Simulations for Particle Physics: Implementing the Colour Dipole Model with Invariant Transverse Momentum Ordering

Author:
Daniel A. OSBORNE

Supervisors:
Professor Frank M. KRAUSS
Mr Holger SCHULZ

A thesis submitted in partial fulfilment of the requirements of the degree MPhys Theoretical Physics from Durham University.

April 2016

DURHAM UNIVERSITY

Abstract

MPhys Theoretical Physics

Simulations for Particle Physics: Implementing the Colour Dipole Model with Invariant Transverse Momentum Ordering

by Daniel A. OSBORNE

A final-state parton shower for an electron-positron collider was implemented in Python following the Colour Dipole Model. Shower ordering by Lorentz invariant transverse momentum and the massless parton approximation were utilised. Both gluon and photon emission were modelled along with gluon splitting to new quark-anti-quark pairs, following similar conventions to the ARIADNE program. The simulation was then configured on resonance with the Z-boson channel. Quark-anti-quark matrix elements were approximated, showered, and subsequently hadronised and decayed using the PYTHIA v8.2 event generator. Comparison with data from the DELPHI, ALEPH and OPAL experiments at LEP showed an improved fit with the shower, relative to matrix element results alone. Whilst some results were in agreement with the data, the simulation was estimated to have an average error of 10%. It was determined that the massless approximation was the greatest contribution, and therefore incorporating massive quark effects was the natural progression for the project.

Acknowledgements

I would first like to thank my supervisors; Professor Krauss, for many informative conversations on all aspects of parton showers, and Mr Schulz, for his assistance in understanding the LHEF format and how to use PYTHIA v8.2. Secondly my family, particularly my father, for their support throughout. Finally Mr Ryan Harris, for discussions on the topic as a whole.

Contents

1. Introduction

1.1 Thesis Aims

The scope of this thesis was to simulate in Python one part of an event generator for electron-positron (e^+e^-) collisions. The emphasis was on producing a working simulation which accurately models the underlying physics. From this, the modelling of a parton shower using the Colour Dipole Model (CDM) was chosen. The aim was to produce a working final-state parton shower (named $\mathrm{PyShower}$) for colour-paired quark-anti-quark ($q\bar{q}$) matrix element (ME) results from the e^+e^- collider. The results produced would then be tested using existing simulation programs and comparison to data from the Large Electron-Positron Collider (LEP).

1.2 Background

Particle colliders are currently pushing the Standard Model (SM) to its limits in the search for new physics. They rely on matching signals generated in experiments to those predicted by theoretical simulations, in order to identify new events from the background of known events. Matching these known SM events also affords experimental physicists the opportunities of tuning new detectors, measuring SM parameters or confirming the existence of a theorised particle. Meanwhile, theoretical physicists have the opportunity to gain further insight into the underlying physics, from which they may refine their existing models or suggest directions for developing completely new theories, which in turn may influence the design of future particle accelerators.

The SM is underpinned by the $\mathrm{SU}(3)$ gauge theory of Quantum Chromodynamics (QCD) which governs the interactions of all colour-charged partons (quarks and gluons) via the strong force. A particle collider event can be broken down into different subprocesses, outlined in the sections below. Most of these can be treated perturbatively through QCD and are directly calculable to Leading Order (LO) or Next to Leading Order (NLO). Meanwhile, other subprocesses rely on Monte Carlo (MC) techniques to approximate the results for all orders. The combination of subprocesses utilising these methods forms the basis of all event simulation programs, either in one program or by interfacing multiple specialised models.

1.3 The Structure of an Event

A single particle collision, such as the e^+e^- collision studied, is often referred to as an event, and has the potential to produce a multitude of particles from the SM, with a large range of momenta. The process of factorisation divides each event into regimes according to the momentum transfer involved, which it is assumed can be treated separately. The higher order (hard) subprocesses are treated perturbatively through QCD, but for lower order (softer) subprocesses, this treatment breaks down or becomes too computationally demanding to calculate directly, so MC techniques are employed.

The initial interaction of two particles, the Primary Hard Subprocess (PHS) (Section 1.3.1) is the highest order regime and deals with the first order interaction between the collider beam particles. The particles produced in this regime are then passed to a Parton Shower (PS) algorithm (Section 1.3.2) which is an evolutionary process dealing with decreasing momentum transfers. Matching and Merging (MM) (Section 1.3.3) takes place next, tying together the two initial processes. At the same scale, the Underlying Event (UE) (Section 1.3.4) incorporates interactions between separately showered particles. Having reached the softer regime, results are passed into a Hadronisation module (Section 1.3.5) and finally a Decay simulation (Section 1.3.6) if appropriate.

This factorisation naturally lends itself to the modular form used in modern particle simulations, where particles are passed through an individual simulation for each subprocess. Standardised particle and event formats ensure this is carried out correctly if multiple programs are used. A brief approximation to the PHS was undertaken, as discussed in Section 2.2. Aside from this and the PS, no other subprocesses were treated and are simply outlined below to provide a complete picture of an event.

1.3.1 Primary Hard Subprocess

The PHS deals with the initial stage of an event, at the highest momentum transfer regime. For an e^+e^- collision this includes the $e^+e^- \to q\bar{q}$ process under consideration. In this hard regime, colour charges are weakly coupled (see Section 2.1) and this asymptotic freedom allows calculating tree level matrix elements through the perturbative treatment of QCD cross sections. Due to the size of the phase space which these must be integrated over, calculations beyond the first few orders are often too arduous. Consequently MEs are usually calculated to LO or NLO at best.

1.3.2 Parton Showers

Just as accelerated Electromagnetic (EM) charges radiate photons (Bremsstrahlung radiation) in the process $e^+ \to e^+\gamma$, accelerated colour charges likewise radiate gluons (e.g. $q \to qg$). The added complication with QCD is that these gluons are themselves colour-charged so can in turn radiate successive gluons ($g \to gg$) or split into a new quark-anti-quark pair ($g \to Q\bar{Q}$). Consequently a shower of partons is produced, each with decreasing momentum transfer and hence the shower acts as a natural evolutionary process from the hard to the soft regimes. As each emission is taken to be probabilistic and only depended upon the current state of a given particle, the process is an example of a Markov chain and lends itself to MC modelling. In theory this evolution could be included in the ME calculations, but due to the computational inefficiency implied in Section 1.3.1, MC techniques are the method of choice.

Section 2.4 outlines how this modelling can encompass the effects of all higher order correction terms to the PHS in the dominant collinear and soft regions of the emission phase space. Whilst primarily used for QCD radiation, parton showers can also include EM radiation. Finally it is noted that both initial-state and final-state parton showers are possible, for showering from the incoming and outgoing partons of the PHS respectively. However, for the e^+e^- colliders considered, initial-state radiation would only involve photon emission and so is not treated here.

1.3.3 Matching and Merging

The MEs and PS are complimentary processes, with MEs preferential for harder partons whilst softer or collinear partons are better handled by a PS. Consequently a combination of the two techniques is commonly applied to simulate partons at all scales of momentum transfer. Matching and merging of the two methods must then take place, in order to prevent double or under counting of phase space regions.

1.3.4 Secondary Interactions

Initial and final-state PSs are simulated separately meaning interactions between partons in each must still be accounted for. These secondary interactions are a soft QCD process simulated via futher MC techniques and are often referred to as the underlying event.

1.3.5 Hadronisation

Hadronisation simulates the next stage of the event as the colour-connected partons created are confined into colourless mesons and baryons. As discussed in Section 2.1, QCD is asymptotically free, meaning the interaction strength increases inversely with the energy scale. The momentum transfer evolution therefore reaches a cut-off where perturbative calculations through QCD are no longer possible, which is where hadronisation models influenced by but not derived from QCD take over.

1.3.6 Decays

The final step for a full event simulation is to incorporate the decay of unstable particles from hadronisation, which might not reach a detector. As this process is probabilistic, MC techniques are employed again.

1.4 The Colour Dipole Model

The model for parton showers used is the Colour Dipole Model (CDM) [1–3], as first implemented by the ARIADNE program [4]. It has proven a popular choice in recent parton shower developments, such as PYTHIA v8.2 [5] where it is slowly replacing the old angularly ordered shower. In this model, partons are constructed into colour-connected dipoles containing a colour and anti-colour pair. Dipoles may be further connected together into a chain from which each dipole has the chance to undergo a branching. Branchings split their dipole, leaving two dipoles in the chain, whilst gluons splitting to a $Q\bar{Q}$ pair split the chain itself. Each chosen emission is treated independently, and then any gluon links to other dipoles are updated which allows the shower to evolve globally. Using the CDM's $2 \to 3$ process is advantageous as energy and momentum are automatically conserved, whilst emissions from both colour lines of the gluon are naturally included via the dipoles. Strong ordering through transverse momentum and further advantages are discussed in Section 2.4 but it is noted here that the CDM provides a good shower approximation in this limit. The model is also known for its success with e^+e^- event simulations which, combined with the above, is why it was chosen.

2. Theory

This chapter outlines the theory which underpins parton showers, from which the simultaion PyShower was written. Further discussion on the applications of some theory sections can be found in Section 3 and Appendix A.

2.1 Running of the Coupling Constants

In QED and QCD, quantum fluctuations induce vacuum polarisation effects which lead to charge screening and a running energy dependence for the coupling constants. An understanding of this running (particularly for the strong coupling constant) proves essential throughout a parton shower (as briefly discussed in section 1.3.5) and so the effect is outlined in this section.

2.1.1 The Fine Structure Constant

In QED, electrons are constantly emitting and re-absorbing virtual photons. First order (in α_{EM}) interactions can be attributed to the absorption of one of these by a neighbouring electron. More complicated processes also occur where the photon itself fluctuates to an e^+e^- pair, leading to one-loop correction terms if the photon is subsequently absorbed by the neighbouring electron. Such e^+e^- pairs behave as dipoles and align themselves such that a neighbouring charge interacts as if with a smaller effective charge than usual. This screening is less effective at shorter distances (and hence higher energies) leading to the running of the coupling constant. To first order the fine structure constant is given by [6]:

$$\alpha_{EM}^{(1)}\left(Q^2\right) = \frac{\alpha_{EM}\left(M_Z^2\right)}{1 - \left(\alpha_{EM}\left(M_Z^2\right)/3\pi\right)\ln\left(Q^2/M_Z^2\right)}, \tag{2.1}$$

where the strength of the coupling is defined relative to a reference value, $\alpha_{EM}\left(M_Z^2\right)$ where $\alpha_{EM}^{-1}\left(M_Z^2\right) = 128.886 \pm 0.090$ was used [7].

2.1.2 The Strong Coupling Constant

Similarly to QED, $q\bar{q}$ pairs cause shielding in QCD. However, as the gluon is itself colour charged, gluon loops are also produced via vacuum fluctuations. The effects

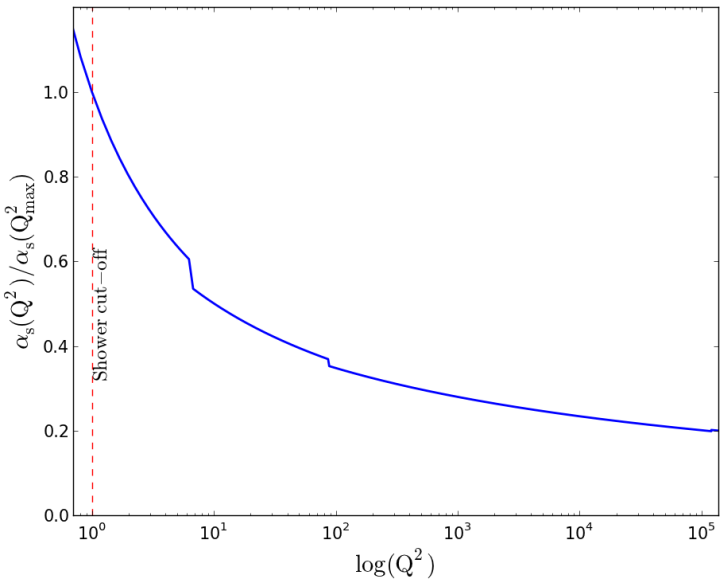

FIGURE 2.1: Weighted One Loop α_s as a function of $\log Q^2$. The red line indicates the parton shower cut-off energy used throughout.

of these dominate those from the $q\bar{q}$ pairs and conversely cause anti-screening of the original colour charge. This results in the asymptotic freedom associated with non-Abelian gauge theories, where the interaction strength decreases as energy increases. The first order running of the strong coupling constant is given by [8]:

$$\alpha_S^{(1)}\left(Q^2\right) = \frac{\alpha_S\left(M_Z^2\right)}{1 + \alpha_S\left(M_Z^2\right)\beta_0\left(N_f\right)\ln\left(Q^2/M_Z^2\right)}, \tag{2.2}$$

$$\beta_0\left(N_f\right) = \frac{\frac{11}{3}C_A - \frac{4}{3}T_r N_f}{4\pi}, \qquad C_A = 2T_r N_c, \tag{2.3}$$

where $T_r = 0.5$ is a colour factor fixed by convention and $N_c = 3$ is the number of colours. Here a reference value of $\alpha_S\left(M_Z^2\right) = 0.118 \pm 0.003$ was used [8]. Figure 2.1 shows a plot of Equation 2.2 weighted on its maximum value at the shower cut-off (see Section 2.5.2). The discontinuities occur due to jumps in the number of available quark flavours but could be smoothed out by using a separate reference value for each section. However, within the scope of this project, it was deemed that the loss of precision would be greater elsewhere in the shower.

2.2 Matrix Elements

Whilst not the focus of this project, matrix element results are required to provide the initialising partons. The process $e^+e^- \to q\bar{q}$ was approximated with partons distributed in direction according to the differential cross section for $e^+e^- \to Z^0 \to f\bar{f}$ [9]:

$$\begin{aligned}\frac{\mathrm{d}\sigma^{(0)}}{\mathrm{d}\Omega} = \frac{{\alpha_{EM}}^2}{4S_{123}}\Bigg\{&\Big[1 + 2g_v^{(e)}g_v^{(f)}\mathrm{Re}\left(\chi\left(S_{123}\right)\right) \\ &+ \left({g_v^{(e)}}^2 + {g_a^{(e)}}^2\right)\left({g_v^{(f)}}^2 + {g_a^{(f)}}^2\right)\left|\chi\left(S_{123}\right)\right|^2\Big]\left(1+\cos^2\theta\right) \\ &+ \left[4g_a^{(e)}g_a^{(f)}\mathrm{Re}\left(\chi\left(S_{123}\right)\right) + 8g_v^{(e)}g_a^{(e)}g_v^{(f)}g_a^{(f)}|\chi\left(S_{123}\right)|^2\right]\cos\theta\Bigg\},\end{aligned} \tag{2.4}$$

where Ω is the solid angle with $\phi \in [0, 2\pi]$ and $\cos\theta \in [-1, 1]$, f is a fermion and:

$$\chi\left(S_{123}\right) = \left(\frac{1}{2\sin\theta_W\cos\theta_W}\right)^2 \frac{S_{123}}{S_{123} - {M_Z}^2 + iM_Z\Gamma_Z}. \tag{2.5}$$

Table 2.1 contains the neutral weak couplings, $g_{a,v}^{(f)}$, as a function of the weak mixing angle, θ_W, where the value used throughout was $\sin^2\theta_W = 0.23126 \pm 0.00005$ [10].

	$g_a^{(f)}$	$g_v^{(f)}$
$g_{a,v}^{(e,\mu,\tau)}$	-0.5	$2\sin^2\theta_W - 0.5$
$g_{a,v}^{(u,c,t)}$	0.5	$0.5 - \frac{4}{3}\sin^2\theta_W$
$g_{a,v}^{(d,s,b)}$	-0.5	$\frac{2}{3}\sin^2\theta_W - 0.5$
$g_{a,v}^{(\nu_e,\nu_\mu,\nu_\tau)}$	0.5	0.5

TABLE 2.1: Neutral weak coupling terms for fermion f as a function of the weak mixing angle, θ_W.

The use of a differential cross section on resonance with the Z-boson channel is in line with the intention to test the code against results from LEP, at $\sqrt{S_{123}} = M_Z = 91.188\ \mathrm{GeV}$. This is the only extent to which matrix elements were considered. The process of sampling this distribution is recapped in Appendix B and tests of the algorithms are presented in Appendix A.4.

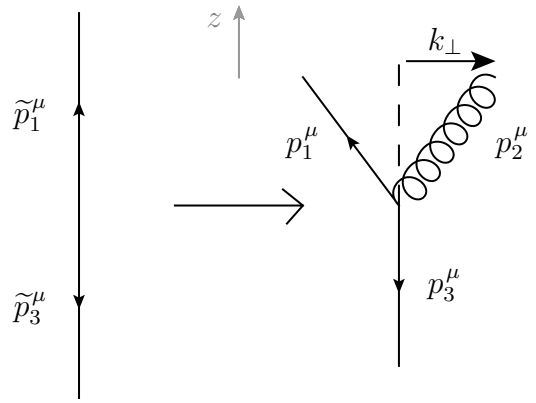

FIGURE 2.2: Schematic for the kinematics before and after dipole branching to a gluon in the orientated centre of mass frame of the initial partons.

2.3 Massless Kinematics

Only the approximation to massless partons was considered when solving the dipole branching kinematics and the notation $p_i^\mu = \left(E_i, P_{\perp i}, P_{\parallel i}\right)$ was adopted for the four-vector of parton i, where $\sim$ identifies an initial parton. Calculations were best processed in the oriented (along z) Centre of Mass Frame (CMF) of the two branching dipole partons, $\widetilde{p}_1$ and $\widetilde{p}_3$, as shown in Figure 2.2 for gluon emission from a $q\bar{q}$ pair. Therefore Lorentz transformations between the CMF and lab frame were implemented before and after every branching. The convention of $\widetilde{p}_3$ recoiling and maintaining its direction whilst $\widetilde{p}_1$ received a transverse momentum, $-k_\perp$ was used, implying p_2 was produced with a transverse momentum $k_\perp$. For each branching, the parton to recoil was assigned following the rules in Section 3.2, so as to 'minimise disturbance to the colour flow' [3]. As discussed in Section 2.4, the shower could have been ordered in $k_\perp$, but instead two Lorentz invariant evolution variables were chosen, defined as [11]:

$$P_\perp^2 = \frac{S_{12}S_{23}}{S_{123}}, \qquad\qquad y = \frac{1}{2}\log\left(\frac{S_{23}}{S_{12}}\right). \tag{2.6}$$

These ensured conservation of the shower ordering during the Lorentz transformations and are approximately the transverse momentum and rapidity of p_2 respectively. They are defined through $S_{ij\{k\}}$, the squared invariant masses of the $ij\{k\}$ systems:

$$S_{ijk} = \left(P_i^\mu + P_j^\mu + P_k^\mu\right)^2, \qquad\qquad S_{123} = S_{12} + S_{23} + S_{13}. \tag{2.7}$$

Rearranging gave the invariant masses as functions of the evolution variables:

$$S_{12,23} = \sqrt{S_{123}} P_\perp e^{\pm y}, \qquad S_{13} = S_{123} - 2\sqrt{S_{123}} P_\perp \cosh y. \tag{2.8}$$

Using these along with energy and momentum conservation in the CMF gave the new energy values:

$$E_{1,3} = \frac{1}{2}\left(\sqrt{S_{123}} - P_\perp e^{\pm y}\right), \qquad E_2 = P_\perp \cosh y, \tag{2.9}$$

from which $k_\perp$ could be calculated as:

$$k_\perp = \sqrt{E_1^2 - \left(\frac{E_1^2 - E_2^2 + E_3^2}{2E_3}\right)^2}. \tag{2.10}$$

Resolving $k_\perp$ into its two components and utilising the Minkowski inner product to ensure the new four-vectors were also lightlike gave the final equations:

$$P_{\parallel,i} = \sqrt{E_i^2 - k_\perp^2}, \qquad k_{\perp,x} = \pm k_\perp \cos(\phi), \qquad k_{\perp,y} = \pm k_\perp \sin(\phi), \tag{2.11}$$

where $i \in \{1,2,3\}$, ϕ is the uniformly distributed angle of p_2 in the x,y plane relative to the x-axis, $k_{\perp 3} = 0$ by definition and the $\pm$ is for p_1 and p_2 respectively.

2.4 Shower Ordering

Multiple ordering variables are available which act akin to time throughout a parton shower. This section discusses the choice of evolution variable made and the motivation behind it.

As parton showers simulate next-order processes such as $e^+e^- \to qg\bar{q}$, the differential cross section for these must be examined. Consider [12]:

$$\frac{\mathrm{d}\sigma_{e^+e^-\to qg\bar{q}}}{\mathrm{d}\cos\theta_{qg}\mathrm{d}z} \approx \frac{C_F\,\alpha_S}{\pi \sin^2\theta_{qg}} \frac{1+(1-z))^2}{z} \times \sigma_{e^+e^-\to q\bar{q}}, \tag{2.12}$$

where $\theta_{qg} \in [0,\pi]$ is the opening angle between the quark and gluon, $z \in [0, z_{max}]$ is the energy fraction of the gluon and $C_F = (N_c^2 - 1)/2N_c$ is a colour factor. Evidently $\theta_{qg} \to 0$, $\theta \to \pi$ and $z \to 0$ at the edges of the phase space each cause Equation 2.12

to diverge. These conditions are consistent with collinear gluon emission (parallel to q or $\bar{q}$ respectively) and soft gluon emission (of low energy) which must dominate the emission phase space. Re-writing Equation 2.12 to include $\theta_{g\bar{q}}$, the angle between the gluon and anti-quark, gives [12]:

$$\frac{2}{\sin^2\theta_{qg}} = \frac{1}{1-\cos\theta_{qg}} + \frac{1}{1+\cos\theta_{qg}} \approx \frac{1}{1-\cos\theta_{qg}} + \frac{1}{1-\cos\theta_{g\bar{q}}}, \tag{2.13}$$

$$\mathrm{d}\sigma_{qg\bar{q}} \approx \sigma_{q\bar{q}} \sum_{partons} \frac{C_F\,\alpha_S}{2\pi} \frac{\mathrm{d}\theta^2}{\theta^2} \frac{1+(1-z))^2}{z} \mathrm{d}z. \tag{2.14}$$

The differential cross section is now written as the sum of emissions from each initial dipole parton, where θ is the angle between each and the emitted parton (i.e between p_2^μ and the z-axis in Figure 2.2 for emission from parton $\widetilde{p}_1$). It is noted that it is therefore possible to treat the two emissions independently in this collinear limit which leads to the notion of spectator and emitter partons used by some shower algorithms. Instead, the term $\mathrm{d}\theta^2/\theta^2$ is of most interest here.

To move from this inclusive distribution of all emissions to sampling individual dipole branchings, θ is introduced as an evolution variable. It can be shown through considerations of QCD coherence that soft gluon emission is also correctly accounted for in a collinear parton shower algorithm, provided that opening angular ordering is observed [12]. However, the relation $k_\perp^2 = z^2\,(1-z)^2\,\theta^2 E^2$ [12], which gives $\mathrm{d}\theta^2/\theta^2 = \mathrm{d}k_\perp^2/k_\perp^2$, implies that parametrising the phase space in terms of the gluon's transverse momentum, $k_\perp$ from Section 2.3, also produces a mathematically identical result in the collinear limit to Equation 2.14. Equally, $k_\perp$ correctly includes angular ordering when using the CDM, making it a suitable evolution variable too [12].

Finally, as $P_\perp$ approximates to $k_\perp$ in the soft and collinear limits [13], strict ordering in $P_\perp^2$ was used to benefit from its Lorentz invariant properties discussed in Section 2.3. Further benefits of this choice of ordering variable are noted in Section 2.5.2 where the branching phase space is examined in more detail.

2.5 Dipole Branchings

The kinematics of a branching in Section 2.3 required values for $P_\perp^2$ and y. Beyond this, it had to be decided if a branching would occur and if so, of what nature. The generation of these variables consistent with QCD and the shower ordering from Section 2.4 is discussed below.

2.5.1 Branching Cross Sections

Analogous with matrix element generation in Section 2.2, differential cross sections were used to describe the probability of a branching at given $P_\perp^2$ and y values. These are well described by calculation from the corresponding QCD Feynman diagrams of the process, using factorisation from the total ME differential cross section as in Equation 2.12. Sampling these distributions ensures the simulation correctly reproduces the physics behind the soft and collinear terms that the parton shower encapsulates.

Unfortunately the Lorentz invariant variables previously discussed, although most convenient for shower ordering and the kinematics, were not most convenient for the branching cross sections. Instead, a variable transformation was performed once in the orientated CMF of a dipole, by introducing the energy fractions x_i of parton i [1, 4]:

$$P_\perp^2 = S_{123}\left(1-x_1\right)\left(1-x_3\right), \qquad y = \frac{1}{2}\log\left(\frac{1-x_1}{1-x_3}\right), \tag{2.15}$$

$$x_i = \frac{2E_i}{\sqrt{S_{123}}}, \qquad \mathrm{d}x_1\mathrm{d}x_3 = \mathrm{d}P_\perp^2\mathrm{d}yS_{123}. \tag{2.16}$$

The spin-averaged differential cross sections used were [1, 3]:

$$\frac{\mathrm{d}\sigma_{q\bar{q}\to qg\bar{q}}}{\mathrm{d}x_1\mathrm{d}x_3} = \frac{2\alpha_S}{3\pi}\frac{x_1^2+x_3^2}{\left(1-x_1\right)\left(1-x_3\right)}, \tag{2.17}$$

$$\frac{\mathrm{d}\sigma_{qg\to qgg}}{\mathrm{d}x_1\mathrm{d}x_3} = \frac{3\alpha_S}{4\pi}\frac{x_1^2+x_3^3}{\left(1-x_1\right)\left(1-x_3\right)}, \tag{2.18}$$

$$\frac{\mathrm{d}\sigma_{gg\to ggg}}{\mathrm{d}x_1\mathrm{d}x_3} = \frac{3\alpha_S}{4\pi}\frac{x_1^3+x_3^3}{\left(1-x_1\right)\left(1-x_3\right)}, \tag{2.19}$$

$$\frac{\mathrm{d}\sigma_{q\bar{q}\to q\gamma\bar{q}}}{\mathrm{d}x_1\mathrm{d}x_3} = \frac{\alpha_{EM}}{2\pi}e_q^2\frac{x_1^2+x_3^2}{\left(1-x_1\right)\left(1-x_3\right)}, \tag{2.20}$$

$$\frac{\mathrm{d}\sigma_{qg\to qQ\bar{Q}}}{\mathrm{d}x_1\mathrm{d}x_3} = \frac{3\alpha_S}{8\pi}\frac{\left(1-x_1\right)^2+\left(1-x_2\right)^2}{\left(1-x_3\right)}, \tag{2.21}$$

where the convention that p_2 is produced and $\widetilde{p}_3$ recoils is maintained from Section 2.3. It is noted that gluon and photon emission differ only in prefactor, implying the suppression of photon emission due to the relative coupling strengths of the EM and strong forces as discussed in Section 2.1. Furthermore, the utilisation of the running coupling constants in Equations 2.17 - 2.21 as functions of $k_\perp^2$ (and by extension $P_\perp^2$) encompasses higher-order Feynman diagrams, such as loop corrections to an emitted gluon, summed over all orders in α_S [12]. The sampling of these distributions is discussed in Section 2.6 and tests of the algorithms are found in Appendix A.4.

Gluons splitting to a new quark-anti-quark pair in Equation 2.21 were treated identically to all other emissions. The incoming gluon, $\widetilde{p}_1$, branched to p_1 and p_2, where the assignment of particle and anti-particle was dependant upon conservation of the colour connections of the active dipole. It is apparent that Equation 2.21 is equally valid for the process $gg \to gQ\bar{Q}$.

2.5.2 Emission Phase Space

The phase space available during each branching also required consideration as it was integrated over to determine the probability of a given branching occurring (see Section 2.5.3). Considering massless partons and requiring energy and momentum conservation, a first obvious limit was $P_{\perp max}^2 < S_{123}/4$, which corresponds to $x_3 \to 0$. Meanwhile, the rapidity limits were determined from Equations 2.7 and 2.8 as follows [11]:

$$S_{123} - S_{13} \leq S_{123}\,, \qquad\qquad |y| \leq \cosh^{-1}\left(\frac{\sqrt{S_{123}}}{2P_\perp}\right). \tag{2.22}$$

It also became apparent (see Section 2.6.1) that an overestimate for the rapidity limits was required. This was calculated in a similar manner to the actual limits:

$$S_{12,23} \leq S_{123}\,, \qquad\qquad |y| \leq \ln\left(\frac{\sqrt{S_{123}}}{P_\perp}\right). \tag{2.23}$$

The branching phase space using both Equations 2.22 and 2.23 are shown in Figure 2.3. The choice of $P_\perp$ as the shower ordering variable is further justified here, as the emission phase space is shown to increase for $P_\perp \to 0$ This also implies the downward evolution expected between the hard ME and soft hadronisation scales.

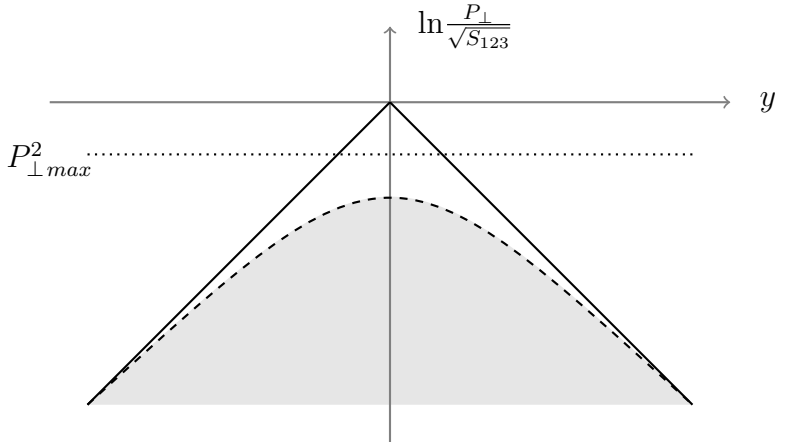

FIGURE 2.3: The available phase space for a branching (shaded) from Equation 2.22 (dashed black line) compared to the overestimate boundaries from Equation 2.23 (solid black lines).

Finally, each dipole branching $\widetilde{p}_1\ \widetilde{p}_3 \rightarrow p_1\ p_2\ p_3$ must follow the differential cross sections discussed in Section 2.5.1. In order to deal with their divergences (see Section 2.4), which lead to an infinite probability for indistinguishable collinear or soft emissions, a shower cut-off was introduced. When reached, the shower simulation was stopped. This is often associated with a scale at which the increasing strength of the strong coupling renders the perturbative QCD treatment inherent in parton showers invalid. Beyond this, hadronisation mechanisms take over.

Once again evolution in $P^2_\perp$ is justified as the cut-off value, $P^2_{\perp c}$, naturally deals with both the soft and collinear divergences (unlike angular ordering for example). A value of $1\ \mathrm{GeV}$ [12] was used throughout, as plotted in Figure 2.1 relative to the strong coupling constant.

2.5.3 Sudakov Form Factors

To ensure the downward ordering of $P^2_\perp$ from Section 2.4, the probability of a branching occurring at a given $P^2_\perp$ must include the probability that no branching has already occurred at a higher $P^2_\perp$, which is known as the Sudakov form factor, Δ:

$$\mathrm{P}\left(P^2_\perp\right) = \frac{-\mathrm{d}\Delta\left(P^2_\perp\right)}{\mathrm{d}P^2_\perp} = \sigma\left(P^2_\perp\right)\Delta\left(P^2_\perp\right). \tag{2.24}$$

Solving this for Δ gave:

$$\Delta\left(P_{\perp}^{2}\right)=\Delta\left(P_{\perp max}^{2}\right)\exp\left\{-\int_{P_{\perp}^{2}}^{P_{\perp max}^{2}}\mathrm{d}P_{\perp}^{2'}\sigma\left(P_{\perp}^{2'}\right)\right\}. \tag{2.25}$$

Then setting $\Delta\left(P_{\perp max}^{2}\right)=1$ and substituting back into Equation 2.24 produced:

$$\mathrm{P}\left(P_{\perp}^{2}\right)=\sigma\left(P_{\perp}^{2}\right)\exp\left\{-\int_{P_{\perp}^{2}}^{P_{\perp max}^{2}}\mathrm{d}P_{\perp}^{2'}\sigma\left(P_{\perp}^{2'}\right)\right\}. \tag{2.26}$$

An added complication arose through the need to sample both $P_{\perp}^{2}$ and y from the same differential cross sections. The final result used was [4]:

$$\frac{\mathrm{dP}\left(P_{\perp}^{2},y\right)}{\mathrm{d}P_{\perp}^{2}\mathrm{d}y}=\frac{\mathrm{d}\sigma\left(P_{\perp}^{2},y\right)}{\mathrm{d}P_{\perp}^{2}\mathrm{d}y}\exp\left\{-\int_{P_{\perp}^{2}}^{P_{\perp max}^{2}}\mathrm{d}P_{\perp}^{2'}\left(\int_{y_{min}\left(P_{\perp}^{2'}\right)}^{y_{max}\left(P_{\perp}^{2'}\right)}\mathrm{d}y'\frac{\mathrm{d}\sigma\left(P_{\perp}^{2'},y'\right)}{\mathrm{d}P_{\perp}^{2'}\mathrm{d}y'}\right)\right\}. \tag{2.27}$$

Sampling this probability distribution in accordance with the phase space limits from Section 2.5.2 ensured that all of the $P_{\perp}^{2},y$ pairs were generated correctly. This was done using the Veto Algorithm discussed in Section 2.6.1.

2.5.4 Competing Branchings

A further complication when sampling the emission probability as in Section 2.5.3 stems from the fact that there are often multiple competing branchings possible for a given dipole. A $q\bar{q}$ dipole is not only a colour dipole but also an electromagnetic dipole and therefore has the potential to produce either a photon or a gluon ($q\bar{q}\rightarrow q\gamma\bar{q}$ or $q\bar{q}\rightarrow qg\bar{q}$). For a dipole containing exactly one gluon, both the production of an additional gluon or the splitting of the existing gluon are possible. Finally for a gg dipole the splitting of either gluon or the emission of a third are all competing processes. Therefore the probability of a chosen emission, i, occurring actually depends upon the probability that none of the j possible competing processes have already occurred:

$$\frac{\mathrm{dP}_{i}\left(P_{\perp}^{2},y\right)}{\mathrm{d}P_{\perp}^{2}\mathrm{d}y}=\frac{\mathrm{d}\sigma_{i}\left(P_{\perp}^{2},y\right)}{\mathrm{d}P_{\perp}^{2}\mathrm{d}y}\prod_{j}\exp\left\{-\int_{P_{\perp}^{2}}^{P_{\perp max}^{2}}\mathrm{d}P_{\perp}^{2'}\left(\int_{y_{min}\left(P_{\perp}^{2'}\right)}^{y_{max}\left(P_{\perp}^{2'}\right)}\mathrm{d}y'\frac{\mathrm{d}\sigma_{j}\left(P_{\perp}^{2'},y'\right)}{\mathrm{d}P_{\perp}^{2'}\mathrm{d}y'}\right)\right\}. \tag{2.28}$$

2.6 Monte Carlo Techniques

The probabilistic nature of parton showers lends itself to the use of MC techniques in their simulation. The differential cross sections alone can only give the probability of the values $P_{\perp i}$ and y_i occurring. Instead, the requirement was to sample single values such that the correct probability distribution was recovered over a large number of iterations. Clearly, pre-calculating probabilities and sampling with a random number would be inaccurate and inefficient. This section covers the techniques applied throughout to enable a more efficient sampling of the branching probability distributions. The simpler sampling of a one-dimensional probability distribution such as Equation 2.4 is recapped in Appendix B whilst tests on the content of this section are shown in Appendix A.4. The Veto Algorithm [14] is central to these discussions.

2.6.1 The Two-Dimensional Veto Algorithm

Two-dimensional sampling of $P^2_{\perp i}$ and y_i from Equation 2.27 is more complicated than the usual one-dimensional case (see Appendix B). In addition there is the inclusion of the non-emission probability (Sudakov form factor, Δ) with its $P^2_\perp$ dependant y limits. In order to correctly sample this probability distribution, a modified Veto Algorithm was used. With $f\left(P^2_\perp, y\right)$ as the cross section in Equation 2.27, a factorisable overestimation function $g\left(P^2_\perp, y\right) = g_1\left(P^2_\perp\right) g_2\left(y\right)$ was used, where $g\left(P^2_\perp, y\right) \geq f\left(P^2_\perp, y\right)$ is satisfied for the entire emission phase space. The algorithm was then:

1. Begin with $i = 0,\ P^2_{\perp 0} = P^2_{\perp max}$
2. Let $i = i + 1$ and generate $P^2_{\perp i} = \mathrm{G}_3^{-1}\left[\ln\left(\mathcal{R}_1\right) + \mathrm{G}_3\left(P^2_{\perp i-1}\right)\right]$
3. If $P^2_{\perp i} < P^2_{\perp c}$ return None
4. If $P^2_{\perp i} > S_{123}/4$ return to step 2
5. Generate $y_i = \mathrm{G}_2^{-1}\left[\mathcal{R}_2\left[\mathrm{G}_2\left(y_+\right) - \mathrm{G}_2\left(y_-\right)\right] + \mathrm{G}_2\left(y_-\right)\right]$
6. If NOT $y\left(P^2_{\perp i}\right)_{min} \leq y_i \leq y\left(P^2_{\perp i}\right)_{max}$ return to step 2
7. If NOT $\mathcal{R}_3 < {}^{f\left(P^2_{\perp i}, y_i\right)}\!/_{g\left(P^2_{\perp i}, y_i\right)}$ return to step 2
8. Accept $P^2_{\perp i},\ y_i$

Here $\mathrm{G}_3\left(P^2_\perp\right)$ and $\mathrm{G}_2\left(y\right)$ are the primitive integrals of $g_3\left(P^2_\perp\right) = g_1\left(P^2_\perp\right)\int_{y_-}^{y_+} \mathrm{d}y' g_2\left(y'\right)$ and $g_2\left(y\right)$ respectively, and $\mathcal{R}_{1,2,3} \in [0, 1)$ are uniformly distributed random numbers. The use of the y overestimates from Section 2.5.2 leads to the additional vetoing in

Step 6 compared with the simplest form of the veto algorithm. As is the case with one-dimensional sampling, the efficiency of the algorithm is dependant upon the choice of $g\left(P_{\perp}^{2}, y\right)$, with a better fit to $f\left(P_{\perp}^{2}, y\right)$ resulting in a more efficient algorithm. More on how this algorithm correctly reproduces the branching cross sections and how it was tested is presented in Appendices C and A.4 respectively.

2.6.2 The Veto Algorithm for Competing Branchings

Equation 2.28 for competing branchings can also be sampled using the two-dimensional Veto Algorithm in Section 2.6.1. However, simply substituting:

$$f\left(P_{\perp}^{2}, y\right) = \sum_{k} \frac{\mathrm{d}\sigma_{k}\left(P_{\perp}^{2}, y\right)}{\mathrm{d}P_{\perp}^{2}\mathrm{d}y}, \tag{2.29}$$

into the Veto Algorithm samples the incorrect distribution:

$$\frac{\mathrm{dP}\left(P_{\perp}^{2}, y\right)}{\mathrm{d}P_{\perp}^{2}\mathrm{d}y} = \sum_{k}\left[\frac{\mathrm{d}\sigma_{k}\left(P_{\perp}^{2}, y\right)}{\mathrm{d}P_{\perp}^{2}\mathrm{d}y}\right]\prod_{k}\exp\left\{-\int_{P_{\perp}^{2}}^{P_{\perp max}^{2}}\mathrm{d}P_{\perp}^{2'}\left(\int_{y_{-}\left(P_{\perp}^{2'}\right)}^{y_{+}\left(P_{\perp}^{2'}\right)}\mathrm{d}y'\frac{\mathrm{d}\sigma_{k}\left(P_{\perp}^{2'}, y'\right)}{\mathrm{d}P_{\perp}^{2'}\mathrm{d}y'}\right)\right\}, \tag{2.30}$$

where k runs over the possible branching functions for a given dipole and $y_{\pm}$ are the usual y_{max} and y_{min} rapidity limits.

The correct distribution is recovered by sampling $P_{\perp i}^{2}$ and y_{i} using Equation 2.29 and selecting a branching j with a weight, $\mathrm{W}_{j}\left(P_{\perp}^{2}, y\right)$, given by:

$$\mathrm{W}_{j}\left(P_{\perp}^{2}, y\right) \times \sum_{k}\left[\frac{\mathrm{d}\sigma_{k}\left(P_{\perp}^{2}, y\right)}{\mathrm{d}P_{\perp}^{2}\mathrm{d}y}\right] = \frac{\mathrm{d}\sigma_{j}\left(P_{\perp}^{2}, y\right)}{\mathrm{d}P_{\perp}^{2}\mathrm{d}y}. \tag{2.31}$$

From this it was possible to sample $P_{\perp i}^{2}$, y_{i} and the type of branching for a given dipole, all in one algorithm.

3. PyShower Documentation

A copy of the PyShower simulation code produced for this thesis can be downloaded from https://github.com/leinadao/PyShower.git and is also available via email on request. Currently the program is dedicated towards showering $e^+e^- \rightarrow q\bar{q}$ matrix element results on resonance with the Z-boson channel. This could easily be modified and extended to suit other applications. Further documentation on PyShower is provided in Appendix A.

3.1 Interfacing

PyShower can be interfaced with most other event simulators allowing it to shower matrix elements from multiple sources and output results to be further processed (e.g. hadronised and decayed) using additional software. This is achieved by utilising the Les Houches Event Files (LHEF) format [15], discussed briefly in Appendix D to standardise the input and output format. This necessitates the use of unique particle identification codes (as specified by the Particle Data Group (PDG) [10]), to ensure consistency with other models. Notably it was found that the LHEF values trials, muf2 and mur2 were not needed and therefore they were omitted. Furthermore the variables PDFGUP (1), PDFGUP (2), PDFSUP (1), PDFSUP (2), IDWTUP and NPRUP were not relevant. Default values of 1 were used in their place. Finally, the mother variables, MOTHUP (1) MOTHUP (2), for the partons produced, required the PDG codes of the initial beam electron and positron. As a result, these were implemented instead of the PDG codes of the partons it branched from.

3.2 Design Choices

During the production of PyShower, certain design choices were made. Following the decision to use the CDM ordered by $P_\perp^2$, gluon emissions were modelled. The simulation was then extended through the inclusion of gluon splitting and photon production. However, it was decided that quark-anti-quark annihilation would not be included and photon emission would only be treated between neighbouring $q\bar{q}$ pairs.

Next, the overestimation functions were chosen for sampling the probability distributions discussed in Section 2. Substituting the $P_\perp^2$ definition from Equation 2.15 for the Jacobian, S_{123}, in Equation 2.16, the branching cross sections in Equations 2.17 - 2.21 were each overestimated for use in the modified veto algorithm (see Section 2.6.1) as:

$$g\left(P_\perp^2\right) = \frac{3\alpha_S\left(P_{\perp c}^2\right)}{2\pi P_\perp^2}. \tag{3.1}$$

Separately it was found that Equation 2.4 was consistently overestimated by the function $g\left(\cos\theta\right)$:

$$g = \frac{6\pi {\alpha_{EM}}^2}{5S_{123}}\left[1 + 2g_v^{(e)}g_v^{(f)}\mathrm{Re}\left(\chi\left(S_{123}\right)\right) + \left({g_v^{(e)}}^2 + {g_a^{(e)}}^2\right)\left({g_v^{(f)}}^2 + {g_a^{(f)}}^2\right)\left|\chi\left(S_{123}\right)\right|^2\right]. \tag{3.2}$$

Both results were utilised throughout the simulation and provide moderate efficiency, although less so for photon emission given the use of α_S not α_{EM} in Equation 3.1.

As specified for the kinematics in Section 2.3, one parton from every dipole was chosen to recoil such as to minimise disturbance to the colour flow. The same prescription as used in ARIADNE v4 [4] was adopted. For a $q\bar{q}$ dipole, the probability that a parton, $\widetilde{p}_i$, recoiled was taken to be proportional to the square of its energy [16]:

$$\mathrm{P}\left(\widetilde{p}_{i\ recoil}\right) = \widetilde{E}_i^2 / \left(\widetilde{E}_1^2 + \widetilde{E}_3^2\right). \tag{3.3}$$

Meanwhile, for a qg or $\bar{q}g$ dipole, the gluon was always chosen to recoil, whilst for gg dipoles, the recoiling parton was always the most energetic of the two. With gluon splitting, the extension that both partons in a gg dipole could split was handled by sampling the probability of $gg \to gQ\bar{Q}$ twice. Only if that process was then chosen were the rules above applied.

Occasionally the Lorentz boosts introduced errors large enough (10^{-10}) that PyShower detected the partons were no longer massless within its running precision. This also caused problems when passing the partons to other simulators. To correct for this, the small discrepancy was absorbed in the energy of the most energetic of the offending partons, under the assumption that this would cause the smallest percentage error.

4. Results

4.1 Overview

As coloured partons are not observed in nature due to asymptotic freedom (see Section 2.1); to determine if the shower successfully reproduced the underling physics it was necessary to hadronise and decay the output partons using the existing event simulator PYTHIA v8.2 [5]. Its output was then compared with experiments on resonance with the Z-boson channel at the LEP collider, using the analysis software Rivet v2.4. The massless parton approximation used limited the quark flavours to u, d and s for the $10,000$ events simulated. To help gauge the showers effectiveness, results were also processed for the raw MEs and plotted alongside. A definition required during this analysis is that of thrust, T [17]:

$$T = \frac{\sum_i |\vec{p}_i \cdot \hat{n}|}{\sum_i |\vec{p}_i|}, \tag{4.1}$$

where the thrust axis is then the direction $\hat{n}$ in which T is maximised ($\max_{\hat{n}} T$).

4.2 Tuning

As it is not directly derived from QCD, hadronisation is dependant upon many free variables. Consequently it was necessary to tune the hadronisation algorithms to best suit the PyShower results. The PYTHIA light quark fragmentation variables used were StringZ:aLund, StringZ:bLund and StringPT:sigma for which 2.0, 0.25 and 0.49 respectively were found to produce the best fit with the data. As the results simply aimed to determine if the shower algorithms were running as expected, these was not investigated further.

4.3 Analysis

The shower simulation resulted in gluon and photon production rates of 99.84% and 0.16%, with 3.33% of these gluons splitting, all to 2.d.p. To investigate these, an average e_q^2 was calculated using the hard coded production rate for each quark flavour (in this instance u:26.97%, d:36.69% and s:36.34% all to 2.d.p; see Appendix A.2). This was

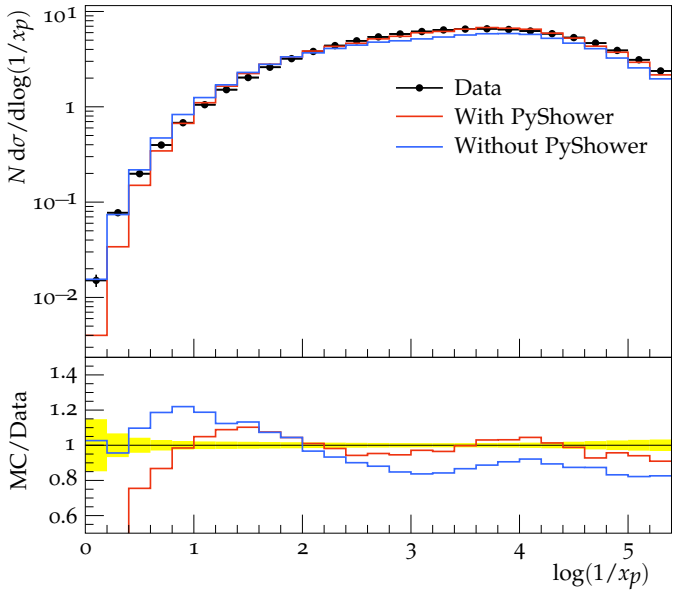

FIGURE 4.1: Comparing PyShower results with data from the DELPHI detector (1996) for the log of the scaled momentum, $\log(1/x_p)$, where $x_p = |p_i| \,/\, |p_{beam}|$.

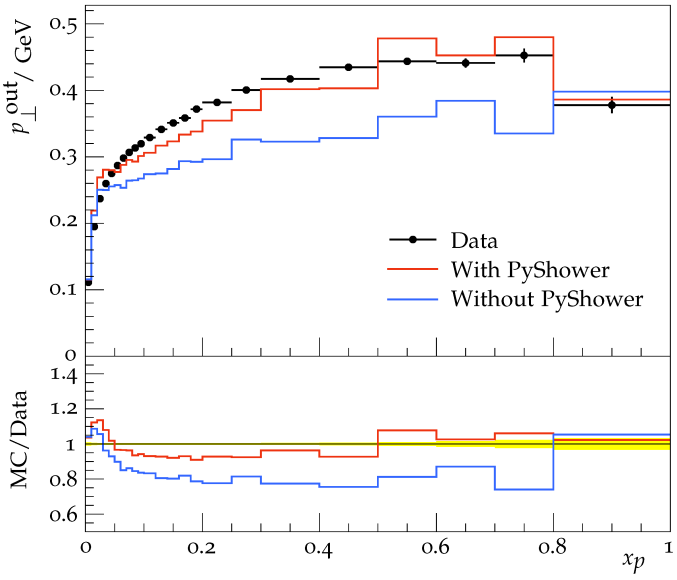

FIGURE 4.2: Comparing PyShower results with data from the DELPHI detector (1996) for the mean out-of-plane transverse momentum, $p_\perp$, (in GeV) with respect to the thrust axes, where $x_p = |p_i| \,/\, |p_{beam}|$.

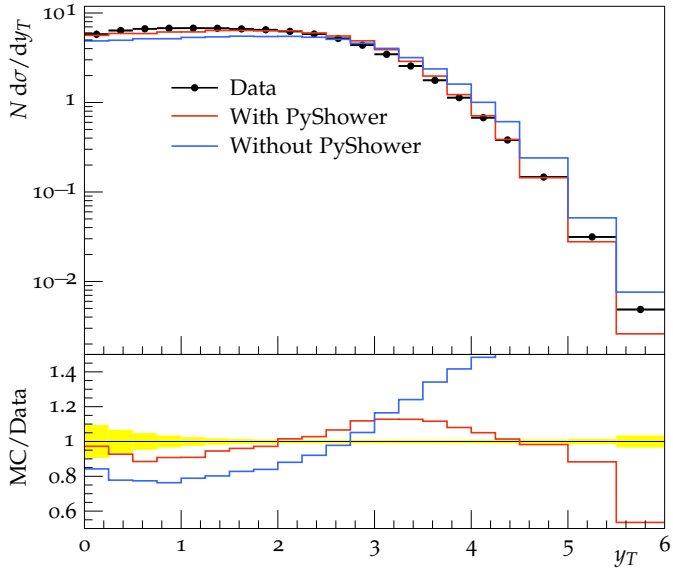

FIGURE 4.3: Comparing PyShower results with data from the DELPHI detector (1996) for rapidity with respect to the thrust axes, y_T.

then used to evaluate the ratio of the prefactor in Equation 2.20 to that in Equation 2.17 at the shower cut-off energy ($P^2_{\perp c} = 1$ GeV), predicting a photon production rate of 0.23% 2.d.p. The corresponding error of approximately 0.07% is attributed to only including emission from $q\bar{q}$ pairs in the same dipole. For a q-g-$\bar{q}$ chain there are four processes competing to branch ($qg \to qQ\bar{Q}$ and $qg \to qgg$ for each of the $qg/g\bar{q}$ dipoles). Approximating each as Equation 2.17, gives a 'first-order' correction of 0.06% (2.d.p).

The average ratio of the gluon splitting to production distributions generated during the showering was 5.13% (2.d.p). However the value presented, the ratio of split gluons to all gluons produced, does not take into account 'secondary' gluons which could not split before the shower ended. Given an average 3 partons were produced per event, it seems reasonable that this may account for the discrepancy. Taking all of this into consideration, the shower appears to be performing as expected when sampling the differential cross sections.

Of the many different analyses performed with LEP data, some of the more promising results are presented here in Figures 4.1 - 4.5, with and without PyShower applied. The yellow highlighted region in the ratio plots corresponds to the error bars on the data. The errors on the PyShower results are to be inferred from these plots.

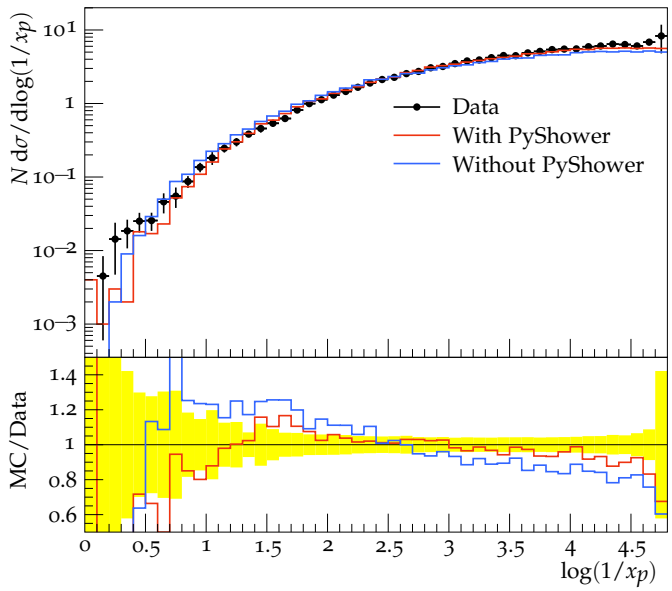

FIGURE 4.4: Comparing PyShower results with data from the ALEPH detector (1996) for the photon spectrum as a function of the log of the scaled momentum, $\log(1/x_p)$, where $x_p = |p_i| \,/\, |p_{beam}|$.

Firstly it is noted that in all of the figures presented, an improved overall fit to the data was obtained when the ME results were showered through PyShower rather than processed on their own, which is a positive indication that the parton shower is filling in some of the physical processes the MEs neglect. Inspecting Figure 4.1 of the resulting scaled momentums shows that whilst there is an approximate error of 10%, the shower results only diverge for higher momentum fractions. As expected, the MEs produce a better fit in that region. Figures 4.2 and 4.3 both display this same characteristic and the drop off at higher rapidity values in Figure 4.3 can equally be attributed to leaving the soft emission regime where the parton shower dominates. Because the parton shower is initialised from a recoiling $q\bar{q}$ pair ($T = 0$), a two-jet structure is expected to envelop the results. Consequently particles along this thrust axis can be considered as evolving from the collinear limit. An agreement of the transverse momentum and rapidity (the shower evolution variables) along the thrust axes then indicates both the shower kinematics and probability distribution sampling are working correctly. In all three plots, the fit of the MEs to the data has an average error of approximately 20%. This implies an improvement of around 10% through utilising the PyShower simulation.

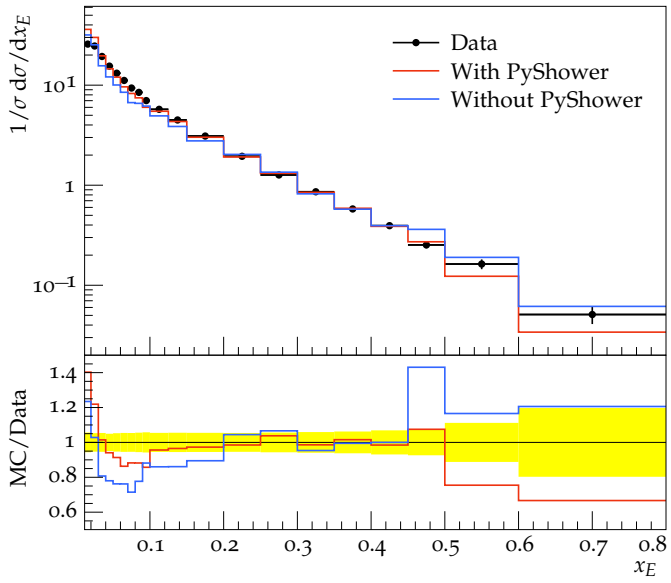

FIGURE 4.5: Comparing PyShower results with data from the OPAL detector (2000) for the scaled momentum of neutral kaons where x_E is the usual energy fraction.

Finally Figures 4.4 and 4.5 for the photon spectrum and neutral kaon scaled momentums are in agreement with the experimental data in the soft emission regions. Again, the MEs alone do not produce this agreement over the same range and the shower results deviate with distance from the soft limit. The quark content of the neutral kaon ($\mathrm{d\bar{s}}$) corresponds to 73.03% (2.d.p) of the quark composition of this analysis, allowing a large percentage of the shower to be analysed together. The agreement in Figure 4.5 also implies that masssless approximation is reasonable for the s-quark, at least in the soft limit.

The average multiplicities of the produced pions, $\pi^{\pm,0}$, were also analysed and found (at 17.12 and 20.73 to 2.d.p) to be in agreement with the data (at 17.16 ± 0.46 20.93 ± 0.26 to 2.d.p). The fit was even better than the kaon result (with an approximate 10% error), most likely due to the lower mass u, d-quark content of the pion. Of the less-favourable analyses not presented here, a large proportion involved the missing heavier quark flavours. It is implied from these that the massless approximation is most likely the greatest contribution to the current deviations from the experimental data.

5. Conclusions

5.1 Concluding Remarks

Overall the results presented in Section 4 indicate that the shower simulation implemented (PyShower) correctly reproduces the physics underlying a parton shower to a reasonable degree of accuracy, once its current limitations are taken into account. The greatest of these limitations being the massless parton approximation used throughout, since not only are the mass effects neglected, but a full analysis of the shower results containing all quark flavours is not possible. It is concluded that this would need to be addressed before other aspects of the shower simulation could be improved.

From the analyses carried out and the testing undertaken during development (see Appendix A) the simulation appears to be robust and functioning in line with the theory in Section 2. The choice of the CDM and $P_{\perp}^2$ as the ordering variable are in line with current developments for parton showers and leave plenty of scope for the simulation to be evolved further.

5.2 Further Development

The natural extension of this project is the simulation beyond the massless approximation, as discussed in Section 5.1. However there are many other interesting avenues which could subsequently be explored. For example, PyShower is written in such a way as to allow the simulation of more exotic initial conditions with only minor modification, such as glueballs (gg and ggg) or $e^+e^- \to W^+W^- \to q\bar{q}q\bar{q}$. The inclusion of quark-anti-quark recombination is an obvious requirement for the first of these suggestions. It would also be possible to extend the simulation to include EM emission from $q\bar{q}$ pairs which do not share the same dipole (although the average error due to this calculated in Section 4.3 is currently much less than that due to the massless approximation). Alternatively it would equally be possible to adapt the EM algorithms to additionally simulate an initial-state shower for e^+e^- collisions, which could be used for more detailed analyses of the processes already covered.

References

[1] U. Pettersson G. Gustafson. *Dipole formulation of QCD cascades*. Nuc. Phys. B 306.4 (1988), pp. 746 –758.

[2] L. Lönnblad B. Andersson G. Gustafson. *The flavouring of a gluon jet*. Phys. Lett. B 236.4 (1990), pp. 461 –465.

[3] L. Lönnblad B. Andersson G. Gustafson. *Gluon splitting in the colour dipole cascades*. Nuc. Phys. B 339.2 (1990), pp. 393 –405.

[4] L. Lönnblad. *ARIADNE version 4: A program for simulation of QCD cascades implementing the colour dipole model*. Comput. Phys. Commun. 71 (Oct. 1992). Revised 2001, pp. 15–31.

[5] T. Sjöstrand *et al*. *An Introduction to PYTHIA 8.2*. Comput. Phys. Commun. 191 (2015), pp. 159–177.

[6] F. M. Krauss. *QCD asymptotic freedom*. Accessed on 14.03.16. Institute for Particle Physics Phenomenology. URL: `https://www.ippp.dur.ac.uk/~krauss/Lectures/QuarksLeptons/QCD/AsymptoticFreedom_1.html`.

[7] L3 Collaboration. *Measurement of the running of the fine-structure constant*. Phys. Lett. B 476.1-2 (Mar. 2000), pp. 40–48.

[8] S Bethke. *Determination of the QCD coupling α_S*. Jour. Phys. G - Nucl. and Part. Phys. 26.7 (July 2000), R27–R66.

[9] B. R. Holstein J. F. Donoghue E. Golowich. *Dynamics of the standard model*. Camb. Monogr. Part. Phys. Nucl. Phys. Cosmol. 2. Cambridge University Press, Oct. 12, 2010. ISBN: 9780521476522.

[10] C. Amsler *et al*. *Review of particle physics*. Phys. Lett. B 667.15 (2008), pp. 1 –6.

[11] F. Krauss W. Jan-Christopher. *Initial-state showering based on colour dipoles connected to incoming parton lines*. JHEP 07 (2008), p. 40.

[12] A. Buckley *et al*. *General-purpose event generators for LHC physics*. Phys. Rept. 504 (2011), pp. 145–233.

[13] P. Z. Skands T. Sjöstrand. *Transverse-momentum-ordered showers and interleaved multiple interactions*. Eur. Phys. J. C39 (2005), pp. 129–154.

[14] P. Skands T. Sjöstrand S. Mrenna. *PYTHIA 6.4 physics and manual*. JHEP 05 (2006), p. 026.

[15] J. Alwall *et al*. *A standard format for Les Houches event files*. Comput. Phys. Commun. 176 (2007), pp. 300–304.

[16] R. Kleiss. *From two to three jets in heavy boson decays: An algorithmic approach*. Phys. Lett. B 180.4 (1986), pp. 400 –405.

[17] J.-M. Grard J.-J. Aubert R. Gastmans. *Particle physics: Ideas and recent developments*. Nato Science Series C: 1. Springer Netherlands, July 26, 1999. ISBN: 978-0-7923-6436-8.

[18] H. van Deurzen. "*Parton Showers in the Colour Dipole Model and the Webs Model*". Masters Thesis. Universiteit van Amsterdam, Nikhef, Theory group, Nov. 19, 2011.

Appendix A

Section 3 provides a brief introduction to PyShower and how it interfaces with external simulation programs along with some of the design choices made. This appendix covers how the simulation should be run as well as how it is written. Tests carried out during the development are also presented in Section A.4.

A.1 Running PyShower

PyShower is written in modular form and is designed to be run from the command line. Matrix element approximation and dipole showering are both run via the corresponding top-level modules, MEs.py and DSs.py. Three low-level modules (precision.py, constants.py and assertions.py) control the core simulation variables which must be modified before the top-level modules are run.

Generating approximate matrix elements requires a command line input of the centre of mass energy squared, the PDG codes of the quark flavours to include, and the number of events desired. Default values are used if these inputs are not specified. The current default configuration is set on resonance with the Z-boson channel and sources quark flavours from the low-level constants.py module. The results are output to the relative path \MEs\dd.mm.yy\filename.lhe which is automatically copied to the clipboard. On completion, a report detailing the relative quark flavour production rates is produced for analysis.

Running PyShower to shower matrix element events also requires a command line input of the quark flavours desired. In this instance, for gluons splitting into a new $Q\bar{Q}$ pair. A default value is sourced as above. The relative path of the matrix elements to load must also be specified. Parton shower results are output to the relative path \ME_DSs\dd.mm.yy\filename.lhe. The simulation automatically iterates over all events in the input file and also generates a report on the produced quark flavours along with the relative occurrance of each cross section event.

A.2 The Simulation Process

The PyShower simulation for MEs randomly samples a $q\bar{q}$ pair flavour, f, according to probabilities that match the hard coded occurrence for each quark, re-weighted for the current flavour set. The hard coded values were derived from their occurrences in the PDG Z-boson decay modes [10]. This flavour is then used to sample Equation 2.4 for an angle θ from the z-axis, utilising the one-dimensional Veto algorithm (see Appendix B). The $q\bar{q}$ partons can then be generated in accordance with f, θ and S_{123}. Finally the e^+e^- pair are also produced in accordance with S_{123}, in opposing directions along the z-axis. It is noted that PyShower uses the accompanying LHEF template to prepare the output results.

For the the parton shower simulation of each event, PyShower loads the matrix element $q\bar{q}$ pair and calculates S_{123}. These partons are then initialised into a dipole which is in turn placed into a chain. Whilst showering is allowed for this event, PyShower iterates over all dipoles in the chain, sampling the differential cross sections in Equations 2.17 - 2.21 to find the winning process for that dipole along with the values of $P^2_\perp$ and y at which it would occur (see Section 2.6.2). In order to maintain the shower ordering in Section 2.4 and evolve the shower globally, the dipole with the highest potential $P^2_\perp$ is allowed to proceed with its winning process. This value of $P^2_\perp$ then sets the maximum value for all subsequent branchings from that chain. Showering is no longer allowed from a chain when all dipoles return a potential value $P^2_{\perp i} \leq P^2_{\perp c}$.

When a dipole is allowed to branch, it is Lorentz boosted into its oriented CMF and the new four-vectors are produced, as per Section 2.3. These three new four-vectors are then returned to the lab frame and the partons have new colours (and particle codes in the case $g \to Q\bar{Q}$) assigned. They are next inserted back into the chain in the form of two new dipoles, which replace the one which branched. Due to the convention used that a gluon appears in two dipoles, once for each of its colour-connections, it is also necessary to update the original gluon copies in neighbouring dipoles. If the chain splits due to the process $g \to Q\bar{Q}$, two separate chains are produced, both with a maximum $P^2_\perp$ defined by the value of $P^2_\perp$ at which the splitting occurred. These then evolve separately in the manner described above.

Finally, once all possible chains (and by extension dipoles) have showered, the resulting dipoles are decomposed back into a list of colour-connected partons, taking care not to double-count gluons. Any photons produced during the shower simulation were stored separately (as they have no colour connections) and are reincorporated as the event is saved to the output LHEF file.

A.3 Coding Style

Because Python is a dynamically typed language, assertion statements were utilised throughout in an attempt to imitate a statically typed language. To improve efficiency, it is possible to run the simulation without these (in Python's optimized mode), but this is not advised. It is noted that name mangling is used as standard in PyShower and all classes contain individual copy functions to produce copies with separate memory pointers. In addition, certain important variables (such as the parton ID codes discussed below) require active confirmation to change.

One coding aim was to enable further development with minimal changes to the existing code structure. To this end, the shower was broken down into related sections which were each individually tested before being complied together. This testing is discussed further in Appendix A.4. A prime example of the extendibility of PyShower is the implementation of the modified Veto algorithm (see Sections 2.6.1 - 2.6.2) in sudakovs.py, where a new evolutionary process could be included simply through the inclusion of a pointer to its probability density function. PyShower already includes provision for many other features thanks to its comprehensive lower-level modules. For example, the tracking of parton production order through unique parton ID codes (separate to the PDG codes) and the value of $P_\perp^2$ at which each was created, allow the reproduction of the simulation visually with only minor additions.

A.4 PyShower Tests

As mentioned in previous sections, constant testing of the simulation was undertaking during its development to ensure each section functioned as expected, before combining them together. The lower-level modules contain specific test code for each of the functions they contain (accessed by running the module from the command

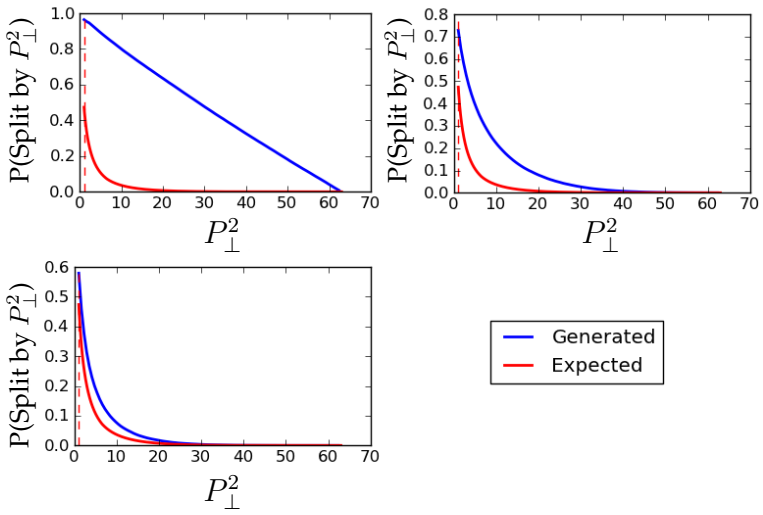

FIGURE A.1: Tests comparing generated $P_\perp^2$ values with the integrated probability distributions expected from Equation 2.17. Later version of the algorithm were implemented from top left to right and then bottom.

line). Higher-level modules were tested as a whole whilst top-level controlling modules were largely tested through detailed analysis during breakdowns of the shower results. A selection of more important test results are presented here. It is noted that some tests (such as those for sampling the cross section probability distributions) are no longer found in the current code due to revisions into a more general format once

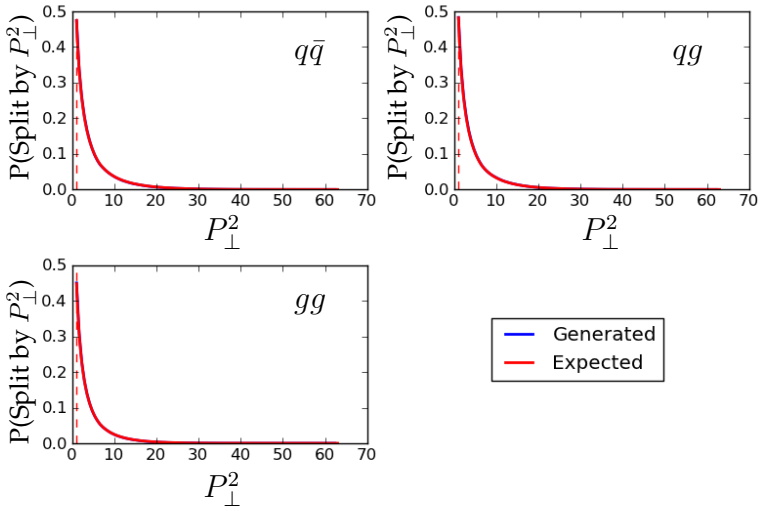

FIGURE A.2: Final gluon emission test results comparing generated $P_\perp^2$ values with the integrated probability distributions expected.

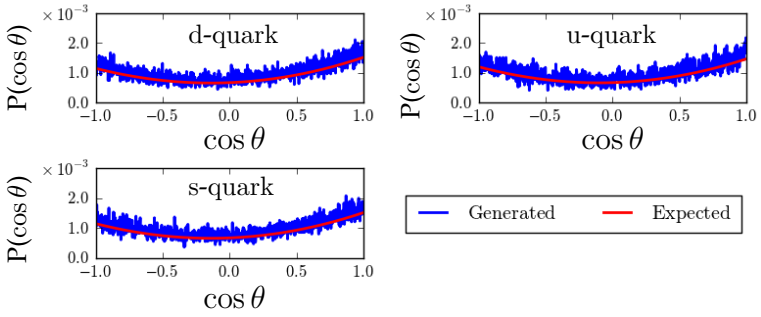

FIGURE A.3: Final quark angle test results comparing generated $\cos\theta$ values with the integrated probability distribution expected.

the tests confirmed the initial functionality.

Examples of the more important tests in lower-level modules include the confirmation that the coupling reference values (e.g. $\alpha_S\left(M_Z^2\right)$) were precisely reproduced by the coupling equations themselves, the testing to exhaustion of the Lorentz transformations to ensure that variables such as S_{123} remained invariant during the transformation of larger numbers of particles, and analysing the conservation of every energy and momentum component and all the invariant masses during the splitting kinematics.

The sampling of the branching probability distributions in Equations 2.17 - 2.19 was first tested before they were combined into one algorithm. Figures A.1 - A.2 show plots of the generated histogram values from sampling the distributions using the modified Veto algorithm presented in Section 2.6.1, alongside the probability distributions integrated over the entire emission phase space (see Section 2.5.2) to give the expected probability distributions. Figure A.1 shows the evolution of the Veto algorithm implemented for $q\bar{q} \to qg\bar{q}$ as several errors were removed, whilst Figure A.2 shows the final results for each of the three gluon emission probability distributions. The probability plotted in these two figures is that of an emission occurring by $P_\perp^2$ in the downward evolution. A similar procedure was implemented to test the sampling of $\cos\theta$ from Equation 2.4 and the final results from these tests are displayed in Figure A.3. These tests were all carried out over $100,000$ iterations.

Finally, as an example of the higher-level module tests, the energy fractions x_3 produced during a run of the simulation are logged, and can be plotted as a function of

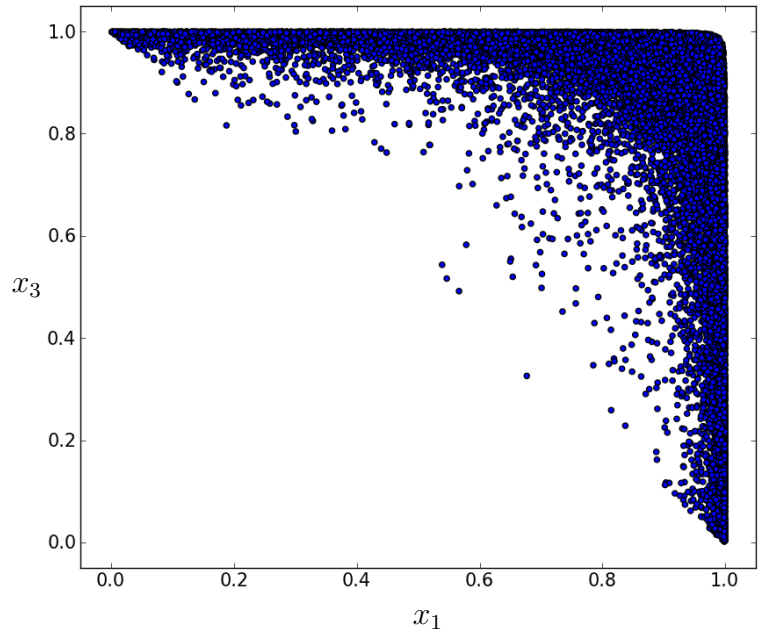

FIGURE A.4: The test plot of the emission phase space parametrised in x_i for the 10,000 events showered in Section 4. Each dot corresponds to an individual branching.

the corresponding energy fraction x_1. The plot corresponding to the shower simulated in Section 4 is presented in Figure A.4, with each individual dot representing a branching that occurred during showering. The top right corner and the top and right edges of the plot correspond to the soft and collinear emissions respectively. It is apparent that these are the dominant areas of the emission phase space and therefore the phase space of the emissions produced matches that expected from Section 2.5.2 for the soft and collinearly dominant parton shower. It can also be seen that the shower kinematics from Section 2.3 are correct.

Appendix B

For a one-dimensional probability distribution, $f(x)$, such as Equation 2.4, ranging from possible values $[x_{min}, x_{max}]$, the probability that a value sampled is in the range $[x_{min}, x]$ is given by:

$$\mathrm{P}\left(x_{min} < x^{'} < x\right) \int_{x_{min}}^{x_{max}} \mathrm{d}x^{'} f\left(x^{'}\right) = \int_{x_{min}}^{x} \mathrm{d}x^{'} f\left(x^{'}\right). \tag{B.1}$$

Rearranging this and replacing the probability with a uniformly distributed random number, $\mathcal{R}_1 \in [0, 1)$, makes it possible to sample x as:

$$x = \mathcal{F}^{-1}\left[\mathcal{R}_1\left[\mathcal{F}\left(x_{max}\right) - \mathcal{F}\left(x_{min}\right)\right] + \mathcal{F}\left(x_{min}\right)\right], \tag{B.2}$$

where $\mathcal{F}(x)$ is the primitive integral of $f(x)$.

If the inverse of the primitive function, $\mathcal{F}^{-1}(x)$, can not be calculated, a simpler over-estimation function $g(x) \geq f(x) \forall x \in [x_{min}, x_{max}]$ can be used whose primitive can be inverted. In this instance, a value sampled using $g(x)$ and Equation B.2 is only accepted if the condition $\mathcal{R}_2 < {}^{f(x)}/_{g(x)}$ is satisfied by a second uniform random number. Multiplying this acceptance probability ${}^{f(x)}/_{g(x)}$ by the first probability sampled, now $\mathrm{P}(x) = g(x)$, correctly reproduces the original distribution $f(x)$ as required, and is the simplest example of the Veto Algorithm [14]. It is clear that the efficiency of this algorithm depends upon as close a fit of the over-estimated function to the original distribution as possible.

Appendix C

In order to give further insight into how the Veto algorithm used in Section 2.6.1 correctly reproduces the relevant cross sections from Section 2.5.1, the unitarity of the one-dimensional case is examined here using [18]. Recalling the Sudakov form factor, $\Delta\left(P_{\perp}^{2}\right)$ as the no-emission probability from Equation 2.25, the probability of no emission over the entire phase space is:

$$\mathrm{P}_0 = \exp\left\{-\int_0^{P_{\perp max}^2} \mathrm{d}P_{\perp}^{2'}\,\sigma\left(P_{\perp}^{2'}\right)\right\}. \tag{C.1}$$

Multiplying Equation 2.26 by the probability than no second emission occurs below $P_{\perp}^{2}$ and integrating over the phase space gives the probability of only one branching occurring:

$$\begin{aligned}\mathrm{P}_1 &= \int_0^{P_{\perp max}^2} \mathrm{d}P_{\perp}^{2}\left[\exp\left\{-\int_{P_{\perp}^2}^{P_{\perp max}^2} \mathrm{d}P_{\perp}^{2'}\,\sigma\left(P_{\perp}^{2'}\right)\right\}\sigma\left(P_{\perp}^{2}\right)\exp\left\{-\int_0^{P_{\perp}^2} \mathrm{d}P_{\perp}^{2'}\,\sigma\left(P_{\perp}^{2'}\right)\right\}\right], \\ &= \int_0^{P_{\perp max}^2} \mathrm{d}P_{\perp}^{2}\left[\sigma\left(P_{\perp}^{2}\right)\exp\left\{-\int_0^{P_{\perp max}^2} \mathrm{d}P_{\perp}^{2'}\,\sigma\left(P_{\perp}^{2'}\right)\right\}\right].\end{aligned} \tag{C.2}$$

Using the same strategy as for one branching, the probability of two branchings occurring requires an additional Sudakov and σ term and another integration over all values of $P_{\perp}^{2}$ the second branching might occur at. Simplifying by combining the exponentials gives:

$$\begin{aligned}\mathrm{P}_2 &= \int_0^{P_{\perp max}^2} \mathrm{d}P_{\perp(1)}^{2}\int_0^{P_{\perp(1)}^2} \mathrm{d}P_{\perp(2)}^{2}\left[\sigma\left(P_{\perp(1)}^{2}\right)\sigma\left(P_{\perp(2)}^{2}\right)\exp\left\{-\int_0^{P_{\perp max}^2} \mathrm{d}P_{\perp}^{2'}\,\sigma\left(P_{\perp}^{2'}\right)\right\}\right], \\ &= \int_0^{P_{\perp max}^2} \mathrm{d}P_{\perp(2)}^{2}\int_{P_{\perp(2)}^2}^{P_{\perp max}^2} \mathrm{d}P_{\perp(1)}^{2}\left[\sigma\left(P_{\perp(1)}^{2}\right)\sigma\left(P_{\perp(2)}^{2}\right)\exp\left\{-\int_0^{P_{\perp max}^2} \mathrm{d}P_{\perp}^{2'}\,\sigma\left(P_{\perp}^{2'}\right)\right\}\right].\end{aligned} \tag{C.3}$$

Combining the two forms of Equation C.3 with a factor of $1/2!$ to compensate for over-counting, the probability of two emission can be re-written as:

$$\begin{aligned}
\mathrm{P}_2 &= \frac{1}{2!}\left[\int_0^{P^2_{\perp max}} \mathrm{d}P^2_{\perp(1)} \int_0^{P^2_{\perp(1)}} \mathrm{d}P^2_{\perp(2)} \sigma\left(P^2_{\perp(1)}\right) \sigma\left(P^2_{\perp(2)}\right)\right. \\
&\left. + \int_0^{P^2_{\perp max}} \mathrm{d}P^2_{\perp(2)} \int_{P^2_{\perp(2)}}^{P^2_{\perp max}} \mathrm{d}P^2_{\perp(1)} \sigma\left(P^2_{\perp(2)}\right) \sigma\left(P^2_{\perp(1)}\right)\right] \exp\left\{-\int_0^{P^2_{\perp max}} \mathrm{d}P^{2'}_{\perp} \sigma\left(P^{2'}_{\perp}\right)\right\}, \\
&= \frac{1}{2!}\left(\int_0^{P^2_{\perp max}} \mathrm{d}P^2_{\perp} \sigma\left(P^2_{\perp}\right)\right)^2 \exp\left\{-\int_0^{P^2_{\perp max}} \mathrm{d}P^{2'}_{\perp} \sigma\left(P^{2'}_{\perp}\right)\right\}.
\end{aligned} \tag{C.4}$$

Comparing this final form with that of zero and one branchings the generalisation to n branchings can be see to be:

$$\mathrm{P}_n = \frac{1}{n!}\left(\int_0^{P^2_{\perp max}} \mathrm{d}P^2_{\perp} \sigma\left(P^2_{\perp}\right)\right)^n \exp\left\{-\int_0^{P^2_{\perp max}} \mathrm{d}P^{2'}_{\perp} \sigma\left(P^{2'}_{\perp}\right)\right\}. \tag{C.5}$$

Finally summing over all possible number of branchings gives:

$$\begin{aligned}
\sum_{n=0}^{\infty} \mathrm{P}_n &= \sum_{n=0}^{\infty} \frac{1}{n!}\left(\int_0^{P^2_{\perp max}} \mathrm{d}P^2_{\perp} \sigma\left(P^2_{\perp}\right)\right)^n \exp\left\{-\int_0^{P^2_{\perp max}} \mathrm{d}P^{2'}_{\perp} \sigma\left(P^{2'}_{\perp}\right)\right\}, \\
&= \exp\left\{\int_0^{P^2_{\perp max}} \mathrm{d}P^{2'}_{\perp} \sigma\left(P^{2'}_{\perp}\right)\right\} \exp\left\{-\int_0^{P^2_{\perp max}} \mathrm{d}P^{2'}_{\perp} \sigma\left(P^{2'}_{\perp}\right)\right\}, \\
&= 1,
\end{aligned} \tag{C.6}$$

which proves unitarity is achieved through this method of sampling. Notably the ending of the shower at $P^2_{\perp c}$, as discussed in Section 2.5.2, was replaced by a continuation down to $P^2_{\perp} = 0$ in this proof. This is due to the existence of a non-zero probability for branching below the cut-off value, which itself is simply included as a limit in parton showering to signify the breakdown of the QCD treatment, and the beginning of hadronisation (see Sections 2.5.2 and 1.3.5).

Appendix D

As discussed in Section 3.1, the LHEF format [15] was utilised to ensure easy integration with external simulators and is discussed in more detail here. The XML format was designed for holding parton-level results and is therefore perfect for both input and output in PyShower. The format is shown below with the use of the official variable names and $\cdots$ to symbolise the continuation of one line:

<LesHouchesEventsversion = 1.0>
<header>
<!−−
Additional program specific information here.
−−>
</header>
<init>
IDBMUP (1) IDBMUP (2) EBMUP (1) EBMUP (2) PDFGUP (1) $\cdots$
PDFGUP (2) PDFSUP (1) PDFSUP (2) IDWTUP NPRUP
XSECUP (IPR) XERRUP (IPR) XMAXUP (IPR) LPRUP (IPR)
</init>
<event>
NUP IDPRUP XWGTUP SCALUP AQEDUP AQEDUP
IDUP ISTUP MOTHUP (1) MOTHUP (2) ICOLUP (1) ICOLUP (2) $\cdots$
PUP (1) PUP (2) PUP (3) PUP (4) PUP (5) VTIMUP SPINUP
</event>
</LesHouchesEvents>

The initialisation section describes the type of events included in the file. Only the first four of these variables were utilised, with the remainder set to their default values. The PDG codes of the initial beam electron and positron were assigned to IDBMUP (1) and IDBMUP (2) and their energies to EBMUP (1) and EBMUP (2). The event section is next, replicated once for each individual event produced. Its top line contains information on that event, of which only the number of particles, NUP, was utilised. The next line is then repeated once for each of the NUP particles in that event. In order of appearance the variables are: PDG code, status, mother 1 and 2, colour, anti-colour, (p_x, p_y, p_z, E), mass, proper lifetime and spin. The particle status was always set to -1 (initial-state) for the electron and positron and 1 (final-state) for all partons produced during the shower.

An example LHEF file of approximated matrix element results generated and then showered by PyShower is shown below. All but one of the 100000 events have been removed in this instance:

```
<LesHouchesEventsversion = 1.0>
<header>
<!--
##Matrix elements approximated by PyShower (Daniel Osborne 2015/16 − Durham University)
##numEvents = 100000, S123 = 8315.251344, qCodes = [1, 2, 3]
##Showered by PyShower (Daniel Osborne 2015/16 − Durham University)
##ME LHEF source file : \MEs\11.04.16\qqBar_100000MEs_at14.07.47.lhe
-->
</header>
<init>
11   -11   45.6   45.6   1   1   1   1   1   1
1    1     1     1
</init>
<event>
7   0   1   0.0   0.0   0.0
11    -1   0   0   0     0     0.00e+00   0.00e+00   4.56e+01   4.56e+01   0.00e+00   0.0   0.5
-11   -1   0   0   0     0     0.00e+00   0.00e+00  −4.56e+01   4.56e+01   0.00e+00   0.0   0.5
2     1    1   2   501   0     1.28e+01  −2.20e+01   1.98e+00   2.56e+01   0.00e+00   0.0   0.5
21    1    1   2   504   501   4.28e+00  −6.00e+00   1.26e+00   7.48e+00   0.00e+00   0.0   1.0
21    1    1   2   503   504   2.23e+00  −1.45e+00   1.02e−01   2.67e+00   0.00e+00   0.0   1.0
-2    1    1   2   0     503   5.85e−01  −3.07e+00   3.86e+00   4.97e+00   0.00e+00   0.0   0.5
2     1    1   2   502   0    −4.80e+00   6.83e+00  −1.25e+00   8.43e+00   0.00e+00   0.0   0.5
-2    1    1   2   0     502  −1.92e+01   2.98e+01  −7.20e+00   3.62e+01   0.00e+00   0.0   0.5
22    1    1   2   0     0     4.04e+00  −4.18e+00   1.25e+00   5.95e+00   0.00e+00   0.0   1.0
</event>
<event>
...
</event>
</LesHouchesEvents>
```

In this example the four-momentum, energy and mass values have been shortened to two decimal places from the default ten usually used. The utilisation of the extra space in the header for personalised variables can also be seen; PyShower stores the user input values given when generating matrix elements and the input LHEF filename used when showering.