useful.
3.3. Particle level

This part of the thesis focuses on the parton shower step with subsequent hadronisation, both of which follow the event generation at parton level. The program output thus contains information at particle level.

Results obtained with various combinations of programs are presented and compared.

3.3.1. Construction of the Rivet $t\bar{t}H$ analysis for particle level samples

Following the conclusions of section 3.2, it should be sufficient to study the new Herwig++-Matchbox framework further by comparing just two of the four previously applied centre-of-mass energies. The differences have proven to be small at parton level, and any unwanted deviation from this predictable behaviour would in any case show obviously enough in the gap between a centre-of-mass energy of $\sqrt{s} = 8$ TeV and 13 TeV. These two energy parameters have consequently been applied for the study of the parton shower and hadronisation step, leading to information at particle level.

The focus in the upcoming part is on the $t\bar{t}H$ process. It is a more sophisticated process than $t\bar{t}$ to simulate and analyse, as it contains three final state particles at the matrix element level. During the parton shower stage all three of them may decay into lighter particles through several possible decay channels.

This makes an analysis at particle level quite complicated: A reconstructed bottom quark, for example, could have been produced by a decayed top quark, top antiquark, or the Higgs boson. Tremendous work is being invested by many collaborations in the ongoing development and perfection of t\bar{t}H analyses.

As a matter of fact, no particle level Rivet $t\bar{t}H$ analysis was available for the usage in this thesis. Therefore, an own crude attempt had to be made.

In the course of the validation of the new Herwig++-Matchbox version it is perfectly acceptable to constrain the decay channels of the Higgs boson to simplify the $t\bar{t}H$ analysis -- and test another option, the constraining of the decay channel, of Herwig++ along the way. This approach is applicable, since the typical approximation of factorising production and decay are used; the decay of the Higgs boson should be straightforward to simulate and neither sensitive to the hard subprocess nor to the parton shower.

A top quark is highly likely to decay into a W boson and a bottom quark, the probability being close to 1. The subsequent decay of a W boson can be hadronic or leptonically, leading to the following $t\bar{t}$ decay channels $[O^{+14c}]$:

\[
\begin{align*}
t\bar{t} &\rightarrow W^+ b W^- \bar{b} \rightarrow q q' b q'' \bar{q} m \bar{q} m & \quad \text{fullhadronic (45.7\%)} \\
t\bar{t} &\rightarrow W^+ b W^- \bar{b} \rightarrow q q' b q'' \ell\bar{\nu} \ell' \bar{\nu} m & \quad \text{semileptonic (43.8\%)} \quad (3.1) \\
t\bar{t} &\rightarrow W^+ b W^- \bar{b} \rightarrow \tilde{\ell} \nu \ell' \bar{\nu} m & \quad \text{dileptonic (10.5\%)}
\end{align*}
\]

The Standard Model Higgs boson is expected to have significant branching ratios for decays to a photon pair ($\gamma\gamma$), two W or Z bosons ($WW$, $ZZ$), two tau leptons ($\tau\tau$), and a bottom quark-antiquark pair ($bb$) $[O^{+14a}]$.

However, as the particle level analysis was to be used for the comparison of the different event generator combinations, and did not need to be applicable on real data sets, the rare but clean decay channel $H \rightarrow \mu^+ \mu^-$ was chosen. An illustration of the semileptonic

\[4A\] short test of the $t\bar{t}$ process at particle level showed no signs of irregular behaviour of the Herwig++-Matchbox framework.
tt channel in combination with the $H \rightarrow \mu^+\mu^-$ decay channel is given in figure 3.20. This decay channel has the advantage that few other muons occur in the $t\bar{t}H$ process otherwise. The corresponding estimated branching ratio, which is a measure of the probability for this decay channel ($t\bar{t}H$, $H \rightarrow \mu^+\mu^-$), implemented in the current developer version of Herwig++-Matchbox, is 0.000218.

![Figure 3.20](image)

**Figure 3.20.** An illustration of the process $pp \rightarrow t\bar{t}H$ with a subsequent decay of the top quark and top antiquark through the semileptonic channel, and a Higgs boson decay to $\mu^+$ and $\mu^-$. Adapted from [DiH11].

The analysis was nevertheless constructed in a way that allows for the application on a sample with non-restricted decay of the Higgs boson and might even perform reasonably well on inclusive data samples that are not limited to the process $t\bar{t}H$. The latter has not been tested though, and the former is not advisable owing to very low statistical precision.

The Rivet Monte Carlo analysis "MC_TTBAR" [B+12] served as a basis.

To begin with, the Rivet FinalState projection is applied twice on the simulated data provided by the HepMC file. The first projection returns all particles, selecting only those with pseudorapidities of less than $|\eta| < 4.2$ and transverse momenta of more than $p_T > 30$ GeV. This happens with the intention of reducing the impact of the underlying event which features mainly softer particles. Another FinalState projection is applied to return all jets in the same pseudorapidity region but without the $p_T$ cut. A summary of these initial cuts is given in table 3.4.

In order to isolate desirable decay channels of the top quark and top antiquark, further phase space cuts are applied to the particle level data. Requiring two muons from the Higgs decay and an additional lepton from a leptonically decaying W boson does reduce the background from other QCD and also EW processes considerably. A reason for discarding the fullhadronic decay channel is also the time-intensive combination and trial of all jets initiated by b-quarks in succeeding analysis steps which would otherwise become necessary.

Additionally the condition of two muons with opposite charges from the Higgs decay is imposed for further event selection.
Table 3.4.: Overview of the applied initial pseudorapidity and transverse momentum cuts in the Rivet particle level analysis.

<table>
<thead>
<tr>
<th>Applied FinalState projection returns:</th>
<th>Parameter affected by the applied cut</th>
<th>Resulting range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Particles</td>
<td>pseudorapidity</td>
<td>$</td>
</tr>
<tr>
<td>Particles</td>
<td>transverse momentum</td>
<td>$p_T &gt; 30$ GeV</td>
</tr>
<tr>
<td>Jets</td>
<td>pseudorapidity</td>
<td>$</td>
</tr>
</tbody>
</table>

It can be concluded from the information about the top decay in eq. (3.1) that exactly two b-jets are expected in total. Further jets might only occur in the case of a semileptonic scenario now, as the fullhadronic case has already been discarded. The quark jets stemming from the W boson decay are most unlikely to contain any further b-jets, because the W boson must decay into an up-type and a down-type quark. However, the up-type partner of the bottom quark is the top quark with a mass that is higher than the mass of the W boson\(^5\).

This means that at least four jets with exactly two “hard” b-jets with large transverse momenta $p_T$ can be required in each event, focussing on the semileptonic $t\bar{t}$ decay channel. Additionally, further $p_T$ cuts are imposed upon the four hardest jets, since the $t\bar{t}$ decay should primarily result in jets with high transverse momenta. Among these four hardest jets, two well isolated jets from b-quarks are then required.

The first recombination step is to check all possible $\mu^+\mu^-$ combinations for the resulting four-vector with an invariant mass closest to the expected Higgs mass of $m_H = 125.9$ GeV, vetoing all events that do not yield any Higgs boson candidate within the mass frame of $120.9 - 130.9$ GeV.

Subsequently, the hadronically decayed W boson is reconstructed, taking the left-over jets that have not been tagged as a b-jet by the FinalState projection. The routine keeps track of the most likely W boson candidate with the mass closest to the experimentally well-known expected value. In the last step, the hadronically decayed W boson of each event is paired once with each of the two bottom quarks. It is not obvious from particle level data which b-jet originally belonged to the W boson, therefore both top quarks reconstructed in this way are considered viable candidates.

During the analysis, several cuts are thus applied to the sample. A basic cut flow histogram has been added to the analysis during the construction in order to keep an eye on these cuts which are afflicting the statistical significance quite perceptibly. After each veto step, the weight of the discarded event is filled into the bin that is reserved for this veto step; exceptions are the first, last and second-last bin (all three coloured), which hold the total or remaining sum of event weights. An exemplary result is shown in figure 3.21.

The first bin contains the full sum of all event weights of the original sample; the succeeding bins show the discarded event weights at cuts described in table 3.5. Eventually, the 11th histogram bin at the central value 9.5 depicts a preliminary result of the event weights that have passed all requirements, where the subsequent W mass cut does not make a huge difference anymore. The remaining events are approximately 5 % of the original generated number of events.

\(^5\)b-jets may only occur in this decay because of the quark generation mixing, with the mixing parameters given by the Cabibbo-Kobayashi-Maskawa Matrix, as for example given in O'14b.
3.3. Particle level

Figure 3.21.: A representative cut flow histogram for the Rivet $t\bar{t}H$ analysis in the semileptonic $t\bar{t}$ and $H \rightarrow \mu^+\mu^-$ channel. This example shows the number of events vetoed at each cut stage for a LO sample at 8 TeV. The first coloured bin shows the original number of events, the coloured bins at 9.5 and 10.5 depict the total sum of weights of remaining events before and after the $W$ mass cut.

Table 3.5.: This table shows the discarded sum of weights of vetoed events at each cut in the particle level analysis written for the $t\bar{t}H$ process in the $H \rightarrow \mu^+\mu^-$ channel. An exception are the histogram bins 1, 11 and 12. These hold the remaining weights. It is an example taken from the LO sample at 8 TeV.

<table>
<thead>
<tr>
<th>Histogram bin</th>
<th>Central value</th>
<th>Applied cut</th>
<th>Sum of weights of events effected by the cut</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.5</td>
<td>Original number of events</td>
<td>2,000,000</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>Lepton multiplicity cut</td>
<td>1,407,020</td>
</tr>
<tr>
<td>3</td>
<td>1.5</td>
<td>Muon multiplicity cut</td>
<td>6,881</td>
</tr>
<tr>
<td>4</td>
<td>2.5</td>
<td>Muon charge cut</td>
<td>7,584</td>
</tr>
<tr>
<td>5</td>
<td>3.5</td>
<td>Missing transverse energy cut</td>
<td>25,775</td>
</tr>
<tr>
<td>6</td>
<td>4.5</td>
<td>Jet multiplicity cut</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>5.5</td>
<td>Jet cuts</td>
<td>264,210</td>
</tr>
<tr>
<td>8</td>
<td>6.5</td>
<td>Post-lepton-isolation b-tagging cut</td>
<td>169,410</td>
</tr>
<tr>
<td>9</td>
<td>7.5</td>
<td>Not enough light jets remaining after lepton-isolation</td>
<td>8,123</td>
</tr>
<tr>
<td>10</td>
<td>8.5</td>
<td>Higgs mass cut</td>
<td>4,468</td>
</tr>
<tr>
<td>11</td>
<td>9.5</td>
<td>Remaining events</td>
<td>106,529</td>
</tr>
<tr>
<td>12</td>
<td>10.5</td>
<td>Remaining events after $W$ mass cut</td>
<td>80,462</td>
</tr>
</tbody>
</table>
### 3.3.2. Results of the Herwig++-Matchbox framework

Having a rudimentary but working analysis for the process of interest \( t\bar{t}H \) at hand, the particle level studies were commenced. Centre-of-mass energies of \( \sqrt{s} = 8 \) and 13 TeV were studied.

In this section, the results obtained from the Herwig++-Matchbox framework are summarised. For a more detailed description of the setup of this first combination of event generators, please refer to section 3.1.6 and figure 3.1 therein.

Because of the fact that Herwig++-Matchbox has not yet been officially released at the time of writing, the changeset of the development version used for the samples is given here. While an earlier version (changeset 5660:df71d6c28c57 from May 20, 2015) worked well for the fixed-order and also particle level LO studies, this same version displayed unphysical behaviour concerning the Higgs transverse momentum at particle level at NLO precision. Therefore, after this had been fixed, a newer version (changeset 5777:4549011ca20b from June 17, 2015) was chosen to compute the NLO samples. An exemplary input file can be found in the appendix in figure B.1.

As a first consistency check, not least between the different changesets, the cross sections can be compared to the fixed-order samples discussed in chapter 3.2. The cross sections are listed in table 3.6 and display reasonable agreement between parton level and particle level.

<table>
<thead>
<tr>
<th>Energy</th>
<th>Precision</th>
<th>Parton level</th>
<th>Particle level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \sigma ) in pb</td>
<td>( \sigma ) in pb</td>
<td></td>
</tr>
<tr>
<td>8 TeV</td>
<td>LO</td>
<td>0.109165 ± 0.000088</td>
<td>0.1000 ± 0.0001</td>
</tr>
<tr>
<td></td>
<td>NLO</td>
<td>0.130304 ± 0.000157</td>
<td>0.128 ± 0.002</td>
</tr>
<tr>
<td>13 TeV</td>
<td>LO</td>
<td>0.405259 ± 0.000313</td>
<td>0.3797 ± 0.0004</td>
</tr>
<tr>
<td></td>
<td>NLO</td>
<td>0.504531 ± 0.000698</td>
<td>0.52 ± 0.01</td>
</tr>
</tbody>
</table>

This simultaneously means that the obtained particle level cross sections are also compatible with the “recommended” CMS values listed in table 3.3 in the parton level section 3.2.

The Herwig++-Matchbox framework also offers the possibility to vary the shower scale \( \mu_Q \) separately from the renormalisation and factorisation scales \( \mu_R = \mu_F \). However, the attempt to get the conventional error bars via scale variation, as it was done in the parton level section 3.2, failed owing to a considerably slowed down parton shower in the changeset 5660:df71d6c28c57 from May 20, 2015, with which the LO samples were conducted. This rendered the computation unfeasible. This error was corrected in later changesets again.

A LO to NLO comparison of some observables is done in the following step. Figures 3.22 to 3.25 depict the LO and NLO results obtained with the Herwig++-Matchbox framework for some representative quantities at energies of \( \sqrt{s} = 8 \) and 13 TeV. Overall, the LO and NLO results – not only for the displayed top quark mass distribution, W mass and Higgs boson energy – are in good agreement at both energies. A difference can be observed in the jet multiplicity depicted in figure 3.25, however. At NLO precision, more hard jets are present owing to additional gluon radiation already at parton level. Consequently, higher jet multiplicities are expected and the result plot displays this tendency accordingly.
While the fluctuations in the NLO plots were acceptable in the parton level sample plots – partly because at this stage, one could argue with the fact that physical behaviour is not necessarily provided for technical reasons, the fluctuations are unphysical and should have been smoothed out by the parton shower and hadronisation step at the latest.

Still, low statistical power is probably the reason for the occurrence of fluctuations in the obtained NLO plots. From the originally demanded and computed number of events, two million in case of the particle level samples, the bulk falls prey to the very basic, self-written Rivet analysis outlined in the previous section 3.3.1. The remaining, roughly 100,000 events cannot be expected to display smooth shapes for statistical reasons. Even these few remaining events are weighted events and must be judged differently than an equal amount of unweighted events. The following estimation serves the purpose of gaining a rough understanding of the statistical significance of 100,000 weighted events.

The variance of an efficiency \( \epsilon = \frac{N_{\text{sel}}}{N} \) from an unweighted sample with \( N \) events is given by

\[
V[\epsilon] = \frac{\epsilon (1 - \epsilon)}{N}.
\]

(3.2)

The variance of a weighted equivalent can be derived from the initial weighted expression of \( \epsilon = \frac{\sum_i w_i \epsilon_i}{\sum_i w_i} \) and subsequent error propagation assuming \( V[\epsilon_i] = \epsilon_i (1 - \epsilon_i) \):

\[
V[\epsilon] = \sum_i \left( \frac{\partial \epsilon}{\partial \epsilon_i} \right)^2 V[\epsilon_i] = \frac{\sum_i w_i^2 \epsilon_i (1 - \epsilon_i)}{(\sum_i w_i)^2}
\]

(3.3)

Comparison of (3.2) and (3.3) leads to an effective number of events

\[
N_{\text{eff}} = \frac{(\sum_i w_i)^2}{\sum_i w_i^2}
\]

(3.4)

for a weighted sample. Following this formula, the effective number of events e.g. of the particle level NLO sample at 13 TeV diminishes from \( N = 2,000,000 \) to \( N_{\text{eff}} = 2776 \) preceding the analysis.
In order to visualize the meaning of this, a LO sample with the same effective number of events \( N_{\text{eff}} \) was generated. After application of the analysis, a mere few hundred events remain. This “fair” LO-NLO comparison with an accordingly lower number of events for the LO sample results in exemplary figure 3.26. It strongly suggests that indeed low statistical power is the reason for the fluctuation issue. The two shapes now make a similarly distorted impression.

While this effective number of events testifies a low statistical power, it was not feasible in the scope of this thesis to further raise the computed number of events owing to storage and computation time requirements. However, in spite of the fluctuations, it has become obvious from all general LO to NLO comparison plots that also at NLO particle level, the Herwig++-Matchbox output remains consistent with expectations.

A concluding note should be that improvement regarding any fluctuations that might not originate from low statistical significance is still well possible before the new Herwig++ version is released.
3.3.3. Comparison to MadGraph5_aMC@NLO and Herwig++-Matchbox

The comparison of the results of different event generators makes sense after the completion of parton shower and hadronisation steps for the first time. Only then do the various technical approaches of different generators claim to yield physical results which describe observables in nature for a given model and from which the expected outcome of particle physics experiments can be deduced.

As outlined in section 3.1.6 and sketched in figure 3.2 the second combination to be compared to the Herwig++-Matchbox results is MadGraph5_aMC@NLO with subsequent Herwig++-Matchbox parton shower and hadronisation step.

Essentially, there are two differences to the previous section: First, the matrix element generation and matching are now assigned entirely to MadGraph5_aMC@NLO, with the settings given in figure C.3. Further, the invocation of Herwig++ is – but for a slight adaptation to the new Herwig++-Matchbox version of the input file (see figure B.2 in the appendix), which is automatically generated by MadGraph5_aMC@NLO for the older Herwig++ version 2.7.1 – steered by MadGraph5_aMC@NLO. This includes the setting of cuts and parameters.

While parameters like the PDF set (NNPDF30), the fundamental particle masses (see appendix, figure C.4), and the running QCD coupling constant ($\alpha_s = 0.118$) have been adjusted, it is impractical to adjust more hidden parameters, cuts, or whole structures and procedures. At some point, it is sensible to credit an existing difference to the fact that one is comparing two separate generators. The manual interventions were kept to the above-mentioned parameters regarding this comparison between Combination 1 on one hand, and Combination 2 and 3 on the other hand.

A direct comparison between Combination 1 and Combination 2 reveals close resemblance between the results of the two sets of event generators: In figures 3.27 and 3.28 the LO results for the Higgs boson transverse momentum at 13 TeV and the reconstructed top quark mass distribution after the W mass cut at 8 TeV are shown in juxtaposition with each other; figures 3.29 and 3.30 compare the NLO results of the Higgs boson transverse momentum at 8 TeV, respectively of the reconstructed top quark energy distribution at 8 TeV.

![Figure 3.27. LO particle level t\bar{t}H results in the muon channel obtained with Combination 1 and 2 at 13 TeV for the Higgs transverse momentum distribution.](image1)

![Figure 3.28. LO particle level t\bar{t}H results in the muon channel obtained with Combination 1 and 2 at 8 TeV for the top quark mass after the application of the W mass cut.](image2)
Apparently, the NLO results of Combination 2 look smoother than the previously discussed Herwig++-Matchbox results. As precedently reasoned, this most probably has its root in the large fraction of negative weights of the Herwig++-Matchbox samples, whereas MadGraph5_aMC@NLO uses a different subtraction method. A “fair” comparison was conducted for these samples as well, following the procedure described in the previous section 3.3.2. Figure 3.31 shows the comparison plot and again, the shapes now make a similarly unstable impression due to statistical reasons.

![Figure 3.29](image1.png)  
Figure 3.29.: NLO particle level $t\bar{t}H$ results in the muon channel obtained with Combination 1 and 2 at 8 TeV for the Higgs transverse momentum distribution.

![Figure 3.30](image2.png)  
Figure 3.30.: NLO particle level $t\bar{t}H$ results in the muon channel obtained with Combination 1 and 2 at 8 TeV for the top quark energy distribution.

![Figure 3.31](image3.png)  
Figure 3.31.: According to the definition of the efficient number of events given in equation 3.4 in the previous section, this plot shows a fair comparison between Combination 1 (Herwig++-Matchbox) and Combination 2 (MadGraph5_aMC@NLO with subsequent Herwig++ parton shower) samples with both $N_{eff} = 2776$.

While the LO sample 3.32 displays systematically only a slightly higher jet multiplicity when produced with the Herwig++-Matchbox samples, the NLO sample 3.33 looks balanced. This means that no significant difference can be made out concerning this quantity between the two combinations, either.

This cross check with another tried and tested event generator like MadGraph5_aMC@NLO confirms the reliability of the Herwig++-Matchbox framework particle level results further.
3.3. Particle level

3.3.1. Introduction

The results presented in this section were computed with MadGraph5_aMC@NLO up to parton level as well. In contrast to the previous section, the shower step was handled by the old Herwig++ version 2.7.1. This section is supposed to give further evidence towards the comparability of the new Matchbox framework.

3.3.2. Comparison to MadGraph5_aMC@NLO and Herwig++ 2.7.1

The results discussed here were computed with MadGraph5_aMC@NLO up to parton level as well. In contrast to the previous section, the shower step was handled by the old Herwig++ version 2.7.1. This section is supposed to give further evidence towards the comparability of the new Matchbox framework.

3.3.3. Results

Exemplary plots are shown in figures 3.34 to 3.37. The comparison again results in an overall impression of similarity between the shapes produced by Combination 1 and 3.

However, a difference in the jet multiplicity plots 3.38 and 3.39 is visible. A trend towards higher jet multiplicities seems to be inherent to the new Herwig++ version compared to the older version 2.7.1.

Identical parton level LHE files were used for the production of the Combination 2 and 3 samples. Thus, a comparison between the results yielded from Combinations 2 and 3
provides access to isolated information about the differences between the two Herwig++ versions.

The comparison between Combination 2 and 3 reveals strikingly similar shapes for almost all quantities obtained with the new version and the older version 2.7.1. This specifically creates a picture of credibility of the Herwig++ parton shower and hadronisation model functionality.

The only exception to this observation is again the jet multiplicity; The tendency to obtain higher numbers of jets seems to be inherent to the newer Herwig++ version. This trend can be observed in figure 3.40 which shows the jet multiplicity at 8 TeV for NLO precision as an example.

As the changesets which were used in this thesis have not yet undergone the tuning procedure of the hadronisation model, this difference in jet multiplicity might vanish. It would be interesting to check this quantity again with a version that is ready for release.
3.3.5. Comparison to the LO matrix elements of Herwig++ 2.7.1

As described in section 3.1.6, the fourth examined “combination” were the LO matrix elements for the $t\bar{t}H$ process which have been hard coded in Herwig++. The resulting particle level LO samples, which were obtained with the version 2.7.1, are the comparison counterpart to the Herwig++-Matchbox framework’s LO results (Combination 1) in this section.

Interestingly, the obtained cross sections differ from the results in section 3.3.2.

To further understand this, samples with 1,000,000 events were produced with an identical setup file with the new Herwig++ version. Table 3.7 lists all obtained LO cross sections, all taken from samples with 1,000,000 events. The two samples obtained from the built-in LO $t\bar{t}H$ matrix elements show a missing factor of 2. Further, the sample which was completed with the older Herwig++ version displays a deviation by several orders of magnitude. This can be traced back to the branching ratio 0.000218 of the examined muon channel. Apparently, the output value in the first column of table 3.7 is not the inclusive cross section for the $t\bar{t}H$ process, but the exclusive cross section for the $H \to \mu^+\mu^-$ channel.

Table 3.7.: Comparison of the obtained LO cross sections for Combination 1 (Herwig++-Matchbox) and Combination 4 (built-in LO MEs) with the subsequent parton shower step computed by two different Herwig++ versions. It seems that a factor of 2 is missing in the built-in LO MEs, explaining the difference between the second and third column. Further, the approximate factor 0.000218 between the first and second column may possibly be ascribed to the branching ratio of the $H \to \mu^+\mu^-$ channel and its internal treatment. The ATLAS-CMS “recommended values” are the same as in section 3.2.

<table>
<thead>
<tr>
<th>$\sqrt{s}$ (TeV)</th>
<th>Combi 4: Built-in MEs, HW++ 2.7.1</th>
<th>Combi 4: Built-in MEs, new HW++ version</th>
<th>Recomm. value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\sigma_{tt} \pm \Delta_{\text{stat}}$</td>
<td>$\sigma_{tt} \pm \Delta_{\text{stat}}$</td>
<td>$\pm$ scale variation</td>
</tr>
<tr>
<td></td>
<td>in pb</td>
<td>in pb</td>
<td>$\pm$ PDF+$\alpha_S$</td>
</tr>
<tr>
<td>8 TeV</td>
<td>$0.01326(1) \cdot 10^{-3}$</td>
<td>$0.05734(3)$</td>
<td>0.1262 $^{+3.8}<em>{-9.3}$ $^{+8.1}</em>{-8.1}$</td>
</tr>
<tr>
<td>13 TeV</td>
<td>$0.05266(2) \cdot 10^{-3}$</td>
<td>$0.2277(1)$</td>
<td>0.4966 $^{+5.7}<em>{-9.3}$ $^{+8.8}</em>{-8.8}$</td>
</tr>
</tbody>
</table>
However, neglecting the cross section discrepancies, the obtained plots can be compared to all previous results. Some fundamental parameters like the W boson mass seem to differ in the initial matrix elements, leading to somewhat different distributions. Figures 3.41 and 3.42 show representative examples.

Figure 3.41.: Comparison of the obtained particle level $t\bar{t}H$ results in the muon channel at 8 TeV. The sample “Combination 4” was obtained with the built-in LO MEs of Herwig++ 2.7.1. The plot displays the $W$ boson mass distribution in comparison to the Herwig++-Matchbox result (Combination 1).

Figure 3.42.: Comparison of the obtained particle level $t\bar{t}H$ results in the muon channel at 8 TeV. The sample “Combination 4” was obtained with the built-in LO MEs of Herwig++ 2.7.1. The plot displays the Higgs boson transverse momentum distribution in comparison to the Herwig++-Matchbox result (Combination 1).

It appears that, beside the difference in the parton shower and hadronisation step between the Herwig++ versions discussed in the previous section, the difference between the MadGraph5 ME and the hard coded Herwig++ ME is contributing to some further deviation. The shapes are therefore only roughly similar.
4. Summary

Monte Carlo event generators like Herwig++ play a crucial part in the simulation of LHC physics. Herwig++ is such an event generator.

A contribution to the validation of the Herwig++-Matchbox framework, with focus on the study of the $t\bar{t}H$ process, was achieved in this thesis. The parton level studies in section 3.2 already gave a first impression of the already good quality of the results.

The cross section results of the new Herwig++-Matchbox framework, upon which the new and soon to be published Herwig++ version will be based, coincide with official ATLAS-CMS recommendations as well as Sherpa+OpenLoops results. Although some fluctuations occur in the shapes of fixed order NLO plots, these are not a reason for concern at this stage of the event generation, as several viable explanations seem perfectly reasonable; the most probable cause being mis-binning issues that occur due to the intended presence of negative weights in the Herwig++ samples combined with the current weight handling in the Rivet analysis applied in this thesis.

The central part of this thesis, the particle level Herwig++-Matchbox results presented in section 3.3.2 have further strengthened the picture of an overall reliable event generation with the upcoming Herwig++ version. Studies of the $H \rightarrow \mu^+\mu^-$ decay channel of the $t\bar{t}H$ process yielded Herwig++-Matchbox results that were comparable to the results of more established versions or other event generators. Figures 4.1 and 4.2 show representative comparison plots with results of all four studied combinations of event generators for the Higgs boson transverse momentum.

Fluctuations that still occurred in the Herwig++-Matchbox particle level samples have been attributed to the low effective sample size. The use of weighted events and events with negative weights owing to the Catani-Seymour subtraction method results in reduced effective numbers of events. Furthermore, a large number of events was lost in the course of analysing the samples with a crude self-written Rivet analysis. The timeframe and resources of this thesis did not allow for a repetition of the sample generation with a notably higher effective number of events. However, the shapes of all observed quantities are well recognisable and agree with results from other combinations of event generators.

Even during the period the samples were produced, the Herwig++-Matchbox development version has seen copious improvements. As the release date still lies in the future at the time of writing, this thesis does not represent the status of any official Herwig++ version. Already now, the output and performance of the new Herwig++-Matchbox make a trustworthy impression. Naturally, further validation procedures are needed – and already being conducted – in order to contribute to a more detailed picture.
Combination 1: Herwig++-Matchbox with MadGraph5-aMC@NLO and OpenLoops ME
Combination 2: MadGraph5-aMC@NLO with subsequent Herwig++-Matchbox PS
Combination 3: MadGraph5-aMC@NLO with subsequent Herwig++ 2.7.1 PS
Combination 4: Herwig++ 2.7.1 built-in LO ME and Herwig++ 2.7.1 PS

Figure 4.1.: LO \( t\bar{t}H \) results in the muon channel at 13 TeV for the Higgs boson transverse momentum.

Figure 4.2.: NLO \( t\bar{t}H \) results in the muon channel at 8 TeV for the Higgs boson transverse momentum.
Bibliography


5. Appendix
Appendix

A. Cross sections

<table>
<thead>
<tr>
<th>√s (TeV)</th>
<th>CTEQ6L1 [pb]</th>
<th>CT10 [pb]</th>
<th>CT10nlo [pb]</th>
<th>NNPDF30 [pb]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>σ_{t\bar{t}} ± Δ_{stat}</td>
<td>σ_{t\bar{t}} ± Δ_{stat}</td>
<td>σ_{t\bar{t}} ± Δ_{stat}</td>
<td>σ_{t\bar{t}} ± Δ_{stat} ± scale var.</td>
</tr>
<tr>
<td>7</td>
<td>83.92 ±0.07(^{+0.07}_{-0.07})</td>
<td>100.74 ±0.08(^{+0.08}_{-0.08})</td>
<td>100.53 ±0.08(^{+0.08}_{-0.08})</td>
<td>105.89 ±0.08(^{+36.27}_{-24.03})</td>
</tr>
<tr>
<td>8</td>
<td>121.24 ±0.09(^{+0.09}_{-0.09})</td>
<td>143.69 ±0.12(^{+0.12}_{-0.12})</td>
<td>143.44 ±0.11(^{+0.11}_{-0.11})</td>
<td>151.15 ±0.11(^{+49.82}_{-33.32})</td>
</tr>
<tr>
<td>13</td>
<td>418.32 ±0.30(^{+0.30}_{-0.30})</td>
<td>470.19 ±0.33(^{+0.33}_{-0.33})</td>
<td>469.77 ±0.33(^{+0.33}_{-0.33})</td>
<td>493.07 ±0.35(^{+140.12}_{-97.29})</td>
</tr>
<tr>
<td>14</td>
<td>499.40 ±0.98(^{+0.98}_{-0.98})</td>
<td>556.20 ±0.39(^{+0.39}_{-0.39})</td>
<td>555.70 ±0.45(^{+0.45}_{-0.45})</td>
<td>582.64 ±0.41(^{+161.53}_{-112.96})</td>
</tr>
</tbody>
</table>

Table A.2.: \(t\bar{t}\) NLO cross section results obtained with the Herwig++-
Matchbox framework.

<table>
<thead>
<tr>
<th>√s (TeV)</th>
<th>CTEQ6L1 [pb]</th>
<th>CT10 [pb]</th>
<th>CT10nlo [pb]</th>
<th>NNPDF30 [pb]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>σ_{t\bar{t}} ± Δ_{stat}</td>
<td>σ_{t\bar{t}} ± Δ_{stat}</td>
<td>σ_{t\bar{t}} ± Δ_{stat}</td>
<td>σ_{t\bar{t}} ± Δ_{stat} ± scale var.</td>
</tr>
<tr>
<td>7</td>
<td>149.88 ±0.15(^{+0.15}_{-0.15})</td>
<td>147.81 ±0.15(^{+0.15}_{-0.15})</td>
<td>147.49 ±0.15(^{+0.15}_{-0.15})</td>
<td>155.25 ±0.16(^{+19.31}_{-17.58})</td>
</tr>
<tr>
<td>8</td>
<td>217.92 ±0.22(^{+0.22}_{-0.22})</td>
<td>211.55 ±0.21(^{+0.21}_{-0.21})</td>
<td>211.16 ±0.21(^{+0.21}_{-0.21})</td>
<td>222.71 ±0.22(^{+26.48}_{-24.73})</td>
</tr>
<tr>
<td>13</td>
<td>763.95 ±0.81(^{+0.81}_{-0.81})</td>
<td>699.12 ±0.73(^{+0.73}_{-0.73})</td>
<td>698.66 ±0.73(^{+0.73}_{-0.73})</td>
<td>732.76 ±0.75(^{+85.91}_{-72.26})</td>
</tr>
<tr>
<td>14</td>
<td>913.78 ±0.98(^{+0.98}_{-0.98})</td>
<td>827.77 ±0.85(^{+0.85}_{-0.85})</td>
<td>827.28 ±0.85(^{+0.85}_{-0.85})</td>
<td>867.96 ±0.89(^{+99.92}_{-86.69})</td>
</tr>
</tbody>
</table>

49
Table A.3: \( \bar{t}H \) LO cross section results obtained with the Herwig++ - Matchbox framework.

<table>
<thead>
<tr>
<th>( \sqrt{s} )</th>
<th>CTEQ6L1 [pb]</th>
<th>CT10 [pb]</th>
<th>CT10nlo [pb]</th>
<th>NNPDF30 [pb]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \sigma_{\bar{t}H} \pm \Delta_{stat} )</td>
<td>( \sigma_{\bar{t}H} \pm \Delta_{stat} )</td>
<td>( \sigma_{\bar{t}H} \pm \Delta_{stat} )</td>
<td>( \sigma_{\bar{t}H} \pm \Delta_{stat} \pm \text{scale var.} )</td>
</tr>
<tr>
<td>7 TeV</td>
<td>0.05955 ( \pm 0.00005 ) ( -0.00005 )</td>
<td>0.07090 ( \pm 0.00006 ) ( -0.00006 )</td>
<td>0.07071 ( \pm 0.00006 ) ( -0.00006 )</td>
<td>0.07351 ( \pm 0.00006 ) ( +0.02493 ) ( -0.01686 )</td>
</tr>
<tr>
<td>8 TeV</td>
<td>0.08815 ( \pm 0.00007 ) ( -0.00007 )</td>
<td>0.10503 ( \pm 0.00008 ) ( -0.00008 )</td>
<td>0.10476 ( \pm 0.00008 ) ( -0.00008 )</td>
<td>0.10917 ( \pm 0.00009 ) ( +0.03904 ) ( -0.02456 )</td>
</tr>
<tr>
<td>13 TeV</td>
<td>0.33078 ( \pm 0.00026 ) ( -0.00026 )</td>
<td>0.38804 ( \pm 0.00030 ) ( -0.00030 )</td>
<td>0.38743 ( \pm 0.00030 ) ( -0.00030 )</td>
<td>0.40526 ( \pm 0.00031 ) ( +0.1969 ) ( -0.08409 )</td>
</tr>
<tr>
<td>14 TeV</td>
<td>0.40012 ( \pm 0.00031 ) ( -0.00031 )</td>
<td>0.46687 ( \pm 0.00036 ) ( -0.00036 )</td>
<td>0.46615 ( \pm 0.00036 ) ( -0.00036 )</td>
<td>0.48780 ( \pm 0.00037 ) ( +0.14140 ) ( -0.09971 )</td>
</tr>
</tbody>
</table>

Table A.4: \( \bar{t}H \) NLO cross section results obtained with the Herwig++ - Matchbox framework.

<table>
<thead>
<tr>
<th>( \sqrt{s} )</th>
<th>CTEQ6L1 [pb]</th>
<th>CT10 [pb]</th>
<th>CT10nlo [pb]</th>
<th>NNPDF30 [pb]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \sigma_{\bar{t}H} \pm \Delta_{stat} )</td>
<td>( \sigma_{\bar{t}H} \pm \Delta_{stat} )</td>
<td>( \sigma_{\bar{t}H} \pm \Delta_{stat} )</td>
<td>( \sigma_{\bar{t}H} \pm \Delta_{stat} \pm \text{scale var.} )</td>
</tr>
<tr>
<td>7 TeV</td>
<td>0.0820 ( \pm 0.0001 ) ( -0.0001 )</td>
<td>0.0836 ( \pm 0.0001 ) ( -0.0001 )</td>
<td>0.0832 ( \pm 0.0001 ) ( -0.0001 )</td>
<td>0.0869 ( \pm 0.0001 ) ( +0.0020 ) ( -0.0068 )</td>
</tr>
<tr>
<td>8 TeV</td>
<td>0.1227 ( \pm 0.0002 ) ( -0.0002 )</td>
<td>0.1254 ( \pm 0.0002 ) ( -0.0002 )</td>
<td>0.1252 ( \pm 0.0002 ) ( -0.0002 )</td>
<td>0.1303 ( \pm 0.0002 ) ( +0.0045 ) ( -0.0098 )</td>
</tr>
<tr>
<td>13 TeV</td>
<td>0.4896 ( \pm 0.0007 ) ( -0.0007 )</td>
<td>0.4821 ( \pm 0.0006 ) ( -0.0006 )</td>
<td>0.4814 ( \pm 0.0007 ) ( -0.0007 )</td>
<td>0.5045 ( \pm 0.0007 ) ( +0.0232 ) ( -0.0390 )</td>
</tr>
<tr>
<td>14 TeV</td>
<td>0.5975 ( \pm 0.0009 ) ( -0.0009 )</td>
<td>0.5816 ( \pm 0.0009 ) ( -0.0009 )</td>
<td>0.5818 ( \pm 0.0008 ) ( -0.0008 )</td>
<td>0.6090 ( \pm 0.0009 ) ( +0.0319 ) ( -0.0449 )</td>
</tr>
</tbody>
</table>

Table A.5: \( \bar{t}H \) NLO K-factors obtained with the Herwig++ - Matchbox framework for different PDF sets.

<table>
<thead>
<tr>
<th>( \sqrt{s} )</th>
<th>CTEQ6L1</th>
<th>CT10</th>
<th>CT10nlo</th>
<th>NNPDF30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( K \pm \Delta_{stat} )</td>
<td>( K \pm \Delta_{stat} )</td>
<td>( K \pm \Delta_{stat} )</td>
<td>( K \pm \Delta_{stat} )</td>
</tr>
<tr>
<td>7 TeV</td>
<td>1.776 ( \pm 0.014 ) ( -0.014 )</td>
<td>1.467 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.467 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.466 ( \pm 0.002 ) ( -0.002 )</td>
</tr>
<tr>
<td>8 TeV</td>
<td>1.797 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.472 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.472 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.473 ( \pm 0.002 ) ( -0.002 )</td>
</tr>
<tr>
<td>13 TeV</td>
<td>1.826 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.487 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.487 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.486 ( \pm 0.002 ) ( -0.002 )</td>
</tr>
<tr>
<td>14 TeV</td>
<td>1.830 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.488 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.489 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.490 ( \pm 0.002 ) ( -0.002 )</td>
</tr>
</tbody>
</table>

Table A.6: \( \bar{t}H \) NLO K-factors obtained with the Herwig++ - Matchbox framework for different PDF sets.

<table>
<thead>
<tr>
<th>( \sqrt{s} )</th>
<th>CTEQ6L1</th>
<th>CT10</th>
<th>CT10nlo</th>
<th>NNPDF30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( K \pm \Delta_{stat} )</td>
<td>( K \pm \Delta_{stat} )</td>
<td>( K \pm \Delta_{stat} )</td>
<td>( K \pm \Delta_{stat} )</td>
</tr>
<tr>
<td>7 TeV</td>
<td>1.377 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.180 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.176 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.182 ( \pm 0.002 ) ( -0.002 )</td>
</tr>
<tr>
<td>8 TeV</td>
<td>1.392 ( \pm 0.003 ) ( -0.003 )</td>
<td>1.194 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.195 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.194 ( \pm 0.002 ) ( -0.002 )</td>
</tr>
<tr>
<td>13 TeV</td>
<td>1.480 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.242 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.243 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.245 ( \pm 0.002 ) ( -0.002 )</td>
</tr>
<tr>
<td>14 TeV</td>
<td>1.493 ( \pm 0.003 ) ( -0.003 )</td>
<td>1.246 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.248 ( \pm 0.002 ) ( -0.002 )</td>
<td>1.248 ( \pm 0.002 ) ( -0.002 )</td>
</tr>
</tbody>
</table>
### Table A.7: $t\bar{t}$ LO cross section results obtained with Sherpa. The obtained cross sections are given with statistical uncertainties.

<table>
<thead>
<tr>
<th>$\sqrt{s}$</th>
<th>Sherpa CTEQ6L1 $\sigma_{t\bar{t}}$ ± scale var. in pb</th>
<th>Sherpa CT10nlo $\sigma_{t\bar{t}}$ ± scale var. in pb</th>
<th>Sherpa NNPDF30 $\sigma_{t\bar{t}}$ ± scale var. in pb</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 TeV</td>
<td>$96.21^{+35.90}_{-24.11}$</td>
<td>$96.22^{+33.29}_{-23.07}$</td>
<td>$101.29^{+35.62}_{-24.59}$</td>
</tr>
<tr>
<td>8 TeV</td>
<td>$139.09^{+50.43}_{-34.37}$</td>
<td>$137.12^{+45.92}_{-31.98}$</td>
<td>$144.39^{+49.29}_{-34.05}$</td>
</tr>
<tr>
<td>13 TeV</td>
<td>$479.52^{+156.08}_{-109.87}$</td>
<td>$449.66^{+131.24}_{-95.30}$</td>
<td>$471.76^{+139.81}_{-100.99}$</td>
</tr>
<tr>
<td>14 TeV</td>
<td>$572.44^{+183.21}_{-129.17}$</td>
<td>$531.92^{+152.39}_{-110.92}$</td>
<td>$557.82^{+160.77}_{-117.09}$</td>
</tr>
</tbody>
</table>

### Table A.8: $t\bar{t}H$ LO cross section results obtained with Sherpa. The obtained cross sections are given with statistical uncertainties.

<table>
<thead>
<tr>
<th>$\sqrt{s}$</th>
<th>Sherpa CTEQ6L1 $\sigma_{t\bar{t}H}$ ± scale var. in pb</th>
<th>Sherpa CT10nlo $\sigma_{t\bar{t}H}$ ± scale var. in pb</th>
<th>Sherpa NNPDF30 $\sigma_{t\bar{t}H}$ ± scale var. in pb</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 TeV</td>
<td>$143.63^{+19.82}_{-17.84}$</td>
<td>$141.76^{+16.23}_{-18.52}$</td>
<td>$148.99^{+17.52}_{-20.44}$</td>
</tr>
<tr>
<td>8 TeV</td>
<td>$210.65^{+28.00}_{-28.84}$</td>
<td>$203.54^{+21.90}_{-26.89}$</td>
<td>$213.58^{+25.31}_{-27.80}$</td>
</tr>
<tr>
<td>13 TeV</td>
<td>$741.43^{+92.10}_{-90.93}$</td>
<td>$672.48^{+74.71}_{-76.35}$</td>
<td>$705.08^{+78.62}_{-86.15}$</td>
</tr>
<tr>
<td>14 TeV</td>
<td>$885.41^{+102.22}_{-117.52}$</td>
<td>$795.39^{+93.94}_{-86.47}$</td>
<td>$832.23^{+97.19}_{-91.33}$</td>
</tr>
</tbody>
</table>

### Table A.9: $t\bar{t}H$ LO cross section results obtained with Sherpa. The obtained cross sections are given with statistical uncertainties.

<table>
<thead>
<tr>
<th>$\sqrt{s}$</th>
<th>Sherpa CTEQ6L1 $\sigma_{t\bar{t}H}$ ± scale var. in pb</th>
<th>Sherpa CT10nlo $\sigma_{t\bar{t}H}$ ± scale var. in pb</th>
<th>Sherpa NNPDF30 $\sigma_{t\bar{t}H}$ ± scale var. in pb</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 TeV</td>
<td>$0.06613^{+0.02409}_{-0.01627}$</td>
<td>$0.06486^{+0.02286}_{-0.01570}$</td>
<td>$0.06741^{+0.02419}_{-0.01645}$</td>
</tr>
<tr>
<td>8 TeV</td>
<td>$0.09791^{+0.03513}_{-0.02373}$</td>
<td>$0.09609^{+0.03310}_{-0.02273}$</td>
<td>$0.10016^{+0.03504}_{-0.02386}$</td>
</tr>
<tr>
<td>13 TeV</td>
<td>$0.36884^{+0.12309}_{-0.08598}$</td>
<td>$0.35794^{+0.11152}_{-0.07914}$</td>
<td>$0.37485^{+0.11785}_{-0.08380}$</td>
</tr>
<tr>
<td>14 TeV</td>
<td>$0.44697^{+0.14707}_{-0.10410}$</td>
<td>$0.43161^{+0.13118}_{-0.09425}$</td>
<td>$0.45151^{+0.13900}_{-0.09927}$</td>
</tr>
</tbody>
</table>

### Table A.10: $t\bar{t}H$ NLO cross section results obtained with Sherpa. The obtained cross sections are given with statistical uncertainties.

<table>
<thead>
<tr>
<th>$\sqrt{s}$</th>
<th>Sherpa CTEQ6L1 $\sigma_{t\bar{t}H}$ ± scale var. in pb</th>
<th>Sherpa CT10nlo $\sigma_{t\bar{t}H}$ ± scale var. in pb</th>
<th>Sherpa NNPDF30 $\sigma_{t\bar{t}H}$ ± scale var. in pb</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 TeV</td>
<td>$0.0777^{+0.0020}_{-0.0067}$</td>
<td>$0.0777^{+0.0021}_{-0.0079}$</td>
<td>$0.0806^{+0.0025}_{-0.0077}$</td>
</tr>
<tr>
<td>8 TeV</td>
<td>$0.1158^{+0.0043}_{-0.0087}$</td>
<td>$0.1155^{+0.0050}_{-0.0105}$</td>
<td>$0.1214^{+0.0036}_{-0.0126}$</td>
</tr>
<tr>
<td>13 TeV</td>
<td>$0.4674^{+0.0190}_{-0.0447}$</td>
<td>$0.4445^{+0.0236}_{-0.0387}$</td>
<td>$0.4651^{+0.0282}_{-0.0413}$</td>
</tr>
<tr>
<td>14 TeV</td>
<td>$0.5632^{+0.0462}_{-0.0482}$</td>
<td>$0.5430^{+0.0415}_{-0.0551}$</td>
<td>$0.5636^{+0.0391}_{-0.0502}$</td>
</tr>
</tbody>
</table>
B. Input files for Herwig++

Figure B.1: Example of an input file. This was used to generate the 8 TeV NLO sample.

```
# Herwig++/Matchbox example input file

Collider type

read Matchbox/PPCollider.in

cd /Herwig/EventHandlers
set EventHandler:LuminosityFunction:Energy 8000*GeV
set EventHandler:Weighted On

# Process selection

# Note that event generation may fail if no matching matrix element has been found. Coupling orders are with respect to the Born process, i.e. NLO QCD does not require an additional power of alphas.

# Set the order of the couplings

cd /Herwig/MatrixElements/Matchbox
set Factory:OrderInAlphaS 2
set Factory:OrderInAlphaEW 1

# Select the process
# You may use identifiers such as p, pbar, j, l, mu+, h0 etc.
do Factory:Process p p -> t tbar h0

# Matrix element library selection

# Select a generic tree/loop combination or a specialized NLO package

read Matchbox/MadGraph-GoSam.in
read Matchbox/MadGraph-MadGraph.in
read Matchbox/MadGraph-NJet.in
read Matchbox/MadGraph-OpenLoops.in
read Matchbox/HJets.in
read Matchbox/VBFNLO.in

# Uncomment this to use ggh effective couplings currently only supported by MadGraph-GoSam
read Matchbox/HiggsEffective.in

# Cut selection
# See the documentation for more options
```
## B. Input files for Herwig++

```plaintext
## cuts on additional jets
# read Matchbox/DefaultPPJets.in

# insert JetCuts:JetRegions 0 FirstJet
# insert JetCuts:JetRegions 1 SecondJet
# insert JetCuts:JetRegions 2 ThirdJet
# insert JetCuts:JetRegions 3 FourthJet

# Scale choice
## See the documentation for more options

cd /Herwig/MatrixElements/Matchbox
# set Factory:ScaleChoice /Herwig/MatrixElements/Matchbox/Scales/
# LeptonPairMassScale
set Scales/FixedScale:FixedScale 235*GeV
# set Factory:ScaleChoice /Herwig/MatrixElements/Matchbox/Scales/FixedScale

## Matching and shower selection

read Matchbox/MCatNLO−DefaultShower.in
# read Matchbox/Powheg−DefaultShower.in
## use for strict LO/NLO comparisons
# read Matchbox/MCatLO−DefaultShower.in
## use for improved LO showering
# read Matchbox/LO−DefaultShower.in

# read Matchbox/MCatNLO−DipoleShower.in
# read Matchbox/Powheg−DipoleShower.in
## use for strict LO/NLO comparisons
# read Matchbox/MCatLO−DipoleShower.in
## use for improved LO showering
# read Matchbox/LO−DipoleShower.in

# read Matchbox/NLO−NoShower.in
# read Matchbox/LO−NoShower.in

## Scale uncertainties

## Shower scale uncertainties

# read Matchbox/MuDown.in
# read Matchbox/MuUp.in

## PDF choice

# read Matchbox/FiveFlavourScheme.in
# do /Herwig/MatrixElements/Matchbox/Amplitudes/OpenLoops:Massless 5
# do /Herwig/MatrixElements/Matchbox/Amplitudes/OpenLoops:Massless −5
```
cd /Herwig/Partons

### CT10
### 
# set HardLOPDF:PDFName CT10 . LHgrid
# set HardNLOPDF:PDFName CT10 . LHgrid
#
# set ShowerLOPDF:PDFName CT10 . LHgrid
# set ShowerNLOPDF:PDFName CT10 . LHgrid
#
# set MPIPDF:PDFName CT10 . LHgrid
#
# cd /Herwig/MatrixElements/Matchbox/AlphaS
#
# set MatchboxLOAlphaS: input_scale 91.1876*GeV
# set MatchboxLOAlphaS: input_alpha_s 0.117982
#
# set MatchboxNLOAlphaS: input_scale 91.1876*GeV
# set MatchboxNLOAlphaS: input_alpha_s 0.117982

### NNPDF30_nlo_as_0118
### 
set HardLOPDF:PDFName NNPDF30_nlo_as_0118 . LHgrid
set HardNLOPDF:PDFName NNPDF30_nlo_as_0118 . LHgrid
set ShowerLOPDF:PDFName NNPDF30_nlo_as_0118 . LHgrid
set ShowerNLOPDF:PDFName NNPDF30_nlo_as_0118 . LHgrid
set MPIPDF:PDFName NNPDF30_nlo_as_0118 . LHgrid

cd /Herwig/MatrixElements/Matchbox/AlphaS

set MatchboxLOAlphaS: input_scale 91.199997*GeV
set MatchboxLOAlphaS: input_alpha_s 0.1180000
set MatchboxNLOAlphaS: input_scale 91.199997*GeV
set MatchboxNLOAlphaS: input_alpha_s 0.1180000

### cteq6l1
### 
set HardLOPDF:PDFName cteq6l1 . LHpdf
set HardNLOPDF:PDFName cteq6l1 . LHpdf
set ShowerLOPDF:PDFName cteq6l1 . LHpdf
set ShowerNLOPDF:PDFName cteq6l1 . LHpdf
#
set MPIPDF:PDFName cteq6l1 . LHpdf
#
# cd /Herwig/MatrixElements/Matchbox/AlphaS
#
# set MatchboxLOAlphaS: input_scale 91.199997*GeV
# set MatchboxLOAlphaS: input_alpha_s 0.1180000
#
# set MatchboxNLOAlphaS: input_scale 91.1876*GeV
# set MatchboxNLOAlphaS: input_alpha_s 0.129783
#
# set /Herwig/Model:QCD/RunningAlphaS /Herwig/MatrixElements/Matchbox/AlphaS/MatchboxNLOAlphaS
cd /Herwig/Analysis

## Rivet

create ThePEG::NLORivetAnalysis Rivet RivetAnalysis.so
create ThePEG::RivetAnalysis Rivet RivetAnalysis.so

insert Rivet::Analyses 0 XXX_2015_ABC123
insert Rivet::Analyses 0 TTBAR_STABLE_FO

insert /Herwig/Generators/EventGenerator::AnalysisHandlers 0 Rivet

## HepMC

In order to get all the generated events written to a HepMC file, one needs to set HepMCFile:PrintEvent to a number larger than the number of generated events.

create ThePEG::NLOHepMCFile HepMC HepMCAnalysis.so
create ThePEG::HepMCFile HepMC HepMCAnalysis.so

set /Herwig/Analysis/HepMC:PrintEvent 10000000
set /Herwig/Analysis/HepMCFile:PrintEvent 100000000
set /Herwig/Analysis/HepMC:Format GenEvent
set /Herwig/Analysis/HepMC:Units GeV

insert /Herwig/Generators/EventGenerator::AnalysisHandlers 0 HepMC

## Underlying Event off

set /Herwig/Shower/ShowerHandler:MPIHandler NULL

Particle properties for the top

set /Herwig/Particles/t:HardProcessMass 173.5*GeV
set /Herwig/Particles/t:NominalMass 173.5*GeV
set /Herwig/Particles/tbar:HardProcessMass 173.5*GeV
set /Herwig/Particles/tbar:NominalMass 173.5*GeV
set /Herwig/Particles/t:HardProcessWidth 0*GeV
set /Herwig/Particles/tbar:HardProcessWidth 0*GeV
set /Herwig/Particles/h0:HardProcessWidth 0*GeV

set /Herwig/Particles/t:Width 0*GeV
set /Herwig/Particles/tbar:Width 0*GeV
set /Herwig/Particles/t:Stable Stable
set /Herwig/Particles/tbar:Stable Stable

# read Matchbox/IncreaseVerbosity.in

set /Herwig/Samplers/Sampler:Verbose On
set /Herwig/MatrixElements/Matchbox/Factory:InitVerbose On

Particle properties for the Higgs

do /Herwig/Particles/h0:SelectDecayModes h0->mu- mu+;

avoid exception in tth nlo shower

set /Herwig/EventHandlers/EventHandler:MaxLoop 1000000
set /Herwig/Generators/EventGenerator:MaxErrors 10000000
## Daniels grid sampler settings

```plaintext
set /Herwig/Samplers/CellGridSampler:InitialPoints 5000
set /Herwig/Samplers/CellGridSampler:ExplorationPoints 2000
set /Herwig/Samplers/CellGridSampler:ExplorationSteps 5
set /Herwig/Samplers/CellGridSampler:RemapperPoints 20000
set /Herwig/Samplers/CellGridSampler:GeneralMapperBins 20
```

## Different seed for second hepmc output

```plaintext
#set /Herwig/Generators/EventGenerator:RandomNumberGenerator:Seed 25381926
```

```plaintext
Figure B.2.: In order to pass the LHE files, which were generated using MadGraph5_aMC@NLO, to the new Herwig++ version, this input file was used. It also explicitly sets the correct PDF set, the NNPDF30.
```

## Necessary settings for running with MC@NLO events (do not modify)

```plaintext
library LesHouches.so
```

```plaintext
# 1.) NECESSARY SETTINGS FOR RUNNING WITH MC@NLO EVENTS (DO NOT MODIFY) #
```

```plaintext
set /Herwig/Shower/Evolver:HardVetoMode 1
set /Herwig/Shower/Evolver:HardVetoScaleSource 1
set /Herwig/Shower/Evolver:MECorrMode 0
```

```plaintext
# create the Handler & Reader
create ThePEG::LesHouchesFileReader /Herwig/EventHandlers/LHEReader
create ThePEG::LesHouchesEventHandler /Herwig/EventHandlers/LHEHandler
insert /Herwig/EventHandlers/LHEHandler:LesHouchesReaders 0 /Herwig/EventHandlers/LHEReader
```

```plaintext
set /Herwig/EventHandlers/LHEReader:AllowedToReOpen 0
set /Herwig/EventHandlers/LHEReader:MomentumTreatment RescaleEnergy
set /Herwig/EventHandlers/LHEReader:WeightWarnings 0
```

```plaintext
set /Herwig/EventHandlers/LHEHandler:WeightOption VarNegWeight
set /Herwig/EventHandlers/LHEHandler:PartonExtractor /Herwig/Partons/QCDExtractor
set /Herwig/EventHandlers/LHEHandler:CascadeHandler /Herwig/Shower/ShowerHandler
set /Herwig/EventHandlers/LHEHandler:HadronizationHandler /Herwig/Hadronization/ClusterHadHandler
set /Herwig/EventHandlers/LHEHandler:DecayHandler /Herwig/Decays/DecayHandler
```

```plaintext
#insert /Herwig/EventHandlers/LHEHandler:PreCascadeHandlers 0 /Herwig/NewPhysics/DecayHandler
```
set /Herwig/Generators/LHCGenerator:EventHandler /Herwig/EventHandlers/LHEHandler
set /Herwig/Generators/LHCGenerator:NumberOfEvents 2000000
set /Herwig/Generators/LHCGenerator:RandomNumberGenerator:Seed 5
set /Herwig/Generators/LHCGenerator:PrintEvent 2
set /Herwig/Generators/LHCGenerator:MaxErrors 20001
set /Herwig/Generators/LHCGenerator:DebugLevel 1

# Define PDF from MCatNLO. inputs PDFSET number
mkdir /LHAPDF
create ThePEG::LHAPDF /LHAPDF/MCNLOPDF ThePEGLHAPDF.so
set /LHAPDF/MCNLOPDF:PDFName NNPDF30nlo as 0118

set PDFSet:PDFName NNPDF30nlo as 0118. LHgrid
set PDFSet:RemnantHandler HadronRemnants
set /Herwig/EventHandlers/LHEReader:PDFA /Herwig/Partons/PDFSet
set /Herwig/EventHandlers/LHEReader:PDFB /Herwig/Partons/PDFSet
cd /

# DEFINE THE INPUT EVENT FILE
set /Herwig/EventHandlers/LHEReader:FileName /home/jhunt/Extern/Julia/Particle_level/Combi_3/aMCatNLO_HW2StoVLO/1M_seed5/Events/run_01_LO/Events.lhe

# 2.) DEFINE PHYSICS PARAMETERS FROM FILE <MCatNLO_MadFKS.inputs> (DO NOT MODIFY)

# TODO set the correct incoming particles

# set masses and widths
set /Herwig/Particles/e−:NominalMass 0.000510999
set /Herwig/Particles/e+::NominalMass 0.000510999
set /Herwig/Particles/μ−::NominalMass 0.105658
set /Herwig/Particles/μ+:::NominalMass 0.105658
set /Herwig/Particles/τ−::NominalMass 1.77699
set /Herwig/Particles/τ+:::NominalMass 1.77699
set /Herwig/Particles/W+:::NominalMass 80.419002
set /Herwig/Particles/W−::NominalMass 80.419002
set /Herwig/Particles/Z0::NominalMass 91.199997
set /Herwig/Particles/t::NominalMass 173.5
set /Herwig/Particles/tbar::NominalMass 173.5
set /Herwig/Particles/h0::NominalMass 125.9
set /Herwig/Particles/b::NominalMass 4.2
set /Herwig/Particles/bbar::NominalMass 4.2
set /Herwig/Particles/W+:Width 2.141
set /Herwig/Particles/W−:Width 2.141
set /Herwig/Particles/Z0::Width 2.4952
set /Herwig/Particles/t::Width 1.4
set /Herwig/Particles/tbar::Width 1.4
set /Herwig/Particles/h0::Width 0.004541

# 3.) ADDITIONAL SETTINGS

# SHOWER SETTINGS
set /Herwig/Shower/Evolver:IntrinsicPtGaussian 2.2+GeV

# CREATE AND APPLY CUTS
create ThePEG::Cuts /Herwig/Cuts/NoCuts
set /Herwig/EventHandlers/LHEReader:Cuts /Herwig/Cuts/NoCuts

# Switching off MPI
set /Herwig/Shower/ShowerHandler:MPIHandler NULL
# HERE YOU CAN TURN ON/OFF DECAY

**If** DecayHandler **is set equal to NULL,** then set CheckQuark equal to 0, to prevent a lot of warnings

```
#set /Herwig/EventHandlers/LHEHandler:DecayHandler NULL
#set /Herwig/Analysis/Basics:CheckQuark 0
```

**do** /Herwig/Particles/h0:SelectDecayModes none 

```
set /Herwig/Particles/h0/h0->mu-,mu+:OnOff On
```

# Prevent particles from decaying

**Particle names can be found in Herwig++/src/default/mesons.in or similar**

```
# pi0
set /Herwig/Particles/pi0:Stable Stable
```

### 4.) HW++ SETTINGS THAT ARE NOT ALLOWED TO BE TOUCHED BY THE USER

**Boost and reconstruction stuff**

```
set /Herwig/Shower/KinematicsReconstructor:ReconstructionOption General
set /Herwig/Shower/KinematicsReconstructor:InitialInitialBoostOption LongTransBoost
```

### 5.) CREATE THE ANALYZER AND SAVE THE RUN (DO NOT MODIFY)

```
# create MCatNLO::hepfortr hepfortr hepfortr.so
# insert /Herwig/Generators/LHCGenerator:AnalysisHandlers 0 hepfortr
```

**Useful analysis handlers for HepMC related output**

```
# Schematic overview of an event (requires —with-hepmc to be set at configure time)
# and the graphviz program 'dot' to produce a plot)
# create MCatNLO::.so
# insert /Herwig/Generators/LHCGenerator:AnalysisHandlers 0
# insert LHCGenerator:AnalysisHandlers 0 /Herwig/Analysis/Plot
# A HepMC dump file (requires —with-hepmc to be set at configure time)
# insert LHCGenerator:AnalysisHandlers 0 /Herwig/Analysis/HepMCFile
insert /Herwig/Generators/LHCGenerator:AnalysisHandlers 0 /Herwig/Analysis/HepMCFile
```

```
set /Herwig/Analysis/HepMCFile:PrintEvent 2000000
set /Herwig/Analysis/HepMCFile:Format GenEvent
set /Herwig/Analysis/HepMCFile:Units GeV mm
```

```
saverun MCATNLO/HERWIGPP /Herwig/Generators/LHCGenerator
```
C. MadGraph5_aMC@NLO cards

Figure C.3.: MadGraph5_aMC@NLO setting instructions which were used before generating the samples with a five flavour scheme.

```
#******************************************************************************
## MadGraph5_aMC@NLO
##
## VERSION 2.3.0 2015-07-01
##
## The MadGraph5_aMC@NLO Development Team – Find us at
## https://server06.fynu.ucl.ac.be/projects/madgraph
##
##******************************************************************************
## Command File for MadGraph5_aMC@NLO
##
## run as ./bin/mg5_aMC filename
##
##******************************************************************************
set group_subprocesses Auto
set ignore_six_quark_processes False
set loop_optimized_output True
set loop_color_flows False
set gauge_unitary
set complex_mass_scheme False
set max_npoint_for_channel 0
import model sm
define p = g u c d s u^+ c^- d^- s^-
define j = g u c d s u^+ c^- d^- s^-
define l+ = e+ mu+
define l- = e- mu-
define vl = ve vm vt
define vl'' = ve'' vm'' vt''
define p = p b b'
define j = p
import model loop_sm-no_b_mass
generate p p > t t^* h [QCD]
output aMCatNLO_HW2_8TeV_NLO
```
Figure C.4: *MadGraph5 aMC@NLO parameter card*

```
# PARAMETER CARD AUTOMATICALLY GENERATED BY MG5 FOLLOWING UFO MODEL   ###
###
### Width set on Auto will be computed following the information  ###
### present in the decay.py files of the model.  ###
### See arXiv:1402.1178 for more details.  ###
###
###
### INFORMATION FOR LOOP
Block loop
  1 9.118800e+01 # MURRE
###
### INFORMATION FOR MASS
###
Block mass
  5 4.200000e+00 # MB
  6 1.735000e+02 # MT
  15 1.777000e+00 # MTA
  23 9.119997e+01 # MZ
  25 1.259000e+02 # MH
### Dependent parameters, given by model restrictions.
### Those values should be edited following the analytical expression. MG5 ignores those values
### but they are important for interfacing the output of MG5 to external programs such as Pythia.
  1 0.000000 # d : 0.0
  2 0.000000 # u : 0.0
  3 0.000000 # s : 0.0
  4 0.000000 # c : 0.0
  11 0.000000 # e- : 0.0
  12 0.000000 # ve : 0.0
  13 0.000000 # mu- : 0.0
  14 0.000000 # vm : 0.0
  16 0.000000 # vt : 0.0
  21 0.000000 # g : 0.0
  22 0.000000 # a : 0.0
  24 80.419002 # w+ : cmath.sqrt(MZ_exp^2/2. + cmath.sqrt(MZ_exp^4/4. - (aEW*cmath.pi*MZ_exp^2)/(Gf*sqrt(2))))
###
### INFORMATION FOR SMINPUTS
###
Block sminputs
  1 1.325070e+02 # aEW
  2 1.166390e-05 # Gf
  3 1.180000e-01 # aS
###
### INFORMATION FOR YUKAWA
###
Block yukawa
  5 4.200000e+00 # ymb
  6 1.735000e+02 # ymt
  15 1.777000e+00 # ymtau
```
C. MadGraph5_aMC@NLO cards

---

```plaintext
# INFORMATION FOR DECAY

# Dependent parameters, given by model restrictions. Those values should be edited following the analytical expression. MG5 ignores those values but they are important for interfacing the output of MG5 to external program such as Pythia.

DECAY 6 1.400000e+00 # WT
DECAY 23 2.495200e+00 # WZ
DECAY 24 2.141000e+00 # WW
DECAY 25 4.541000e-03 # WH
DECAY 1 0.000000 # d : 0.0
DECAY 2 0.000000 # u : 0.0
DECAY 3 0.000000 # s : 0.0
DECAY 4 0.000000 # c : 0.0
DECAY 5 0.000000 # b : 0.0
DECAY 11 0.000000 # e- : 0.0
DECAY 12 0.000000 # ve : 0.0
DECAY 13 0.000000 # mu- : 0.0
DECAY 14 0.000000 # vm : 0.0
DECAY 15 0.000000 # ta- : 0.0
DECAY 16 0.000000 # vt : 0.0
DECAY 21 0.000000 # g : 0.0
DECAY 22 0.000000 # a : 0.0
```

---

**Figure C.5**: MadGraph5_aMC@NLO run card

```plaintext
#******************************************************************************
# INFORMATION FOR DECAY
#******************************************************************************
DECAY 6 1.400000e+00 # WT
DECAY 23 2.495200e+00 # WZ
DECAY 24 2.141000e+00 # WW
DECAY 25 4.541000e-03 # WH
DECAY 1 0.000000 # d : 0.0
DECAY 2 0.000000 # u : 0.0
DECAY 3 0.000000 # s : 0.0
DECAY 4 0.000000 # c : 0.0
DECAY 5 0.000000 # b : 0.0
DECAY 11 0.000000 # e- : 0.0
DECAY 12 0.000000 # ve : 0.0
DECAY 13 0.000000 # mu- : 0.0
DECAY 14 0.000000 # vm : 0.0
DECAY 15 0.000000 # ta- : 0.0
DECAY 16 0.000000 # vt : 0.0
DECAY 21 0.000000 # g : 0.0
DECAY 22 0.000000 # a : 0.0
```

---

```
#******************************************************************************
# This file is used to set the parameters of the run.
#
# Some notation/conventions:
#
# Lines starting with a hash (#) are info or comments
#
# mind the format: value = variable ! comment
#
#******************************************************************************

# Running parameters

# Tag name for the run (one word)
tag_1 = run_tag ! name of the run

# Number of LHE events (and their normalization) and the required (relative) accuracy on the Xsec.
# These values are ignored for fixed order runs
2000000 = nevents ! Number of unweighted events requested
-1 = req_acc ! Required accuracy (-1=auto determined from nevents)
-1 = nevt_job! Max number of events per job in event generation. ! (-1= no split).
```
5. Appendix

# Normalize the weights of LHE events such that they sum or average to *
# the total cross section  *
#**************************************************************
# average = event_norm ! average or sum *
#**************************************************************
# Number of points per integration channel (ignored for aMC@NLO runs) *
#**************************************************************
0.01 = req_acc FO ! Required accuracy (-1=ignored, and use the *
# number of points and iter. below) *
# These numbers are ignored except if req_acc FO is equal to -1 *
5000 = npoints FO_grid ! number of points to setup grids *
4 = niter FO_grid ! number of iter. to setup grids *
10000 = npoints FO ! number of points to compute Xsec *
6 = niter FO ! number of iter. to compute Xsec *
#**************************************************************
# Random number seed *
#**************************************************************
0 = 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) *
4000 = ebeam1 ! beam 1 energy in GeV *
4000 = ebeam2 ! beam 2 energy in GeV *
#**************************************************************
# PDF choice: this automatically fixes also alpha_s (MZ) and its evol. *
#**************************************************************
! lhapdf = plabel ! PDF set *
260000 = lhaid ! if plabel=lhapdf, this is the lhapdf number *
#**************************************************************
# Include the NLO Monte Carlo subtr. terms for the following parton *
# shower (HERWIG6 | HERWIGPP | PYTHIA6Q | PYTHIA6PT | PYTHIA8) *
# WARNING: PYTHIA6PT works only for processes without FSR!!!! *
#**************************************************************
! HERWIGPP = parton_shower *
#**************************************************************
# Renormalization and factorization scales *
# (Default functional form for the non-fixed scales is the sum of *
# the transverse masses of all final state particles and partons. This *
# can be changed in SubProcesses/set_scales.f) *
#**************************************************************
.true. = fixed_ren_scale ! if .true., use fixed ren scale *
.true. = fixed_fac_scale ! if .true., use fixed fac scale *
235 = muR_ref_fixed ! fixed ren reference scale *
235 = muF1_ref_fixed ! fixed fact reference scale for pdf1 *
235 = muF2_ref_fixed ! fixed fact reference scale for pdf2 *
#**************************************************************
# Renormalization and factorization scales (advanced and NLO options) *
#**************************************************************
.true. = fixed_QES_scale ! if .true., use fixed Ellis-Sexton scale *
235 = QES_ref_fixed ! fixed Ellis-Sexton reference scale *
1 = muR_over_ref ! ratio of current muR over reference muR *
1 = muF1_over_ref ! ratio of current muF1 over reference muF1 *
1 = muF2_over_ref ! ratio of current muF2 over reference muF2 *
1 = QES_over_ref ! ratio of current QES over reference QES
# Reweight flags to get scale dependence and PDF uncertainty
# For scale dependence: factor rw_scale_up/down around central scale
# For PDF uncertainty: use LHAPDF with supported set
#
# ******************************************
# .true. = reweight_scale ! reweight to get scale dependence
# 0.5 = rw_Rscale_down ! lower bound for ren scale variations
# 2.0 = rw_Rscale_up  ! upper bound for ren scale variations
# 0.5 = rw_Fscale_down ! lower bound for fact scale variations
# 2.0 = rw_Fscale_up  ! upper bound for fact scale variations
# .false. = reweight_PDF ! reweight to get PDF uncertainty
# 260001 = PDF_set_min ! First of the error PDF sets
# 260100 = PDF_set_max ! Last of the error PDF sets
#
# ******************************************
#
# Merging - WARNING! Applies merging only at the hard–event level.
# After showering an MLM-type merging should be applied as well.
# See http://amcatnlo.cern.ch/FxFx_merging.htm for more details.
# ******************************************
#
# 0 = ickkw ! 0 no merging, 3 FxFx merging, 4 UNLOPS
#
# ******************************************
#
# BW cutoff (M+-/bwcutoff+Gamma)
# 15 = bwcutoff
#
# Cuts on the jets
#
# Jet clustering is performed by FastJet.
# When matching to a parton shower, these generation cuts should be
# considerably softer than the analysis cuts.
# (more specific cuts can be specified in SubProcesses/cuts.f)
# 1 = etagamma ! Max photon abs (pseudo-rap) (a value .lt.0 means no cut)
# 0.7 = jetradius ! The radius parameter for the jet algorithm
# 10 = ptj   ! Min jet transverse momentum
# -1 = etaj  ! Max jet abs (pseudo-rap) (a value .lt.0 means no cut)
#
# Cuts on the charged leptons (e+, e-, mu+, mu-, tau+ and tau-)
# (more specific gen cuts can be specified in SubProcesses/cuts.f)
# 0 = ptl    ! Min lepton transverse momentum
# -1 = etal   ! Max lepton abs (pseudo-rap) (a value .lt.0 means no cut)
# 0 = drll    ! Min distance between opposite sign lepton pairs
# 0 = drll_sf ! Min distance between opp. sign same–flavor lepton pairs
# 0 = mll    ! Min inv. mass of all opposite sign lepton pairs
# 30 = mll_sf ! Min inv. mass of all opp. sign same–flavor lepton pairs
#
# Photon–isolation cuts, according to hep–ph/9801442
# When ptgmin=0, all the other parameters are ignored
# 20 = ptgmin ! Min photon transverse momentum
# -1 = etagamma ! Max photon abs (pseudo–rap)
# 0.4 = R0gamma ! Radius of isolation code
# 1.0 = xn  ! n parameter of eq.(3.4) in hep–ph/9801442
# 1.0 = epsgamma ! epsilon–gamma parameter of eq.(3.4) in hep–ph/9801442
# .true. = isoEM ! isolate photons from EM energy (photons and leptons)
#
# Maximal PDG code for quark to be considered a jet when applying cuts.
# At least all massless quarks of the model should be included here.
# 5 = maxjetflavor
#
# For aMCfast+APPLGRID use in PDF fitting (http://amcfast.hepforge.org)
# 0 = iappl ! aMCfast switch (0=OFF, 1=prepare APPPlgrids, 2=fill grids)
#
# ******************************************
D. Rivet analyses

Figure D.6.: Fixed order analysis for $t\bar{t}$ samples based on a version written by Daniel Rauch

```cpp
// --- C++ ---
#include "Rivet/Analysis.hh"
#include "Rivet/RivetAIDA.hh"
#include "Rivet/Tools/Logging.hh"
#include "Rivet/Projections/FSFinalState.hh"
#include "Rivet/Projections/FastJets.hh"
#include "Rivet/Projections/VetoedFinalState.hh"
#include "Rivet/Projections/IdentifiedFinalState.hh"

namespace Rivet {

    class TTBAR : public Analysis {

    public:

        /// Name constructors etc.
        /// Constructor
        TTBAR() : Analysis("TTBAR") {

        }

    public:

        /// Book histograms and initialise projections before the run
        void init() {

            // Initialise and register projections here
            const FinalState fs;
            addProjection(fs, "FS");
            IdentifiedFinalState top_fs(fs);
            top_fs.acceptIdPair(6);
            addProjection(top_fs, "TOP_FS");

            VetoedFinalState vetoed_fs;
            vetoed_fs.addVetoPairId(6);
            addProjection(FastJets(vetoed_fs, FastJets::ANTIKT, 0.6), "JET_FS");

            // Book histograms here

            _h_t_m = bookHistogram1D("t_m", 50, 150.0, 200.0);
            _h_t_pT = bookHistogram1D("t_pT", 50, 0.0, 1000.0);
            _h_t_pT_low = bookHistogram1D("t_pT_low", 50, 0.0, 100.0);
            _h_t_y = bookHistogram1D("t_y", 50, -5.0, +5.0);
            _h_t_phi = bookHistogram1D("t_phi", 50, 0.0, 6.28);

            _h_tbar_m = bookHistogram1D("tbar_m", 50, 150.0, 200.0);
            _h_tbar_pT = bookHistogram1D("tbar_pT", 50, 0.0, 1000.0);
            _h_tbar_pT_low = bookHistogram1D("tbar_pT_low", 50, 0.0, 100.0);
            _h_tbar_y = bookHistogram1D("tbar_y", 50, -5.0, +5.0);
            _h_tbar_phi = bookHistogram1D("tbar_phi", 50, 0.0, 6.28);
        }
    }
}
```
D. Rivet analyses

```cpp
    _h_ttb_m = bookHistogram1D("ttb_m", 50, 300.0, 400.0);
    _h_ttb_pT = bookHistogram1D("ttb_pT", 50, 0.0, 1000.0);
    _h_ttb_pT_low = bookHistogram1D("ttb_pT_low", 50, 0.0, 100.0);
    _h_ttb_y = bookHistogram1D("ttb_y", 50, -5.0, +5.0);
    _h_ttb_phi = bookHistogram1D("ttb_phi", 50, 0.0, 6.28);

    _h_nj = bookHistogram1D("nj", 20, 0, 20);
    _h_j1_pT = bookHistogram1D("j1_pT", 50, 0.0, 1000.0);
    _h_j1_pT_low = bookHistogram1D("j1_pT_low", 50, 0.0, 100.0);
    _h_j1_y = bookHistogram1D("j1_y", 50, -6.0, +6.0);
    _h_j1_phi = bookHistogram1D("j1_phi", 50, 0.0, +6.28);
    _h_j1_m = bookHistogram1D("j1_m", 50, 0.0, 200.0);
    _h_j1_E = bookHistogram1D("j1_E", 50, 0.0, 1000.0);

    _h_j2_pT = bookHistogram1D("j2_pT", 50, 0.0, 1000.0);
    _h_j2_pT_low = bookHistogram1D("j2_pT_low", 50, 0.0, 100.0);
    _h_j2_y = bookHistogram1D("j2_y", 50, -6.0, +6.0);
    _h_j2_phi = bookHistogram1D("j2_phi", 50, 0.0, +6.28);
    _h_j2_m = bookHistogram1D("j2_m", 50, 0.0, 200.0);
    _h_j2_E = bookHistogram1D("j2_E", 50, 0.0, 1000.0);

    _h_j3_pT = bookHistogram1D("j3_pT", 50, 0.0, 1000.0);
    _h_j3_pT_low = bookHistogram1D("j3_pT_low", 50, 0.0, 100.0);
    _h_j3_y = bookHistogram1D("j3_y", 50, -6.0, +6.0);
    _h_j3_phi = bookHistogram1D("j3_phi", 50, 0.0, +6.28);
    _h_j3_m = bookHistogram1D("j3_m", 50, 0.0, 200.0);
    _h_j3_E = bookHistogram1D("j3_E", 50, 0.0, 1000.0);

    // Book normalized histograms

    _h_t_m_norm = bookHistogram1D("t_m_norm", 50, 150.0, 200.0);
    _h_t_pT_norm = bookHistogram1D("t_pT_norm", 50, 0.0, 1000.0);
    _h_t_pT_low_norm = bookHistogram1D("t_pT_low_norm", 50, 0.0, 100.0);
    _h_t_y_norm = bookHistogram1D("t_y_norm", 50, -5.0, +5.0);
    _h_t_phi_norm = bookHistogram1D("t_phi_norm", 50, 0.0, 6.28);

    _h_tbar_m_norm = bookHistogram1D("tbar_m_norm", 50, 150.0, 200.0);
    _h_tbar_pT_norm = bookHistogram1D("tbar_pT_norm", 50, 0.0, 1000.0);
    _h_tbar_pT_low_norm = bookHistogram1D("tbar_pT_low_norm", 50, 0.0, 100.0);
    _h_tbar_y_norm = bookHistogram1D("tbar_y_norm", 50, -5.0, +5.0);
    _h_tbar_phi_norm = bookHistogram1D("tbar_phi_norm", 50, 0.0, 6.28);

    _h_ttb_m_norm = bookHistogram1D("ttb_m_norm", 50, 300.0, 400.0);
    _h_ttb_pT_norm = bookHistogram1D("ttb_pT_norm", 50, 0.0, 1000.0);
    _h_ttb_pT_low_norm = bookHistogram1D("ttb_pT_low_norm", 50, 0.0, 100.0);
    _h_ttb_y_norm = bookHistogram1D("ttb_y_norm", 50, -5.0, +5.0);
    _h_ttb_phi_norm = bookHistogram1D("ttb_phi_norm", 50, 0.0, 6.28);

    _h_nj_norm = bookHistogram1D("nj_norm", 20, 0, 20);
    _h_j1_pT_norm = bookHistogram1D("j1_pT_norm", 50, 0.0, 1000.0);
    _h_j1_pT_low_norm = bookHistogram1D("j1_pT_low_norm", 50, 0.0, 100.0);
    _h_j1_y_norm = bookHistogram1D("j1_y_norm", 50, -6.0, +6.0);
    _h_j1_phi_norm = bookHistogram1D("j1_phi_norm", 50, 0.0, +6.28);
    _h_j1_m_norm = bookHistogram1D("j1_m_norm", 50, 0.0, 200.0);
```
/// Perform the per-event analysis
void analyze(const Event& event) {

const double JetPTMin = 0.0;
const double JetPRapMin = -100.0;
const double JetPRapMax = 100.0;
const double weight = event.weight();

/// @todo Do the event by event analysis here

// const FinalState& fs = applyProjection<FinalState>(event, "FS");
const IdentifiedFinalState& top_fs = applyProjection<
    IdentifiedFinalState>(event, "TOP_FS");
const Jets jets = applyProjection<FastJets>(event, "JET_FS").jetsByPt
    (JetPTMin*GeV, MAXDOUBLE, JetPRapMin, JetPRapMax);

// only analyze if there is exactly a top–antitop pair in the event
if (top_fs.particles().size() != 2) return;

int it = -1;
int itb = -1;
if (top_fs.particles()[0].pdgId() == 6 && top_fs.particles()[1].pdgId
    () == -6){
    it = 0;
itb = 1;
} else if (top_fs.particles()[0].pdgId() == -6 && top_fs.particles()
    [1].pdgId() == 6){
    it = 1;
itb = 0;
} else {
    cerr << "Somehow could not find a top and antitop pair in the event
         !!!\n" << std::flush;
    cerr << " top_fs.particles()[0].pdgId() = " << top_fs.particles()
         [0].pdgId() << "\n";
    cerr << " top_fs.particles()[1].pdgId() = " << top_fs.particles()
         [1].pdgId() << "\n";
    return;
}

// fill separate top and antitop histograms
_h_t_m->fill(top_fs.particles()[it].mass(), weight);
_h_t_pT->fill(top_fs.particles()[it].momentum().pT(), weight);
_h_t_pT_low->fill(top_fs.particles()[it].momentum().pT(), weight);
_h_t_y->fill(top_fs.particles()[it].momentum().rapidity(), weight);
```cpp
    _h_t_phi->fill(top_fs.particles()[it].momentum().phi(), weight);
    _h_t_m->fill(top_fs.particles()[it].mass(), weight);
    _h_t_pT->fill(top_fs.particles()[it].momentum().pT(), weight);
    _h_t_pT_low->fill(top_fs.particles()[it].momentum().pT(), weight);
    _h_t_y->fill(top_fs.particles()[it].momentum().rapidity(), weight);

    _h_t_pT_norm->fill(top_fs.particles()[it].momentum().pT(), weight);
    _h_t_m_norm->fill(top_fs.particles()[it].mass(), weight);
    _h_t_y_norm->fill(top_fs.particles()[it].momentum().rapidity(), weight);
    _h_t_phi_norm->fill(top_fs.particles()[it].momentum().phi(), weight);

    _h_tbar_m->fill(top_fs.particles()[itb].mass(), weight);
    _h_tbar_pT->fill(top_fs.particles()[itb].momentum().pT(), weight);
    _h_tbar_pT_low->fill(top_fs.particles()[itb].momentum().pT(), weight);
    _h_tbar_y->fill(top_fs.particles()[itb].momentum().rapidity(), weight);
    _h_tbar_phi->fill(top_fs.particles()[itb].momentum().phi(), weight);
    _h_tbar_m_norm->fill(top_fs.particles()[itb].mass(), weight);
    _h_tbar_pT_norm->fill(top_fs.particles()[itb].momentum().pT(), weight);
    _h_tbar_y_norm->fill(top_fs.particles()[itb].momentum().rapidity(), weight);
    _h_tbar_phi_norm->fill(top_fs.particles()[itb].momentum().phi(), weight);

    // fill histograms with combined top and antitop
    FourMomentum pttb = top_fs.particles()[it].momentum() + top_fs.particles()[itb].momentum();
    _h_ttb_m->fill(pttb.mass(), weight);
    _h_ttb_pT->fill(pttb.pT(), weight);
    _h_ttb_pT_low->fill(pttb.pT(), weight);
    _h_ttb_y->fill(pttb.rapidity(), weight);
    _h_ttb_phi->fill(pttb.phi(), weight);
    _h_ttb_m_norm->fill(pttb.mass(), weight);
    _h_ttb_pT_norm->fill(pttb.pT(), weight);
    _h_ttb_y_norm->fill(pttb.rapidity(), weight);
    _h_ttb_phi_norm->fill(pttb.phi(), weight);

    // fill jet histograms
    _h_nj->fill(jets.size(), weight);
    _h_nj_norm->fill(jets.size(), weight);
    if (jets.size() > 0){
      _h_j1_pT->fill(jets[0].momentum().pT(), weight);
      _h_j1_pT_low->fill(jets[0].momentum().pT(), weight);
      _h_j1_y->fill(jets[0].momentum().rapidity(), weight);
      _h_j1_phi->fill(jets[0].momentum().phi(), weight);
      _h_j1_m->fill(jets[0].momentum().mass(), weight);
      _h_j1_E->fill(jets[0].momentum().E(), weight);
      _h_j1_pT_norm->fill(jets[0].momentum().pT(), weight);
      _h_j1_y_norm->fill(jets[0].momentum().rapidity(), weight);
      _h_j1_phi_norm->fill(jets[0].momentum().phi(), weight);
      _h_j1_m_norm->fill(jets[0].momentum().mass(), weight);
      _h_j1_E_norm->fill(jets[0].momentum().E(), weight);
    }
    if (jets.size() > 1){
      _h_j2_pT->fill(jets[1].momentum().pT(), weight);
      _h_j2_pT_low->fill(jets[1].momentum().pT(), weight);
      _h_j2_y->fill(jets[1].momentum().rapidity(), weight);
      _h_j2_phi->fill(jets[1].momentum().phi(), weight);
```
/// Normalize histograms etc., after the run

void finalize() {

    // Todo Normalize, scale and otherwise manipulate histograms here
    scale(_h_j2_m, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section
    scale(_h_j2_E, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section
    scale(_h_j2_pT, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section
    scale(_h_j2_pT_low, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section
    scale(_h_j2_y, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section
    scale(_h_j2_phi, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section
    scale(_h_j2_m_norm, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section
    scale(_h_j2_E_norm, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section
    scale(_h_j2_m, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section
    scale(_h_j2_pT, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section
    scale(_h_j2_pT_low, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section
    scale(_h_j2_y, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section
    scale(_h_j2_phi, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section
    scale(_h_j2_m_norm, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section
    scale(_h_j2_E_norm, 1000.0*crossSection() / sumOfWeights()); // normalize to cross section

    _h_j2_m->fill(jets[1].momentum().mass(), weight);
    _h_j2_E->fill(jets[1].momentum().E(), weight);
    _h_j2_pT_norm->fill(jets[1].momentum().pT(), weight);
    _h_j2_y_norm->fill(jets[1].momentum().rapidity(), weight);
    _h_j2_phi_norm->fill(jets[1].momentum().phi(), weight);
    _h_j2_m_norm->fill(jets[1].momentum().mass(), weight);
    _h_j2_E_norm->fill(jets[1].momentum().E(), weight);
}

if (jets.size() > 2) {
    _h_j3_pT->fill(jets[2].momentum().pT(), weight);
    _h_j3_pT_low->fill(jets[2].momentum().pT(), weight);
    _h_j3_y->fill(jets[2].momentum().rapidity(), weight);
    _h_j3_phi->fill(jets[2].momentum().phi(), weight);
    _h_j3_m->fill(jets[2].momentum().mass(), weight);
    _h_j3_E->fill(jets[2].momentum().E(), weight);
    _h_j3_pT_norm->fill(jets[2].momentum().pT(), weight);
    _h_j3_pT_low_norm->fill(jets[2].momentum().pT(), weight);
    _h_j3_y_norm->fill(jets[2].momentum().rapidity(), weight);
    _h_j3_phi_norm->fill(jets[2].momentum().phi(), weight);
    _h_j3_m_norm->fill(jets[2].momentum().mass(), weight);
    _h_j3_E_norm->fill(jets[2].momentum().E(), weight);
}
scale(_h_nj, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j1_pT, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j1_pT_low, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j1_y, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j1_phi, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j1_m, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j1_E, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j2_pT, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j2_pT_low, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j2_y, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j2_phi, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j2_m, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j2_E, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j3_pT, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j3_pT_low, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j3_y, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j3_phi, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j3_m, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section
scale(_h_j3_E, 1000.0*crossSection()/sumOfWeights()); //
  normalize to cross section

  // normalize(_h_YYYY); # normalize to unity
normalize(_h_t-m_norm, 1.0); // normalize to unity
normalize(_h_t-pT_norm, 1.0); // normalize to unity
normalize(_h_t-pT_low_norm, 1.0); // normalize to unity
normalize(_h_t-y_norm, 1.0); // normalize to unity
normalize(_h-t_phi_norm, 1.0); // normalize to unity
normalize(_h-tbar-m_norm, 1.0); // normalize to unity
normalize(_h-tbar-pT_norm, 1.0); // normalize to unity
normalize(_h-tbar-pT_low_norm, 1.0); // normalize to unity
normalize(_h-tbar-y_norm, 1.0); // normalize to unity
normalize(_h-tbar_phi_norm, 1.0); // normalize to unity
normalize(_h-ttb-m_norm, 1.0); // normalize to unity
normalize(_h-ttb-pT_norm, 1.0); // normalize to unity
normalize(_h-ttb-pT_low_norm, 1.0); // normalize to unity
normalize(_h-ttb-y_norm, 1.0); // normalize to unity
normalize(_h-ttb_phi_norm, 1.0); // normalize to unity
normalize(_h-nj_norm, 1.0); // normalize to unity
normalize(_h-j1-pT_norm, 1.0); // normalize to unity
normalize(_h-j1_pT_low_norm, 1.0); // normalize to unity
normalize(_h-j1-y_norm, 1.0); // normalize to unity
normalize(_h-j1_phi_norm, 1.0); // normalize to unity
normalize(_h-j1_m_norm, 1.0); // normalize to unity
normalize(h_j1_E_norm, 1.0); // normalize to unity
normalize(h_j2_pT_norm, 1.0); // normalize to unity
normalize(h_j2_pT_low_norm, 1.0); // normalize to unity
normalize(h_j2_y_norm, 1.0); // normalize to unity
normalize(h_j2_phi_norm, 1.0); // normalize to unity
normalize(h_j2_m_norm, 1.0); // normalize to unity
normalize(h_j2_E_norm, 1.0); // normalize to unity
normalize(h_j3_pT_norm, 1.0); // normalize to unity
normalize(h_j3_pT_low_norm, 1.0); // normalize to unity
normalize(h_j3_y_norm, 1.0); // normalize to unity
normalize(h_j3_phi_norm, 1.0); // normalize to unity
normalize(h_j3_m_norm, 1.0); // normalize to unity
normalize(h_j3_E_norm, 1.0); // normalize to unity

private:
// Data members like post-cuts event weight counters go here

private:
/// @name Histograms
//@
{
// regular histograms
aida::ihistogram1d *h_t_m, *h_t_pT, *h_t_pT_low, *h_t_y, *h_t_phi;
aida::ihistogram1d *h_tbar_m, *h_tbar_pT, *h_tbar_pT_low, *h_tbar_y , *h_tbar_phi;
aida::ihistogram1d *h_ttb_m, *h_ttb_pT, *h_ttb_pT_low, *h_ttb_y , *h_ttb_phi;
aida::ihistogram1d *h_nj;
aida::ihistogram1d *h_j1_pT, *h_j1_pT_low, *h_j1_y, *h_j1_phi, * h_j1_m, *h_j1_E;
aida::ihistogram1d *h_j2_pT, *h_j2_pT_low, *h_j2_y, *h_j2_phi, * h_j2_m, *h_j2_E;
aida::ihistogram1d *h_j3_pT, *h_j3_pT_low, *h_j3_y, *h_j3_phi, * h_j3_m, *h_j3_E;
//@
}
DECLARE_RIVET_PLUGIN(TTBAR_STABLE);

}
D. Rivet analyses

Figure D.7.: Fixed order analysis for $t\bar{t}H$ samples based on a version written by Daniel Rauch

```cpp
#include "Rivet/Analysis.hh"
#include "Rivet/RivetAIDA.hh"
#include "Rivet/Tools/Logging.hh"
#include "Rivet/Projections/FinalState.hh"
/// @todo Include more projections as required, e.g. ChargedFinalState, FastJets, ZFinder...

namespace Rivet {

  class TTH_STABLE_FO : public Analysis {
    public:
      /// @name Constructors etc.
      //@
      /// Constructor
      TTH_STABLE_FO() : Analysis("TTH_STABLE_FO")
      {
      }
      //@

      public:
      /// @name Analysis methods
      //@
      /// Book histograms and initialise projections before the run
      void init() {
        /// @todo Initialise and register projections here
        const FinalState fs;
        addProjection(fs, "FS");

        /// @todo Book histograms here, e.g.:
        /// book regular histograms
        _h_t_pT = bookHistogram1D("t_pT", 25, 0.0, 1000.0);
        _h_t_pT_low = bookHistogram1D("t_pT_low", 12, 0.0, 100.0);
        _h_t_y = bookHistogram1D("t_y", 50, -5.0, +5.0);
        _h_t_theta = bookHistogram1D("t_theta", 25, 0.0, 3.15);
        _h_t_phi = bookHistogram1D("t_phi", 25, 0.0, 6.28);
        _h_t_mass = bookHistogram1D("t_mass", 50, 160, 200);

        _h_tbar_pT = bookHistogram1D("tbar_pT", 25, 0.0, 1000.0);
        _h_tbar_pT_low = bookHistogram1D("tbar_pT_low", 12, 0.0, 100.0);
        _h_tbar_y = bookHistogram1D("tbar_y", 50, -5.0, +5.0);
        _h_tbar_theta = bookHistogram1D("tbar_theta", 25, 0.0, 3.15);
        _h_tbar_phi = bookHistogram1D("tbar_phi", 25, 0.0, 6.28);
        _h_tbar_mass = bookHistogram1D("tbar_mass", 50, 160, 200);

        _h_h0_pT = bookHistogram1D("h0_pT", 25, 0.0, 1000.0);
        _h_h0_pT_low = bookHistogram1D("h0_pT_low", 12, 0.0, 100.0);
        _h_h0_y = bookHistogram1D("h0_y", 50, -5.0, +5.0);
        _h_h0_theta = bookHistogram1D("h0_theta", 25, 0.0, 3.15);
        _h_h0_phi = bookHistogram1D("h0_phi", 25, 0.0, 6.28);
        _h_h0_mass = bookHistogram1D("h0_mass", 50, 110, 140);
      }

  };

} // namespace Rivet
```
5. Appendix

// book regular histograms
_h_tth_pT = bookHistogram1D("tth_pT", 25, 0.0, 1000.0);
_h_tth_pT_low = bookHistogram1D("tth_pT_low", 12, 0.0, 1000.0);
_h_tth_y = bookHistogram1D("tth_y", 50, -5.0, +5.0);
_h_tth_theta = bookHistogram1D("tth_theta", 25, 0.0, 3.15);
_h_tth_phi = bookHistogram1D("tth_phi", 25, 0.0, 6.28);
_h_tth_mass = bookHistogram1D("tth_mass", 50, 0.0, 7000);

_h_j_pT = bookHistogram1D("j_pT", 25, 0.0, 1000.0);
_h_j_pT_low = bookHistogram1D("j_pT_low", 12, 0.0, 100.0);
_h_j_y = bookHistogram1D("j_y", 50, -5.0, +5.0);
_h_j_theta = bookHistogram1D("j_theta", 25, 0.0, 3.15);
_h_j_phi = bookHistogram1D("j_phi", 25, 0.0, 6.28);
_h_j_mass = bookHistogram1D("j_mass", 25, 0.0, 500);

_h_weights = bookHistogram1D("weights", 50, -1.0, 1.0);
_h_weights_2bin = bookHistogram1D("weights_2bin", 2, -1.0, 1.0);
_h_weights_small = bookHistogram1D("weights_small", 50, -0.2, 0.2);

// book regular histograms
_h_t_pT_norm = bookHistogram1D("t_pT_norm", 25, 0.0, 1000.0);
_h_t_pT_low_norm = bookHistogram1D("t_pT_low_norm", 12, 0.0, 100.0);
_h_t_y_norm = bookHistogram1D("t_y_norm", 50, -5.0, +5.0);
_h_t_theta_norm = bookHistogram1D("t_theta_norm", 25, 0.0, 3.15);
_h_t_phi_norm = bookHistogram1D("t_phi_norm", 25, 0.0, 6.28);
_h_t_mass_norm = bookHistogram1D("t_mass_norm", 50, 0, 7000);

_h_tbar_pT_norm = bookHistogram1D("tbar_pT_norm", 25, 0.0, 1000.0);
_h_tbar_pT_low_norm = bookHistogram1D("tbar_pT_low_norm", 12, 0.0, 100.0);
_h_tbar_y_norm = bookHistogram1D("tbar_y_norm", 25, 0.0, 3.15);
_h_tbar_theta_norm = bookHistogram1D("tbar_theta_norm", 25, 0.0, 6.28);
_h_tbar_mass_norm = bookHistogram1D("tbar_mass_norm", 50, 160, 200);

_h_h0_pT_norm = bookHistogram1D("h0_pT_norm", 25, 0.0, 1000.0);
_h_h0_pT_low_norm = bookHistogram1D("h0_pT_low_norm", 12, 0.0, 100.0);
_h_h0_y_norm = bookHistogram1D("h0_y_norm", 25, 0.0, 3.15);
_h_h0_theta_norm = bookHistogram1D("h0_theta_norm", 25, 0.0, 6.28);
_h_h0_mass_norm = bookHistogram1D("h0_mass_norm", 25, 0.0, 6.28);
_h_h0_mass_norm = bookHistogram1D("h0_mass_norm", 50, 0, 7000);
D. Rivet analyses

112 \_h\_j\_pT\_norm = bookHistogram1D("\_j\_pT\_norm", 25, 0.0, 1000.0);
113 \_h\_j\_pT\_low\_norm = bookHistogram1D("\_j\_pT\_low\_norm", 12, 0.0, 100.0);
114 \_h\_j\_y\_norm = bookHistogram1D("\_j\_y\_norm", 50, -5.0, +5.0);
115 \_h\_j\_theta\_norm = bookHistogram1D("\_j\_theta\_norm", 25, 0.0, 3.15);
116 \_h\_j\_phi\_norm = bookHistogram1D("\_j\_phi\_norm", 25, 0.0, 6.28);
117 \_h\_j\_mass\_norm = bookHistogram1D("\_j\_mass\_norm", 25, 0.0, 500);
119 \_h\_weights\_norm = bookHistogram1D("\_weights\_norm", 50, -1.0, 1.0);
120 \_h\_weights\_2bin\_norm = bookHistogram1D("\_weights\_2bin\_norm", 2, -1.0, 1.0);
121 \_h\_weights\_small\_norm = bookHistogram1D("\_weights\_small\_norm", 50, -0.2, 0.2);
122 }

125 /// Perform the per-event analysis
126 void analyze(const Event& event) {
128 const double JetPTMin = 10.0;
130 const double weight = event.weight();
132 /// @todo Do the event by event analysis here
133 const FinalState& fs = applyProjection<FinalState>(event, "FS");
135 int it = -1;
136 int itb = -1;
137 int ih0 = -1;
138 int ij = -1;
140 for (int i = 0; i < (int)fs.particles().size(); i++){
141 if (fs.particles()[i].pdgId() == 6) it = i;
142 if (fs.particles()[i].pdgId() == -6) itb = i;
143 if (fs.particles()[i].pdgId() == 25) ih0 = i;
144 if (fs.particles()[i].pdgId() == 21 || abs(fs.particles()[i].pdgId()) < 6) ij = i;
147 FourMomentum t = fs.particles()[it].momentum();
148 FourMomentum tb = fs.particles()[itb].momentum();
149 FourMomentum h0 = fs.particles()[ih0].momentum();
151 FourMomentum tth = t + tb + h0;
153 if (it != -1 && itb != -1 && ih0 != -1) {
155 \_h\_t\_pT\_norm->fill(t.pT(), weight);
156 \_h\_t\_pT\_low\_norm->fill(t.pT(), weight);
157 \_h\_t\_y\_norm->fill(t.rapidity(), weight);
158 \_h\_t\_theta\_norm->fill(t.theta(), weight);
159 \_h\_t\_phi\_norm->fill(t.phi(), weight);
160 \_h\_t\_mass\_norm->fill(t.mass(), weight);
162 \_h\_t\_bar\_pT\_norm->fill(tb.pT(), weight);
163 \_h\_t\_bar\_pT\_low\_norm->fill(tb.pT(), weight);
164 \_h\_t\_bar\_y\_norm->fill(tb.rapidity(), weight);
165 \_h\_t\_bar\_theta\_norm->fill(tb.theta(), weight);
166 \_h\_t\_bar\_phi\_norm->fill(tb.phi(), weight);
167 \_h\_t\_bar\_mass\_norm->fill(tb.mass(), weight);
if (ij != -1) {
  if (fs.particles()[ij].momentum().pT() >= JetPTMin) {
    _h_j_pT->fill(fs.particles()[ij].momentum().pT(), weight);
    _h_j_pT_low->fill(fs.particles()[ij].momentum().pT(), weight);
    _h_j_y->fill(fs.particles()[ij].momentum().rapidity(), weight);
    _h_j_theta->fill(fs.particles()[ij].momentum().theta(), weight);
    _h_j_phi->fill(fs.particles()[ij].momentum().phi(), weight);
    _h_j_mass->fill(fs.particles()[ij].momentum().mass(), weight);
  }
}

_weightsephirightarrowfill(weight, 1);
_weights2bin_rightarrowfill(weight, 1);
_weights_small_rightarrowfill(weight, 1);

_weights_norm_rightarrowfill(weight, 1);
_weights2bin_norm_rightarrowfill(weight, 1);
_weights_small_norm_rightarrowfill(weight, 1);
D. Rivet analyses

231    _h_j_pT_norm->fill(fs.particles()[ij].momentum().pT(), weight);
232    _h_j_pT_low_norm->fill(fs.particles()[ij].momentum().pT(), weight);
233    _h_j_y_norm->fill(fs.particles()[ij].momentum().rapidity(), weight);
234    _h_j_theta_norm->fill(fs.particles()[ij].momentum().theta(), weight);
235    _h_j_phi_norm->fill(fs.particles()[ij].momentum().phi(), weight);
236    _h_j_mass_norm->fill(fs.particles()[ij].momentum().mass(), weight);
237 }
238 }
239 }

242 /// Normalise histograms etc., after the run
243 void finalize() {
244     /// @todo Normalise, scale and otherwise manipulate histograms here
245     scale(_h_t_pT, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
246     scale(_h_t_pT_low, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
247     scale(_h_t_y, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
248     scale(_h_t_theta, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
249     scale(_h_t_phi, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
250     scale(_h_t_mass, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
251     scale(_h_tbar_pT, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
252     scale(_h_tbar_pT_low, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
253     scale(_h_tbar_y, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
254     scale(_h_tbar_theta, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
255     scale(_h_tbar_phi, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
256     scale(_h_tbar_mass, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
257     scale(_h_h0_pT, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
258     scale(_h_h0_pT_low, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
259     scale(_h_h0_y, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
260     scale(_h_h0_theta, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
261     scale(_h_h0_phi, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
262     scale(_h_h0_mass, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
263     scale(_h_tth_pT, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
264     scale(_h_tth_pT_low, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
265     scale(_h_tth_y, 1000.0*crossSection()/sumOfWeights()); // normalize to cross section
scale(_h_{tth_theta}, 1000.0*crossSection()/sumOfWeights());  //
normalize to cross section

scale(_h_{tth_phi}, 1000.0*crossSection()/sumOfWeights());  //
normalize to cross section

scale(_h_{tth_mass}, 1000.0*crossSection()/sumOfWeights());  //
normalize to cross section

scale(_h_{j}_pT, 1000.0*crossSection()/sumOfWeights());  //
normalize to cross section

scale(_h_{j}_mass, 1000.0*crossSection()/sumOfWeights());  //
normalize to cross section

scale(_h_{tth_pT}, 1000.0*crossSection()/sumOfWeights());  //
normalize to cross section

scale(_h_{j}_y, 1000.0*crossSection()/sumOfWeights());  //
normalize to cross section

scale(_h_{j}_theta, 1000.0*crossSection()/sumOfWeights());  //
normalize to cross section

scale(_h_{j}_phi, 1000.0*crossSection()/sumOfWeights());  //
normalize to cross section

scale(_h_{j}_pT_low, 1000.0*crossSection()/sumOfWeights());  //
normalize to cross section

scale(_h_{j}_y_low, 1000.0*crossSection()/sumOfWeights());  //
normalize to cross section

scale(_h_{j}_theta_low, 1000.0*crossSection()/sumOfWeights());  //
normalize to cross section

scale(_h_{j}_phi_low, 1000.0*crossSection()/sumOfWeights());  //
normalize to cross section

scale(_h_{j}_mass_low, 1000.0*crossSection()/sumOfWeights());  //
normalize to cross section

// normalize(_h_{YYYY}); # normalize to unity
normalize(_h_{tth_pT}norm, 1.0);  // normalize to unity
normalize(_h_{tth_pT_low}norm, 1.0);  // normalize to unity
normalize(_h_{tth_y}norm, 1.0);  // normalize to unity
normalize(_h_{tth_theta}norm, 1.0);  // normalize to unity
normalize(_h_{tth_phi}norm, 1.0);  // normalize to unity
normalize(_h_{tth_mass}norm, 1.0);  // normalize to unity

normalize(_h_{j}_pT_norm, 1.0);  // normalize to unity
normalize(_h_{j}_pT_low_norm, 1.0);  // normalize to unity
normalize(_h_{j}_y_norm, 1.0);  // normalize to unity
normalize(_h_{j}_theta_norm, 1.0);  // normalize to unity
normalize(_h_{j}_phi_norm, 1.0);  // normalize to unity
normalize(_h_{j}_mass_norm, 1.0);  // normalize to unity

normalize(_h_{tth_pT}norm, 1.0);  // normalize to unity
normalize(_h_{tth_pT_low}norm, 1.0);  // normalize to unity
normalize(_h_{tth_y}norm, 1.0);  // normalize to unity
normalize(_h_{tth_theta}norm, 1.0);  // normalize to unity
normalize(_h_{tth_phi}norm, 1.0);  // normalize to unity
normalize(_h_{tth_mass}norm, 1.0);  // normalize to unity

normalize(_h_{j}_pT_norm, 1.0);  // normalize to unity
normalize(_h_{j}_pT_low_norm, 1.0);  // normalize to unity
normalize(_h_{j}_y_norm, 1.0);  // normalize to unity
normalize(_h_{j}_theta_norm, 1.0);  // normalize to unity
normalize(_h_{j}_phi_norm, 1.0);  // normalize to unity
normalize(_h_{j}_mass_norm, 1.0);  // normalize to unity

normalize(_h_{weights}norm, 1.0);  // normalize to unity
normalize(_h_{weights}_2bin_norm, 1.0);  // normalize to unity
normalize(_h_{weights}_small_norm, 1.0);  // normalize to unity
}

//@}
private:

// Data members like post-cuts event weight counters go here

private:

/// @name Histograms
//@
{
// regular histograms
AIDA::IHistogram1D *h_t_pT, *h_t_pT_low, *h_t_y, *h_t_theta, *
    *h_t_phi, *h_t_mass;
AIDA::IHistogram1D *h_tbar_pT, *h_tbar_pT_low, *h_tbar_y, *
    *h_tbar_theta, *h_tbar_phi, *h_tbar_mass;
AIDA::IHistogram1D *h_h0_pT, *h_h0_pT_low, *h_h0_y, *h_h0_theta, *
    *h_h0_phi, *h_h0_mass;
AIDA::IHistogram1D *h_tth_pT, *h_tth_pT_low, *h_tth_y, *h_tth_theta
    , *h_tth_phi, *h_tth_mass;
AIDA::IHistogram1D *h_j_pT, *h_j_pT_low, *h_j_y, *h_j_theta, *
    *h_j_phi, *h_j_mass;
AIDA::IHistogram1D *h_weights, *h_weights_small, *h_weights_2bin;

// normalized histograms
AIDA::IHistogram1D *h_t_pT_norm, *h_t_pT_low_norm, *h_t_y_norm, *
    *h_t_theta_norm, *h_t_phi_norm, *h_t_mass_norm;
AIDA::IHistogram1D *h_tbar_pT_norm, *h_tbar_pT_low_norm, *h_tbar_y_norm
    , *h_tbar_theta_norm, *h_tbar_phi_norm, *
    *h_tbar_mass_norm;
AIDA::IHistogram1D *h_h0_pT_norm, *h_h0_pT_low_norm, *h_h0_y_norm, *
    *h_h0_theta_norm, *h_h0_phi_norm, *h_h0_mass_norm;
AIDA::IHistogram1D *h_tth_pT_norm, *h_tth_pT_low_norm, *h_tth_y_norm
    , *h_tth_theta_norm, *h_tth_phi_norm, *h_tth_mass_norm;
AIDA::IHistogram1D *h_j_pT_norm, *h_j_pT_low_norm, *h_j_y_norm, *
    *h_j_theta_norm, *h_j_phi_norm, *h_j_mass_norm;
AIDA::IHistogram1D *h_weights_norm, *h_weights_small_norm, *
    *h_weights_2bin_norm;
//@
}

// The hook for the plugin system
DECLARE_RIVET_PLUGIN(TTH_STABLE_FO);
}
Appendix

Figure D.8: Particle level Rivet analysis for $t\bar{t}H$ samples. Best performance is achieved on samples with restricted Higgs decay $H \rightarrow \mu^+\mu^-$, as this is the channel on which the analysis operates, and the branching ratio is of order $O(10^{-3})$. Based on the Rivet analysis MC_TTBAR \[B^{+12}\].

```cpp
#include "Rivet/Analysis.h"
#include "Rivet/Projections/FinalState.h"
#include "Rivet/Projections/VetoedFinalState.h"
#include "Rivet/Projections/ChargedLeptons.h"
#include "Rivet/Projections/MissingMomentum.h"
#include "Rivet/Projections/FastJets.h"
#include "Rivet/AnalysisLoader.h"
#include "Rivet/RivetAIDA.h"
#include "Rivet/Tools/Logging.h"

namespace Rivet {

    class MC_TTHMUON : public Analysis {
public:
        /// Minimal constructor
        MC_TTHMUON() : Analysis("MC_TTHMUON")
        {
        }

        /// @name Analysis methods
        //@

        /// Set up projections and book histograms
        void init() {
            // A FinalState is used to select particles within $|\eta| < 4.2$ and
            // $p_T > 30$ GeV, out of which the ChargedLeptons projection picks only
            // electrons and muons, to be accessed later as "LFS".
            ChargedLeptons lfs(FinalState(-4.2, 4.2, 30*GeV));
            addProjection(lfs, "LFS");
            // A second FinalState is used to select all particles in $|\eta| < 4.2$,
            // with no $p_T$ cut. This is used to construct jets and measure missing
            // transverse energy.
            VetoedFinalState fs(FinalState(-4.2, 4.2, 0*GeV));
            fs.addVetoOnThisFinalState(lfs);
            addProjection(FastJets(fs, FastJets::ANTIKT, 0.6), "Jets");
            addProjection(MissingMomentum(fs), "MissingET");

            // Booking of histograms
            _h_njets = bookHistogram1D("jet_mult", 11, -0.5, 10.5);
            //
            _h_jet_1_pT = bookHistogram1D("jet_1_pT", 50, 20.0, 500.0);
            _h_jet_2_pT = bookHistogram1D("jet_2_pT", 50, 20.0, 400.0);
            _h_jet_3_pT = bookHistogram1D("jet_3_pT", 50, 20.0, 300.0);
            _h_jet_4_pT = bookHistogram1D("jet_4_pT", 50, 20.0, 200.0);
            _h_jet_HT = bookHistogram1D("jet_HT", 50, 100.0, 2000.0);
            //
            _h_bjet_1_pT = bookHistogram1D("jetb_1_pT", 50, 20.0, 400.0);
            _h_bjet_2_pT = bookHistogram1D("jetb_2_pT", 50, 20.0, 300.0);
        }
    }
}
```
void analyze(const Event& event) {
    const double weight = event.weight();

    // Selection
    // Use the "LFS" projection to require at least three hard charged
    // lepton. This is an experimental signature for the leptonically
    // decaying
    // W. This helps to reduce pure QCD backgrounds.
    const ChargedLeptons& lfs = applyProjection<ChargedLeptons>(event, "LFS");
    MSG_DEBUG("Charged lepton multiplicity = " << lfs.chargedLeptons().size());
    foreach (const Particle& lepton, lfs.chargedLeptons()) {
        MSG_DEBUG("Lepton pT = " << lepton.momentum().pT());
    }

    // All events
    _h_CF->fill(-0.5);

    if (lfs.chargedLeptons().size() < 3) {
        MSG_DEBUG("Event failed lepton multiplicity cut");
        _h_CF->fill(0.5);
        vetoEvent;
    }
}
// Require 2 muons of opposite charge from H decay
vector<Particle> allmuons, restleptons;
bool mu_minus = false;
bool mu_plus = false;
foreach (const Particle& lepton, lfs.chargedLeptons()) {
  if (fabs(lepton.pdgId()) == 13) {
    allmuons.push_back(lepton);
    if (lepton.pdgId() == -13) {
      mu_minus = true;
    }
    if (lepton.pdgId() == 13) {
      mu_plus = true;
    }
  }
} else {
  restleptons.push_back(lepton);
}
if (allmuons.size() < 2) {
  MSG_DEBUG("Event failed muon multiplicity cut");
  _h_CF->fill(1.5);
  vetoEvent;
}
if ( !(mu_minus) || !(mu_plus) ) {
  MSG_DEBUG("Event failed muon charge cut");
  _h_CF->fill(2.5);
  vetoEvent;
}
// Use a missing ET cut to bias toward events with a hard neutrino from
// the leptonically decaying W. This helps to reduce pure QCD backgrounds.
const MissingMomentum& met = applyProjection<MissingMomentum>(event, "MissingET");
MSG_DEBUG("Vector ET = " << met.vectorEt().mod() << " GeV");
if (met.vectorEt().mod() < 30*GeV) {
  MSG_DEBUG("Event failed missing ET cut");
  _h_CF->fill(3.5);
  vetoEvent;
}
// Use the "Jets" projection to check that there are at least 4 jets of
// any pT. Getting the jets sorted by pT ensures that the first jet is the
// hardest, and so on. We apply no pT cut here only because we want to
// plot all jet pTs to help optimise our jet pT cut.
const FastJets& jetpro = applyProjection<FastJets>(event, "Jets");
const Jets alljets = jetpro.jetsByPt();
if (alljets.size() < 4) {
  MSG_DEBUG("Event failed jet multiplicity cut");
  _h_CF->fill(4.5);
  vetoEvent;
}
// Update passed--cuts counter and fill all-jets histograms
_h_jet_1_pT->fill(alljets[0].momentum().pT()/GeV, weight);
_h_jet_2_pT->fill(alljets[1].momentum().pT()/GeV, weight);
_h_jet_3_pT->fill(alljets[2].momentum().pT()/GeV, weight);
_h_jet_4_pT->fill(alljets[3].momentum().pT()/GeV, weight);
// Insist that the hardest 4 jets pass pT hardness cuts. If we don’t find
// at least 4 such jets, we abandon this event.
const Jets jets = jetpro.jetsByPt(30*GeV);
_h_njets->fill(jets.size(), weight);
double ht = 0.0;
foreach (const Jet& j, jets) { ht += j.momentum().pT(); }
_h_jetHT->fill(ht/GeV, weight);
if (jets.size() < 4 ||
jets[0].momentum().pT() < 60*GeV ||
jets[1].momentum().pT() < 50*GeV ||
jets[3].momentum().pT() < 30*GeV) {
MSGDEBUG("Event failed jet cuts");
_h_CF->fill(5.5);
vetoEvent;
}

/// Analysis
/// Sort the jets into b–jets and light jets. We expect one hard b–jet from
/// each top decay, so our 4 hardest jets should include two b–jets. The
/// Jet : : containsBottom() method is equivalent to perfect experimental
/// b–tagging, in a generator–independent way.
Jets bjets, ljets;
foreach (const Jet& jet, jets) {
    // // Don’t count jets that overlap with the hard leptons
    bool isolated = true;
    foreach (const Particle& lepton, lfs.chargedLeptons()) {
        if (deltaR(jet.momentum(), lepton.momentum()) < 0.3) {
            isolated = false;
            break;
        }
    }
    if (!isolated) {
        MSGDEBUG("Jet failed lepton isolation cut");
        break;
    }
    if (jet.containsBottom()) {
        bjets.push_back(jet);
    } else {
        ljets.push_back(jet);
    }
}
MSGDEBUG("Number of b–jets = " << bjets.size());
MSGDEBUG("Number of l–jets = " << ljets.size());
if (bjets.size() != 2) {
    MSGDEBUG("Event failed post–lepton–isolation b–tagging cut");
    _h_CF->fill(6.5);
vetoEvent;
}
if (ljets.size() < 2) {
    MSGDEBUG("Event failed since not enough light jets remaining after
    lepton–isolation");
    _h_CF->fill(7.5);
vetoEvent;
}

// Remaining events
_h_CF->fill(8.5);
// Plot the pTs of the identified jets.
_h_bjet_1_pT->fill(bjets[0].momentum().pT(), weight);
_h_bjet_2_pT->fill(bjets[1].momentum().pT(), weight);
_h_ljet_1_pT->fill(ljets[0].momentum().pT(), weight);
_h_ljet_2_pT->fill(ljets[1].momentum().pT(), weight);

// Construct the Higgs boson from a mu+ and a mu−. Analogous to W construction below.
FourMomentum H(10*sqrtS(), 0, 0, 0);
for (size_t i = 0; i < allmuons.size()-1; ++i) {
    for (size_t j = i + 1; j < allmuons.size(); ++j) {
        if (allmuons[i].pdgId() == -allmuons[j].pdgId()) {
            const FourMomentum Hcand = allmuons[i].momentum() + allmuons[j].momentum();
            MSG_TRACE(i <<"," << j <<": candidate H mass = " << Hcand.mass()/GeV
                    << " GeV, vs. incumbent candidate with ":
                    " << H.mass()/GeV << " GeV") ;
            if (fabs(Hcand.mass() - 125.9*GeV) < fabs(H.mass() - 125.9*GeV)) {
                H = Hcand;
            }
        }
    }
}
MSG_DEBUG("Candidate H mass = " << H.mass() << " GeV");
if ((H.mass() < 120.9) or (130.9 < H.mass())) {
    MSG_DEBUG("Event failed Higgs mass cut");
    _h_CF->fill(9.5);
    vetoEvent;
}
_h_H_mass->fill(H.mass(), weight);
_h_H_pT->fill(H.pT(), weight);

// Construct the hadronically decaying W momentum 4−vector from pairs of non−b−tagged jets. The pair which best matches the W mass is used. We start // with an always terrible 4−vector estimate which should always be "beaten" by a real jet pair.
FourMomentum W(10*sqrtS(), 0, 0, 0);
for (size_t i = 0; i < ljets.size()-1; ++i) {
    for (size_t j = i + 1; j < ljets.size(); ++j) {
        const FourMomentum Wcand = ljets[i].momentum() + ljets[j].momentum();
        MSG_TRACE(i <<"," << j <<": candidate W mass = " << Wcand.mass() /GeV
                    << " GeV, vs. incumbent candidate with " << W.mass() /GeV << " GeV") ;
        if (fabs(Wcand.mass() - 80.4*GeV) < fabs(W.mass() - 80.4*GeV)) {
            W = Wcand;
        }
    }
}
MSG_DEBUG("Candidate W mass = " << W.mass() << " GeV");
// There are two b-jets with which this can be combined to make the
// hadronically decaying top, one of which is correct and the other
// not... but we have no way to identify which is which, so we
// construct
// both possible top momenta and fill the histograms with both.
const FourMomentum t1 = W + bjets[0].momentum();
const FourMomentum t2 = W + bjets[1].momentum();
_h_W_mass->fill(W.mass(), weight);
_h_t_mass->fill(t1.mass(), weight);
_h_t_mass->fill(t2.mass(), weight);

// Placing a cut on the well-known W mass helps to reduce backgrounds
if (inRange(W.mass()/GeV, 75.0, 85.0)) {
  MSG_DEBUG(" W found with mass ", W_mass/GeV);
  _h_W_mass_W_cut->fill(W.momentum(), weight);
  _h_t_mass_W_cut->fill(t1.momentum(), weight);
  _h_t_mass_W_cut->fill(t2.momentum(), weight);

  _h_jetb_1.jetb_2.dR->fill(deltaR(bjets[0].momentum(), bjets[1].momentum()), weight);
  _h_jetb_1.jetb_2.deta->fill(fabs(bjets[0].momentum().eta()-jets[1].momentum().eta()), weight);
  _h_jetb_1.jetb_2.dphi->fill(deltaPhi(bjets[0].momentum(), bjets[1].momentum()), weight);

  _h_jetb_1.jetl_1.dR->fill(deltaR(bjets[0].momentum(), ljets[0].momentum()), weight);
  _h_jetb_1.jetl_1.deta->fill(fabs(bjets[0].momentum().eta()-ljets[0].momentum().eta()), weight);
  _h_jetb_1.jetl_1.dphi->fill(deltaPhi(bjets[0].momentum(), ljets[0].momentum()), weight);

  _h_jetl_1.jetl_2.dR->fill(deltaR(ljets[0].momentum(), ljets[1].momentum()), weight);
  _h_jetl_1.jetl_2.deta->fill(fabs(ljets[0].momentum().eta()-ljets[1].momentum().eta()), weight);
  _h_jetl_1.jetl_2.dphi->fill(deltaPhi(ljets[0].momentum(), ljets[1].momentum()), weight);

  _h_jetb_1.W.dR->fill(deltaR(bjets[0].momentum(), W), weight);
  _h_jetb_1.W.deta->fill(fabs(bjets[0].momentum().eta()-W.eta()), weight);
  _h_jetb_1.W.dphi->fill(deltaPhi(bjets[0].momentum(), W), weight);

  _h_ljets_1.ljets_2.dR->fill(deltaR(ljets[0].momentum(), ljets[1].momentum()), weight);
  _h_ljets_1.ljets_2.deta->fill(fabs(ljets[0].momentum().eta()-ljets[1].momentum().eta()), weight);
  _h_ljets_1.ljets_2.dphi->fill(deltaPhi(ljets[0].momentum(), ljets[1].momentum()), weight);

  _h_W->fill(W.momentum(), weight);
  _h_ljets->fill(ljets[0].momentum(), weight);
  _h_ljets->fill(ljets[1].momentum(), weight);
  _h_ljets->fill(ljets[2].momentum(), weight);
  _h_ljets->fill(ljets[3].momentum(), weight);
  _h_ljets->fill(ljets[4].momentum(), weight);
}

void finalize() {
  normalize(_h_njets);
  normalize(_h_jet_1.pT);
  normalize(_h_jet_2.pT);
  normalize(_h_jet_3.pT);
  normalize(_h_jet_4.pT);
  normalize(_h_jet_HT);
normalize(_h_bjet_1_pT);
normalize(_h_bjet_2_pT);
normalize(_h_jet_1_pT);
normalize(_h_jet_2_pT);
normalize(_h_H_mass);
normalize(_h_H_pT);
normalize(_h_W_mass);
normalize(_h_t_mass);
normalize(_h_t_mass_W_cut);
normalize(_h_jetb_1_jetb_2_dR);
normalize(_h_jetb_1_jetb_2_deta);
normalize(_h_jetb_1_jetb_2_dphi);
normalize(_h_jetb_1_jetl_1_dR);
normalize(_h_jetb_1_jetl_1_deta);
normalize(_h_jetb_1_jetl_1_dphi);
normalize(_h_jetb_1_W_dR);
normalize(_h_jetb_1_W_deta);
normalize(_h_jetb_1_W_dphi);
AIDA::IHistogram1D *_h_njets;
AIDA::IHistogram1D *_h_jet_1_pT , _h_jet_2_pT , _h_jet_3_pT , _h_jet_4_pT ;
AIDA::IHistogram1D *_h_jet_HT ;
AIDA::IHistogram1D *_h_bjet_1_pT , _h_bjet_2_pT ;
AIDA::IHistogram1D *_h_jet_1_pT , _h_bjet_2_pT ;
AIDA::IHistogram1D *_h_H_mass , _h_H_pT ;
AIDA::IHistogram1D *_h_W_mass ;
AIDA::IHistogram1D *_h_t_mass , _h_t_mass_W_cut ;
AIDA::IHistogram1D *_h_jetb_1_jetb_2_dR , _h_jetb_1_jetb_2_deta , _h_jetb_1_jetb_2_dphi ;
AIDA::IHistogram1D *_h_jetb_1_jetl_1_dR , _h_jetb_1_jetl_1_deta , _h_jetb_1_jetl_1_dphi ;
AIDA::IHistogram1D *_h_jetl_1_jetl_2_dR , _h_jetl_1_jetl_2_deta , _h_jetl_1_jetl_2_dphi ;
AIDA::IHistogram1D *_h_jetb_1_W_dR , _h_jetb_1_W_deta , _h_jetb_1_W_dphi ;
AIDA::IHistogram1D *_h_jetb_1_1_dR , _h_jetb_1_1_deta , _h_jetb_1_1_dphi , _h_jetb_1_1_mass ;
AIDA::IHistogram1D *_h_CF ;

// The hook for the plugin system
DECLARE_RIVET_PLUGIN(MC_TTH_MUON) ;