

Improvements to the treatment of gluons within QCD-CR Junctions

Thomas Garnett

Supervisor: Professor Peter Skands

Date: November 13, 2022

ABSTRACT: The mechanisms of hadronisation comprise one of the most important research areas within modern particle physics, and yet their exact formulation is still not fully developed. This report will discuss hadronisation in terms of the Lund String model, and provide an overview on the physics behind how this is employed in the QCD-CR model of the code library PYTHIA 8. This model's method of utilising string junctions to predict baryon production requires further refinement to produce reliable results, as its treatment of gluons through its pull-vector algorithm can lead to the formation of incorrect junction system configurations. To rectify this, a new scheme is introduced in this report, the iterative-removal algorithm. This report will compare and analyse the behaviour of the new algorithm with respect to the old, and then conclude with some preliminary particle production spectra generated from the new algorithm. These will be discussed in reference to experimental data collected at the LHC.

Contents

1	Introduction	1
2	Fundamentals of QCD	2
2.1	The quark level	2
2.2	Colour	3
2.3	Confinement and hadrons	4
3	The Lund String Model	5
3.1	The Lund String	5
3.2	The yoyo model	5
3.3	String Fragmentation	8
3.4	The Fragmentation Function	9
4	Baryon production	10
4.1	Simple mechanisms	10
4.2	The junction	10
4.3	Junction motion and orientation	12
4.4	The LC limit and Colour Reconnections	13
4.5	Beyond the LC limit	14
5	Treatment of gluons	15
5.1	The Pull-vector	15
5.2	The current method for reaching the Junction Rest Frame	17
5.3	Problems with JRF convergence	18
5.3.1	The angle measure	19
5.3.2	The soft gluons	21
5.4	The iterative-removal algorithm	22
6	Comparison of Old and New-scheme events	24
6.1	The Dalitz plot	24
6.2	Plot generation method	24
6.2.1	The starting momenta configuration	25
6.2.2	The values of the Dalitz plot	25
6.3	Compendium of test cases without gluons	26
6.4	Compendium of test cases containing gluons	28
6.4.1	Gluon energy variation	28
6.4.2	Gluon position variation	32
7	Predictions from the New Model	34
7.1	Current successes of the QCD-CR model	34
7.1.1	CR introduction	34
7.1.2	Generator parameters	34
7.1.3	Current progress	36

7.2	New scheme modelling	37
7.2.1	Mode 2 predictions	37
7.2.2	Comparison of modes	40
7.3	Additional predictions: Other hadron spectra	41
7.4	Multiplicity dependence	42
8	Conclusion	45
9	Acknowledgements	46
A	Parameter tunings of each mode	48
B	Coding syntax of old and new algorithms	48

1 Introduction

Much of the theoretical initiative behind developments in modern physics concerns an impetus to reduce observed phenomena to their most fundamental constituents. This can be seen in the lengthy timeline of discoveries in quantum physics that have slowly revealed successively smaller ingredients of matter; from the discovery of the atom, to the nucleon, to the quark, and to even smaller potential, unproven, theoretical models such as quantum strings. Equally important to these discoveries, however, is the knowledge of the mechanism by which these fundamental “building blocks” of matter combine to form the more familiar phenomena we have observed. In particular, the move from quarks to the more well known protons and neutrons that quarks combine to create, remains an area of active research in the scientific literature.

The more precise nature of quarks will be explored in Chapter 2, however, broadly, they are subatomic particles that combine to form multi-quark structures called hadrons. The quark model was independently formulated in 1964 by Gell-Mann and Zweig [1, 2] but only experimentally confirmed in 1974 after discovery of the charm quark [3]. The precise method of how quarks actually form into hadrons, called *hadronisation*, exists currently as several proposed theoretical processes.

The mechanism underpinning the work within this report is the Lund String model. This model considers the inter-quark interactions in terms of the potentials that arise between the particles. These potentials are theoretically represented by classical strings stretching between two quarks. Given enough energy, these strings can “break,” producing new quark pairs, which can form hadrons [4]. From the Lund String model we will see a straightforward and intuitive mechanism for the production of baryons, particles which are frequently obtained from particle collisions. Comparing the predictions from the theoretical model to the production rates found in experiment independently verifies the success of the Lund String model.

Predictions from the Lund String model are made possible through the PYTHIA 8 event generator. This is a program that stochastically simulates sequences of events from various physical processes with specific parameters. An event corresponds to a “snapshot” of a number of outgoing particles, as might be recorded experimentally by a detector. As the total probability distributions of final states in processes like particle collisions are often very complicated, they cannot be expressed analytically.

Instead, PYTHIA models them using numerical methods, using Markov Chain Monte Carlo (MCMC) techniques [5].

Importantly, PYTHIA’s MCMC algorithm differs from other similar programs such as HERWIG [6] or SHERPA [7] by allowing the possibility for inter-parton interactions through colour reconnections (CR). The theory underpinning CR will be discussed in Chapter 4, however its implementation into PYTHIA greatly affects the final distribution of hadrons predicted from the generator [8]. As this model explains interactions within quantum chromodynamics (QCD), discussed in the next chapter, it is termed the QCD-CR model.

The main focus of this report is on the improvement of the numerical algorithms within PYTHIA 8. Currently, PYTHIA’s utilisation of string junctions as a model for hadronisation encounters issues when the junction systems also contain gluons. A refinement is needed to the existing algorithm, and the development and implementation of a new scheme for the treatment of gluons, the iterative-removal algorithm, is shown in detail in this report. Overall, the work in this report will generate new insights as to the mechanisms behind baryon production, and hopefully seek to further the complete picture of hadronisation as provided by the Lund String Model.

This report will be divided into two major sections. Firstly, chapters 2-4 introduce the required theory underpinning the QCD-CR model, starting with a brief discussion of QCD, then explaining the Lund String Model, and then extending these concepts to string junctions and colour reconnections. The second section will explain the old and new gluon-treatment models within PYTHIA, and then present preliminary results obtained from the new algorithm. Chapter 6 will analyse its behaviour in a variety of specific test cases, and chapter 7 will conclude with the analysis of new particle production spectra generated in PYTHIA. These will be discussed in context with recent measurements made at the LHC.

2 Fundamentals of QCD

2.1 The quark level

Particle collisions are instrumental in illuminating new physics at small length scales, and at high energies. Analysing the products formed from the collision of two particles can reveal previously unknown information about their more fundamental substructure.

Taking electron-proton ($e^-p \rightarrow e^-p$) scattering, at low energies (the elastic limit), the two particles simply exchange energy and momentum, but otherwise remain unchanged. However, at high energies, what is known as deep inelastic scattering “breaks up” the proton and reveals a complex sea of interacting sub-particles within it [9]. A diagram comparing the $e^-p \rightarrow e^-p$ scattering products is given in Figure 1.

These point-like particles within the proton are the previously mentioned quarks, as well as the gluons, the latter of which will be defined later in this section. Quarks constitute some of the most fundamental building blocks of matter, at least above a level where effects of quantum gravity become significant [10]. They most famously bind together to form protons (2 up-quarks and 1 down), and neutrons (1 up and 2 down), but can combine to form a number of different hadrons, as will be seen later in this chapter. More generally, point-like objects that combine to form larger hadronic structures are referred to as “partons”: This term also encompasses anti-quarks and gluons.

Quarks come in six varieties, known as “flavours,” differentiated primarily by their mass. Each of these is shown below. The mass values are given in natural units, which are used throughout this report.

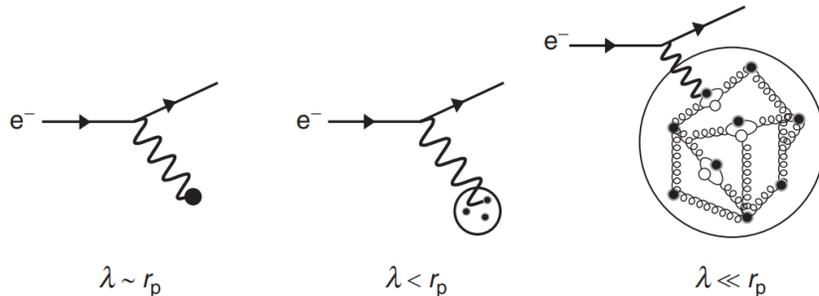


Figure 1. Three diagrams showing the resultant $e^-p \rightarrow e^-p$ collision products at progressively higher energies (quantified by comparing the virtual photon wavelength λ to the proton radius r_p). Note how the higher the energy the more the proton breaks up. Figure reproduced from [9].

Flavour	Mass(GeV)
Up (u)	0.33
Down (d)	0.33
Strange (s)	0.5
Charm (c)	1.5
Bottom (b)	4.8
Top (t)	$172.13^{+0.76}_{-0.77}$

Table 1. Each quark flavour, with corresponding constituent mass as used in PYTHIA 8 [11]. Top quark mass obtained from [12].

Table 1 displays the constituent mass of each quark, which considers the quark mass as a fraction of the particle structure it is contained within. For example, the constituent mass of the up quark is 1/3 of the proton mass, as the proton contains two up quarks and one down quark. These quark mass values were tabulated as they are the specific values used in PYTHIA [11].¹

Generally, the u , d and s quarks are referred to as the “light” quarks and the c , b and t quarks the “heavy” quarks. Hadrons containing the heavy quarks can be also referred to as “heavy.”

Finally, the additional structures seen in the far-right panel of Figure 1 that appear as spring-like chains joining the quarks are *gluons*. Gluons are the quark-level equivalent of photons: while photons carry the electromagnetic force between interacting particles (for example, electrons), gluons carry the force that mediates interacting quarks, the strong force [10]. We shall see some consequences of quarks interacting with each other like this in the later parts of this chapter.

2.2 Colour

Particles feel the electromagnetic force due to their charge, an interaction governed by Quantum Electrodynamics (QED). For example, two negatively charged electrons will repel one another, whereas an electron and a positively charged positron will attract one another. This interaction is mathemat-

¹The mass of the top quark as used in PYTHIA, at the time of writing this report, is given as $m_T = 171.0 \pm 1.4$ GeV. This is a relatively outdated value, however, which has been superseded by many more recent experiments. Thus, for the sake of providing as accurate a tabulation as possible, the value of the top quark mass included in Table 1 is obtained from recent measurements made by CMS in 2021 [12].

ically straightforward; it is governed by the (single type of) photon, which corresponds to the single generator of the U(1) local gauge symmetry [9]. Note the U(1) group corresponds to the set of all complex numbers with absolute value 1, under multiplication.

Quarks have an additional “charge,” but its structure has a much more complicated formalism. Quark interactions are governed by eight (different types of) gluons, which correspond to eight matrix generators of an SU(3) symmetry group. This is a group containing unitary 3×3 matrices with determinant 1 [10].

We represent these interactions by defining the “colour” charge of quarks, hence the name “Quantum Chromodynamics”, or QCD. Each SU(3) matrix generator is 3×3 , which allows any particle within QCD three unique colour states. These are designated as the quark colours: Red (r), green (g) and blue (b). Only quarks that possess a colour charge couple with gluons and thus interact through the strong force. Similarly to QED, QCD also requires anti-quarks, which possess anti-colour: Anti-red (\bar{r}), anti-green (\bar{g}) and anti-blue (\bar{b}).

Like electric charge, colour charge is conserved in quark-to-quark interactions. For this to be upheld, however, gluons must also carry a combination of colour charge and anti-colour charge. For example, the scattering process $br \rightarrow r\bar{b}$ necessitates the exchange of a $r\bar{b}$ or a $b\bar{r}$ gluon depending on the time-ordering of the process. As gluons are the generators of the aforementioned colour SU(3) symmetry group, it follows that there are eight different colour/anti-colour gluon configurations [9].

2.3 Confinement and hadrons

A fundamental part of the quark model is that “free” quarks, ones that are not influenced by other quarks and simply propagate through spacetime on their own, have not been observed in nature. Instead, the theory of *colour confinement* is proposed (although has not been analytically proven) [9]. Confinement suggests that quarks will only appear in colour-singlet states, or specifically a multi-quark state with no net colour charge.

The intuition for the reasoning behind confinement can be seen by considering the colour-charge potential that arises between two arbitrary quarks. The quarks interact through gluons, which, as they also carry colour charge, allows attractive interactions between the gluons. This in turn shapes the colour field potential lines between the quarks into a sort of “tube,” as opposed to the spreading out of the field lines in QED interactions.

The shape of the QCD field means that at large distances the potential is approximately given by a linear relationship [13]:

$$V(r) \rightarrow \kappa r, r \rightarrow \infty . \tag{2.1}$$

This term forms part of what is known as the Cornell potential, and its implications will be discussed in more detail in the following section. With reference to this chapter, equation (2.1) suggests that attempting to separate two quarks an infinite distance apart would require an infinite amount of energy. This is of course energetically unfavourable, so quarks will instead prefer to arrange themselves into objects with no net colour field between them [9].

Confinement gives rise to hadrons, multi-quark bound states that encompass the proton, neutron and many other composite structures. *Mesons* usually refer to two-quark states consisting of a quark and antiquark (e.g. $u\bar{d}$). *Baryons* usually refer to three-quark states (e.g. uud) [10]. More exotic quark states like tetraquarks and pentaquarks have been observed but are not relevant for the analysis within this report.

It is clear that understanding of the confinement process will be instrumental in more comprehensively understanding the methods by which hadronisation occurs. In the subsequent chapter, I will introduce a phenomenological model that provides a direct mechanism for the formation of additional hadrons from interactions between partons.

3 The Lund String Model

3.1 The Lund String

More generally, the Cornell confinement potential mentioned previously comprises two distinct regimes. The first is the near (small r) ‘‘Coulomb-like’’ regime [13], where the potential V tends to the expression

$$V(r) \rightarrow \frac{-4\alpha_S}{3r}, r \rightarrow 0. \quad (3.1)$$

Here α_S is the strong coupling constant. Precise determination of the value of α_S is quite complicated and beyond the scope of this report. Broadly, it varies from ~ 0.1 to 1 based on the current energy and distance scale [14]. Irrespective of the value of α_S , the singular behaviour of V is evident as $r \rightarrow 0$. The second potential regime is the aforementioned far ‘‘linear’’ regime, where V is given by equation (2.1) in Chapter 2. Clearly, the contribution of the Coulomb regime to the total V is negligible at large r in comparison to the linear regime [13].

A value of $\kappa \approx 1$ GeV/fm was first determined from hadron spectroscopy [15], so the contribution from equation (2.1) is only significant compared to the Coulomb regime at distance scales larger than one femtometre.

For $r > 1$ fm, an intuitive method of describing quark-antiquark interactions is through the Lund String model. As introduced in Chapter 2.3, quarks interact with other quarks through a confining colour field, which corresponds to an exchange of any number of intermediate gluons between the particles [16]. As the gluon-interactions force the potential lines into a tube-like shape, the inter-quark potential can be naturally analogised to a string-like artefact. This region of the potential is what is represented by a Lund string.

In its simplest case, a Lund String is a massless, relativistic string representing a gluonic flux tube running from one quark endpoint to an antiquark endpoint; it parameterises a uniform width cylindrical region along the length between the endpoints [15, 17]. The strings can be readily analogised to a vortex line in a superconductor [16].

A basic diagram of a Lund String is shown in Figure 2. Represented in this Figure is the resultant vector field obtained from the Cornell potential, generated from a quark and an anti-quark positioned 5 femtometres apart. The field is derived from the equivalent relation from classical electrostatics:

$$\mathbf{E} = -\nabla V. \quad (3.2)$$

In this case of course V refers to the Cornell potential, and E the corresponding colour-charge field.

After generating this field, the higher-magnitude bar-like artefact between the two pseudo-charges is clearly visible in Figure 2. This corresponds to the region of the potential modelled by the Lund String.

3.2 The yoyo model

As the two particles propagate through spacetime, each is subject to a potential due to the opposing particle, and they exchange energy. From equation (2.1) at this length scale the potential between the

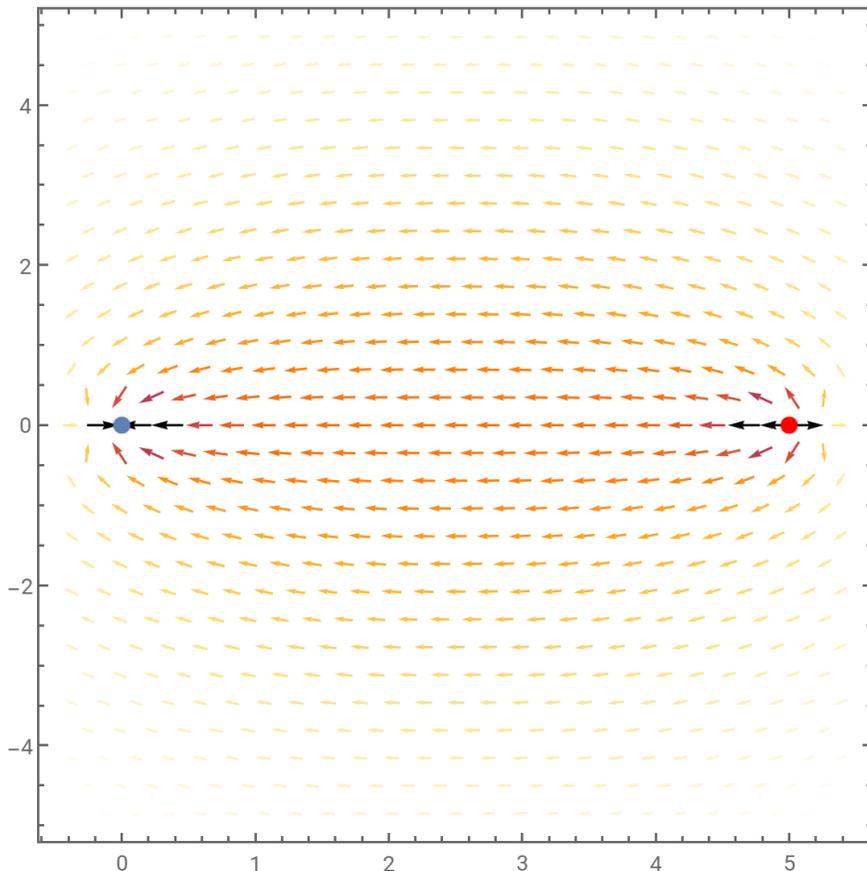


Figure 2. Vector plot of field from (total) Cornell potential from a quark-antiquark pair, made in Wolfram Mathematica. Vector arrow colours indicate magnitude of the field strength. Each axis is in units of femtometre. The blue circle on the left marks the quark location, and the red on the right the anti-quark. This plot used $\alpha_s = 0.1184$ as the value of the strong coupling constant [14]. Choice of $\alpha_s \in [0.1, 1]$ will not visibly affect behaviour of vector field.

particles is approximately linear ($V \approx \kappa r$). Thus, the potential functions effectively as a string with string tension (energy per unit length) of $\kappa \approx 1$ GeV/fm [16]. As this system lacks any transverse degrees of freedom, it can be represented by the Hamiltonian [15]

$$H = E_q + E_{\bar{q}} + \kappa |z_q - z_{\bar{q}}| , \quad (3.3)$$

where $|z_q - z_{\bar{q}}|$ is the absolute distance between the particles. In the energy-momentum picture, this lets us derive the linear expression for the rate of change of the particles' energy with respect to time:

$$\left| \frac{dE_{q/\bar{q}}}{dt} \right| = \kappa , \quad (3.4)$$

where the sign of the derivative changes at every “vertex” (change of direction) in Figure 3. $E_{q/\bar{q}}$ specifies the energy of either the q or \bar{q} . [15]

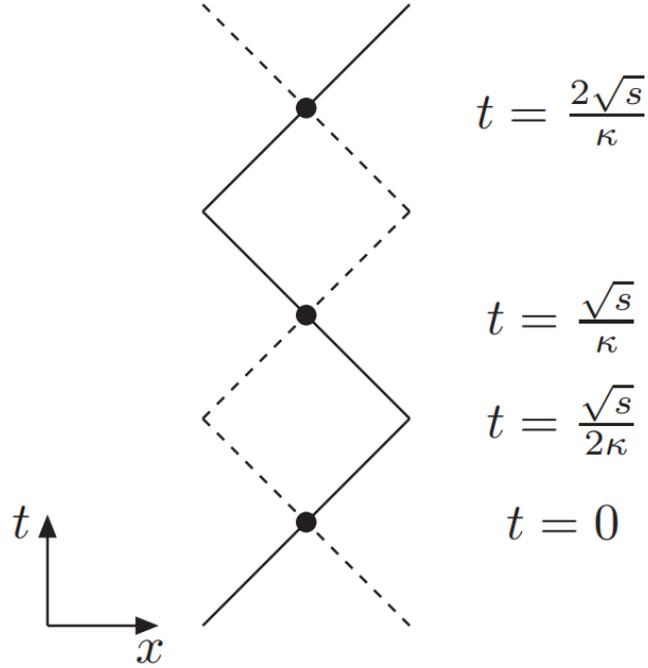


Figure 3. Schematic of the paths of the $q\bar{q}$ pair in the yoyo model with key points. Quark is shown with a solid line, antiquark with a dashed line. Taken from [5].

Note that in order to more easily understand the behaviour of the yoyo model, we are considering here a simplified picture where the quark and anti-quark are massless. If mass was accounted for, the paths of the two particles in Figure 3 would lose their “sharp” vertices and exhibit a structure visibly akin to simple harmonic motion. As the particles move further apart, more energy is transferred to the string, until it reaches its maximal extension at $t = \frac{\sqrt{s}}{2\kappa}$, as shown in Figure 3. Here \sqrt{s} is the centre-of-mass energy. At this point the quark and antiquark have no energy (as we are considering massless quarks) because all of it has been transferred to the string:

$$(E, P_x)_q = (0, 0), E_{str} = \sqrt{s} . \quad (3.5)$$

The other extremal point(s) in Figure 3 is where all energy has been transferred from the string to the particles, at $t = 0, \frac{\sqrt{s}}{\kappa}, \dots$ [5]:

$$(E, P_x)_q = \left(\frac{\sqrt{s}}{2}, \pm \frac{\sqrt{s}}{2} \right), E_{str} = 0 . \quad (3.6)$$

Assuming no perturbing outside influence, this process thus propagates in a similar manner to that of a classical pendulum (although with a notably non-classical linear potential), with extrema at the points of maximum potential or kinetic energy. This is what gives rise to the yoyo-like motion shown in Figure 3.

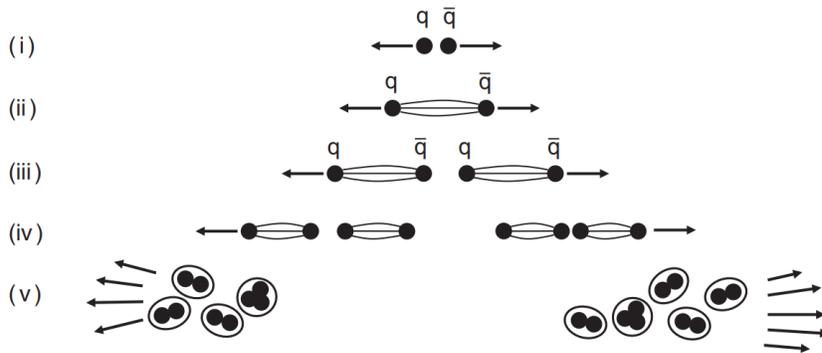


Figure 4. Diagram of five sequential stages of hadron formation from string breaking: i) The initial $q\bar{q}$ pair, ii) The formation of the Lund String at sufficient distance, iii) The breaking of the string into a new $q\bar{q}$ pair, iv) The subsequent repetitions of this process, v) The possible final-state organisation of the $q\bar{q}$ pairs into hadrons. Figure reproduced from [9].

3.3 String Fragmentation

If the Lund string of the $q\bar{q}$ pair has sufficient energy, it can become energetically favourable for virtual particles to quantum tunnel through the potential generated from the string, and create new, real (on-shell) particles [4]. Importantly, for these new particles to create a confining $q\bar{q}$ pair, they must have the same colour; thus this process functions in “colour screening” the particles created from the Lund Strings.

This process can be seen as a QCD analogue of the Schwinger mechanism of electron-positron production. The Schwinger mechanism is a hypothetical process where the electron and positron are generated from fluctuations of the vacuum in the presence of a strong electromagnetic field [18].

When a $q\bar{q}$ pair is able to tunnel out from the colour field, this can be modelled as the Lund string breaking at this point in the field to create new particles. This process is shown in steps i)-iii) in Figure 4. As the newly created particles are a $q\bar{q}$ pair, this process will only occur if the potential generated in the Lund string exceeds that of the rest-mass energy of the lightest meson, the π^0 meson.

As the new quarks similarly generate new colour tubes between their nearest neighbours, the process can repeat with subsequent breaks within the newly fragmented Lund string pieces, thus generating further $q\bar{q}$ pairs. This process can continue indefinitely until the string pieces do not possess sufficient energy to generate quark-antiquark pairs. Once the $q\bar{q}$ pairs are created, the resultant particles can combine to form hadrons, as in the final stage of Figure 4. The exact process by which this occurs will be discussed in more detail in the next subsection.

Mapping this string-breakup on an 1D spacetime diagram it can be seen that the $q\bar{q}$ pairs are produced around a hyperbola. Figure 5 shows this process.

In the diagram A is the amount of space-time area covered by the string before break-up, in units of the string tension κ [16]. The total mass m_i of the i th $q\bar{q}$ pair produced on the string is determined entirely by the potential energy between its two adjacent breaking vertices i and j , which can be expressed by the following equation [5]:

$$m_i^2/\kappa^2 = (x_i - x_j)^2 . \quad (3.7)$$

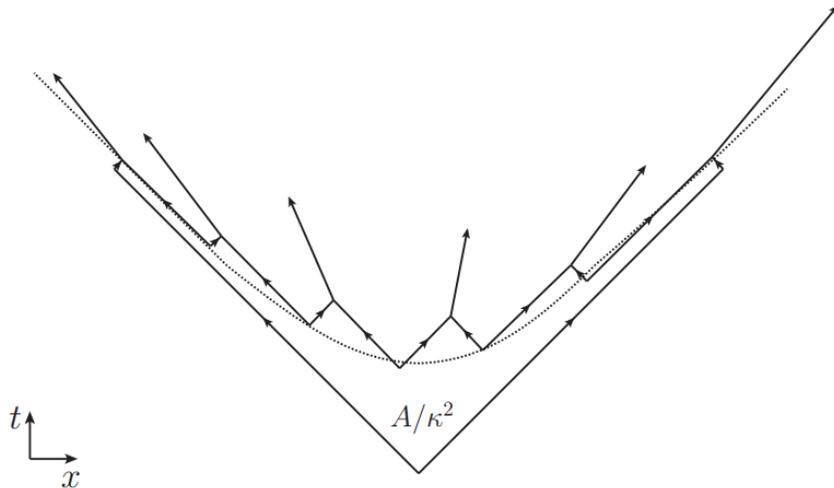


Figure 5. Span of $q\bar{q}$ production on a (massless) Lund string. The boundaries of the diagram correspond to the trajectories of the original quark and antiquark, aligned on the light-like lines of a Minkowski diagram. Figure taken from [5].

For a single string state m , the probability of the production of n $q\bar{q}$ pairs with respective momenta p_i is given by [19]:

$$P \propto \left(\left[\prod_1^n N d^2 p_i \delta(p_i^2 - m^2) \right] \delta^{(2)} \left(\sum p_i - P_{tot} \right) \right) \exp(-bA), \quad (3.8)$$

where N is a dimensionless constant describing the relative weight between states with different numbers of quark-antiquark pairs. b is a constant with dimension $(\text{energy})^{-2}$.

3.4 The Fragmentation Function

The effect described by equation (3.8) can be described iteratively. Each $q\bar{q}$ pair created will remove a fraction z of the available light-cone momentum $E \pm p_z$, while the string will retain the remainder $1 - z$ [17].

Equation 3.8 describes the probability of a Lund string fragmenting into n separate, “unphysical”, string pieces. It does not, however provide information as to how these string states eventually form final-state hadrons. QFT requires that the strings cannot permanently break into purely unphysical states - so final state hadrons should form - however the mechanism by which this occurs is quite complicated.

To avoid the issue of algorithmically describing the formation of hadrons from the complex interacting network of fragmented string pieces, PYTHIA models the string breaks in a more helpful frame. As each of the string breaks are causally disconnected, their relative time-ordering is unimportant. This allows PYTHIA to boost to the endpoint of the string and fragment pieces off sequentially. Only in this frame do individual breaks actually produce string pieces that are hadrons. Repeating this process of “popping” off hadrons from the end of the string thus allows the process of hadronisation from string fragmentation to be modelled without the need to otherwise consider the complex interactions between the unphysical string pieces.

As the string break-up points are causally disconnected, the function describing the probability of this hadron-creation process occurring must have “left-right symmetry.” [15] More generally, this is due to the fact that the same results should be obtained, on average, if the string begins fragmenting from the left or the right.

From this constraint we can derive the probability distribution for each hadron taking a fraction z of the light-cone momentum [5]:

$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(-\frac{bm^2}{z}\right). \quad (3.9)$$

Here constant a is related to N and b via the normalisation condition: $\int f(z) = 1$. Equation (3.9) is the *Lund symmetric fragmentation function*. This equation is particularly important as it is the function used by default in PYTHIA’s MCMC algorithms that it uses to generate events.

Important to note that equation (3.9) does not consider particles with transverse momenta p_\perp . If p_\perp is significant, equation (3.9) can be easily modified with the substitution given by the expression

$$m^2 \rightarrow m^2 + p_\perp^2. \quad (3.10)$$

Further additions to the fragmentation function can also be considered, including flavour change, and consideration of massive quark endpoints [5]. The details of their corresponding adaptations to the fragmentation function are given in [5] but are not relevant to this report.

4 Baryon production

4.1 Simple mechanisms

The mechanisms described in the previous chapter are evidently only capable of describing mesons (in this case 2-quark states), and an additional process must be required to facilitate the production of baryons (3-quark states). Several of these exist.

The simplest method by which baryons can form is through the possible breaking of a $q\bar{q}$ string into a diquark-antidiquark pair. Such a break will produce two string pieces, each connecting a diquark/antidiquark to a quark/antiquark. If each of the constituent quarks within these structures is of a different colour, the two string pieces can form colour singlet states, and thus baryons and antibaryons [16].

Similarly, a baryon-antibaryon pair can be generated through a mechanism known as “popcorn production,” where additional vacuum fluctuations on a single string produce a quark-antiquark pair of different colour to the original endpoint quarks [5]. These quark pairs generate a colour field between them, which in turn can generate a new quark-antiquark pair, of a new colour. Like the diquark-antidiquark method, this can facilitate the string breaking into a triplet and anti-triplet (as well as a meson if sufficient quark-antiquark pairs are generated from the colour field) [16]. The sequence of steps required for this process can be seen in Figure 6.

4.2 The junction

For the goal of modelling baryon production, we can extend the idea behind the basic mesonic “yoyo” interaction introduced previously to a three-quark process [20]. Developing this concept allows the formulation of the concept of junctions, a 3-string interaction with a Y-shaped topology, as shown in Figure 7.

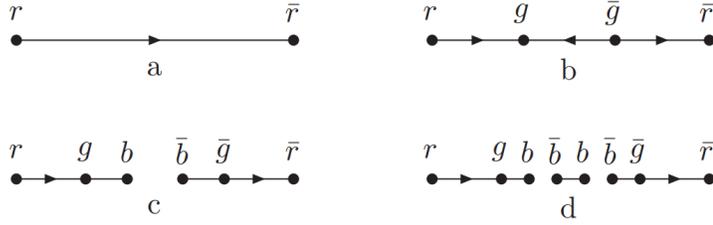


Figure 6. Steps a)-d) of the popcorn process of baryon production. In step b), vacuum fluctuations produce a $g\bar{g}$ pair. In step c), the resultant colour field generates a $b\bar{b}$ pair and the string breaks. Step d) shows the creation of an additional $b\bar{b}$ meson. Figure reproduced from [19].

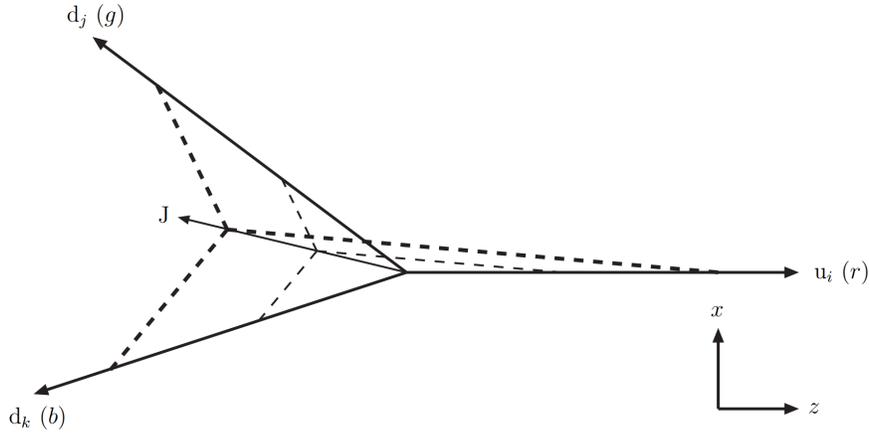


Figure 7. Example junction interaction between u,d,d quarks. J arrow indicates resulting motion of the junction J. Figure taken from [20].

A junction is formed from the separation of three quarks initially bound in a colour-singlet state. The string piece each quark forms that connects it with the initial production vertex to give the junction the characteristic Y-shape [5].

Although there exists no first-principles description of the precise means of junction fragmentation, PYTHIA fragments the two lower-energy legs first as standard $q\bar{q}$ strings. This is achieved through introducing a “fictitious” leg pointing in the opposite direction to the “real” leg, also connected to the centre of the junction. The parton on this fictitious leg is the anti-quark of the parton of the real leg. The directions of these fictitious legs are shown in the dashed lines in the first stage panel within Figure 8.

Once these two strings have fragmented, the remaining two unmatched quarks in the centre of the junction are combined to form a diquark, which forms the end of the remaining unfragmented string. This string is then fragmented as normal [5]. The full process of junction fragmentation is shown in Figure 8.

Importantly, junction fragmentation produces the *junction baryon*, a three-quark triplet state in

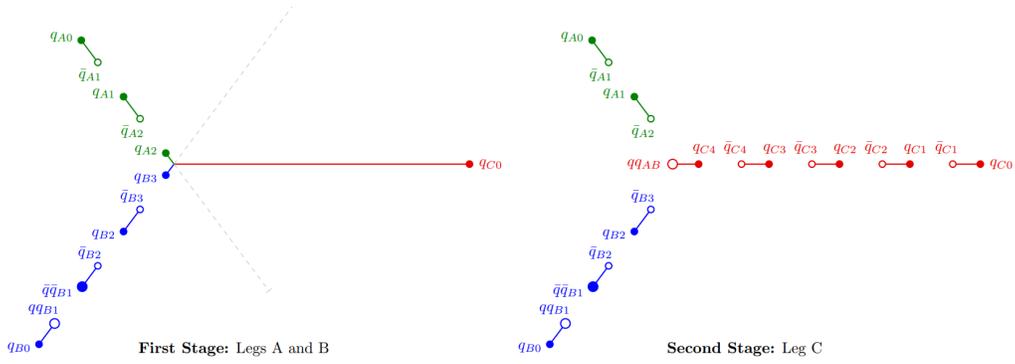


Figure 8. Example diagram of the first two stages of junction fragmentation. The first stage shows the two lower-energy string pieces fragmenting and forming the qq_{AB} diquark in the centre. The second shows the fragmentation of the remaining leg with this diquark as the left endpoint. Taken from [5].

the centre of the junction formed from the “remaining quarks” not paired off in the fragmentation of each of the junction arms. In the case of Figure 8, the junction baryon would correspond to the three quark state $(qq_{AB}q_{C_4})$ in the centre of the second stage panel.

This method of baryon production has proved very useful in explaining the discrepancies of heavy-flavour baryon production ratios in recent ALICE and LHCb experiments. Spectra of these production ratios will form the basis of data used in evaluation of our results in this report.

4.3 Junction motion and orientation

Of significant importance within this report is the understanding of the motion of junctions, in particular the *Junction Rest Frame* (JRF). Being able to reach a frame where the junction is at rest is a necessity within a program that uses junctions to model baryon production, as it allows one to also reach a frame where the junction baryon at the centre is at rest. If this is not achieved, then the resultant junction baryon will be calculated with some extraneous 3-momentum due to the junction’s motion. It is thus much more difficult to reliably determine the junction baryon’s mass and classify it.

A junction is in its rest frame if the angle between each of its legs is 120° , also known as having the “Mercedes topology” given the shape of the car logo. Reaching this frame can be achieved analytically in the case of a junction with purely massless endpoints, however must be solved for numerically when considering massive quarks [20].

Using the invariant four-product a_{ij} of two momenta in the junction p_i and p_j ,

$$a_{ij} = p_i p_j = E_i E_j - \mathbf{p}_i \mathbf{p}_j, \quad (4.1)$$

we take the geometric expansion of the 3-vector scalar product and substitute in the desired angle of 120° . We can thus derive an expression for the “desired” four-product value for two momenta in the JRF:

$$a_{ij} = E_i E_j + 1/2 |\mathbf{p}_i| |\mathbf{p}_j|. \quad (4.2)$$

This lets us set up a function f_{ij} with its root as the “desired” configuration:

$$f_{ij} \equiv \sqrt{|\mathbf{p}_i|^2 + m_i^2} \sqrt{|\mathbf{p}_j|^2 + m_j^2} + 1/2 |\mathbf{p}_i| |\mathbf{p}_j| - a_{ij}, \quad (4.3)$$

which can then be solved numerically [20]. Further details on exactly how this function is used to generate each of the JRF leg-momenta in PYTHIA are given in Chapter 5.

The goal of reaching the Junction Rest Frame forms the main motivation behind the new developments implemented into PYTHIA documented in this report. These new implementations will be discussed at length also in the next chapter.

4.4 The LC limit and Colour Reconnections

The previous chapters have outlined the basic physics regarding the behaviour of Lund strings that have formed between partons. Unfortunately, a more fundamental question regarding these potentials still exists: the Lund string model does not provide any information determining between which partons a confining potential will form. While the simple $q\bar{q}$ case is quite trivial, in practice MC event generators are dealing with arbitrarily large numbers of partons.

For this report we will be considering the partons generated from Multi Parton Interactions (MPIs), which arise from the interactions between many partons within a hadronic scattering process (as opposed to Single Parton Scattering (SPS)) [21]. MPIs are proposed to play a vital role in hadron formation, with predictions of “heavy” hadrons containing b and c quarks having already been made in PYTHIA [22].

In processes occurring at large enough energies for perturbative QCD (pQCD) models to be accurate, event generators such as PYTHIA utilise what is called the leading colour (LC) limit. While in standard QCD, quarks can only be one of three colours (r, g and b), in the LC limit of pQCD the number of colour types is, mathematically, taken to be $N_c \rightarrow \infty$. For large numbers of partons output from a process, this is functionally equivalent to allowing each final state parton to be colour-connected to *one* other parton.

From the $SU(3)$ colour algebra, we can see that the probability of quark-antiquark pair forming a colour singlet is heavily suppressed:

$$\mathbf{3} \otimes \bar{\mathbf{3}} = \mathbf{8} \oplus \mathbf{1} . \quad (4.4)$$

For an arbitrary number of colours, we can generalise equation (4.4) to become

$$N_c \otimes \bar{N}_c = (N_c^2 - 1) \oplus 1 . \quad (4.5)$$

Clearly, equation (4.5) shows us that as our effective number of quark colours increases ($N_c \rightarrow \infty$), the weight of the singlet in the direct sum becomes negligible.

It is worth noting that the group relations given in equations (4.4) and (4.5) only hold for *uncorrelated* quarks, that is, quarks selected at random with no information about where or how they were produced. In many cases for correlated quarks, (for example, a $q\bar{q}$ pair produced from a W boson decay) the probability of them forming a colour singlet is much larger. The individual effects of correlated quarks, however, become negligible when considering the large number of partons and interactions between them in most MPIs.

In summary, in the LC limit, for each outgoing parton there will be a unique other parton carrying the corresponding anti-colour. This produces what are called LC dipole connections. Because of the lack of interference between a given connection and every other dipole connection formed, this leads to each quark-antiquark pair forming its own string piece. [5].

A visual reference showing the LC colour reconnection (CR) process is shown in Figure 9.

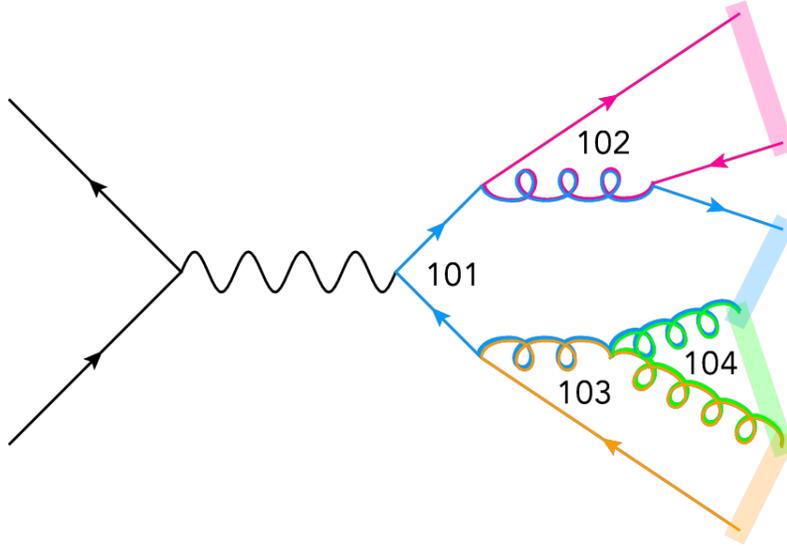


Figure 9. Example diagram showing the LC colour reconnection process for a parton shower from a $e^+e^- \rightarrow q\bar{q}$ event. The shaded regions correspond to each individual confining potential formed. Reproduced from [5].

4.5 Beyond the LC limit

At high energies the LC limit for modelling colour reconnections has been successful at predicting some hadron production mechanisms, such as those from e^+e^- collisions [23]. However, there are several discrepancies that LC-pQCD can not readily explain, such as the observed enhancement of Λ baryons at the LHC [24].

Within quark-antiquark pairs, each particle has 3 possibilities for its colour, $N_c = 3$. As seen in equation (4.4) the probability of the pair confining in a singlet is - relatively - low, at a probability of exactly 1/9 for a fully uncorrelated quark and antiquark. This non-LC probability term is not always negligible, however.

Consider the possible configurations of a pair of gluons, each of which exist as a “mixture” of a colour and an anti-colour, for example red/anti-green. As stated in Chapter 2, there are 8 different types, so the gluons are in the octet representation of SU(3) group. Thus, the colour algebra for a pair of gluons yields the following possibilities:

$$\mathbf{8} \otimes \mathbf{8} = \mathbf{27} \oplus \mathbf{10} \oplus \bar{\mathbf{10}} \oplus \mathbf{8} \oplus \mathbf{8} \oplus \mathbf{1} \quad (4.6)$$

The total number of states here is 64. Only the 27-tuplet (vigintiseptet) corresponds to a total incoherent addition of the two gluons (as opposed to the singlet state, for example, where the two gluons are in the requisite colour states to exactly “compensate” for each other). The consequence of this is the probability of being in the LC multiplet is now $27/64 \approx 42\%$ [5]. Thus, restricting the formation of potentials to purely the leading colour topologies ignores significant “interference” effects that arise from colour connections between partons that do have the possibility to “accidentally” colour connect to other partons [23]. The diagram in Figure 10 shows a potential colour space ambiguity that is ignored in the leading colour limit.

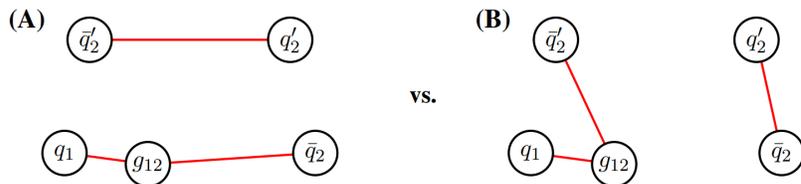


Figure 10. Diagram of a multi-parton state. Sub-diagram A) shows the colour connections allowed in the LC limit. Sub-diagram B) shows a possible alternative string topology. Diagram taken from [23]

Importantly, the consideration of beyond-leading colour reconnections allows a mechanism for the production of junction/anti-junction pairs. In addition to the two decuplets in equation (4.6), we also have the triplet and sextuplet states given from the configurations in

$$\mathbf{3} \otimes \mathbf{3} = \mathbf{6} \oplus \bar{\mathbf{3}} \quad (4.7)$$

and

$$\mathbf{3} \otimes \mathbf{8} = \mathbf{15} \oplus \bar{\mathbf{6}} \oplus \mathbf{3} \quad (4.8)$$

that arise for quark-quark pairs, and quark-gluon pairs, respectively.

With the formation of junctions within these possible colour tuples, we see that beyond-LC colour reconnections are essential in providing a feasible explanation for baryon-antibaryon production in parton showers [5]. The subsequent effective increase in baryon fraction due to colour reconnections is well documented within a range of experiments, which will be discussed in more detail in a later section of the report.

5 Treatment of gluons

5.1 The Pull-vector

One of the present deficiencies within PYTHIA’s string fragmentation modelling concerns the method by which gluons are treated in junction fragmentation. This problem forms the main focus of the work in this report. This chapter and those following will further elaborate on this issue, introduce a possible solution, and discuss predictions made using this new algorithm.

Currently, PYTHIA uses objects called pull-vectors to model the effect of gluons on junction motion. These are necessary when considering the change in the overall junction string topology with additional gluons. This can be seen in the diagram of a qqq junction radiating three gluons, shown in Figure 11.

The additional string pieces shown in blue will work in “pulling” the junction in the direction of their respective gluon. An important comparison to highlight is the difference between a low energy gluon, like the one emitted from the q_j leg, and a high energy gluon similar to the one emitted from the q_z leg. In the former case, the string “kink” created from the transverse excitation of the low energy gluon is small, and thus the effect it has pulling on the motion of the junction is minimal compared to that of the adjacent q_j quark. Formally, this means that in the momentum sum of the q_j leg and its gluon kink, the direction of the q_j dominates.

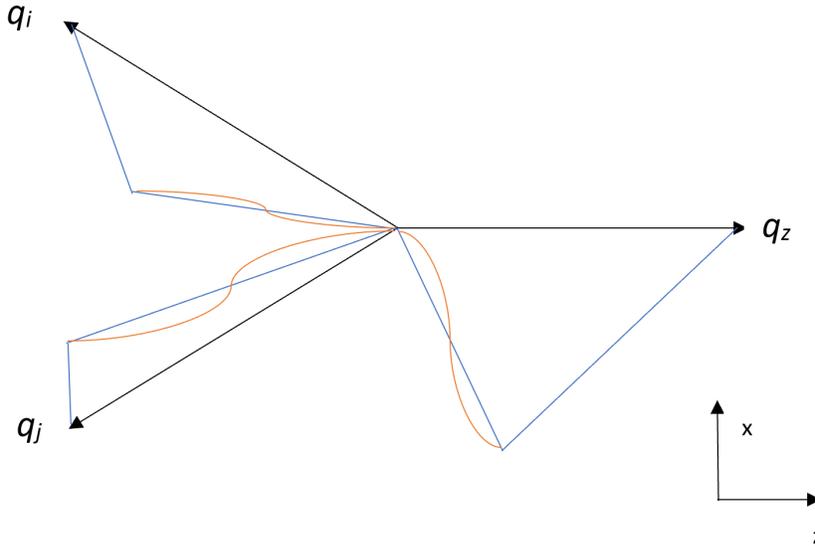


Figure 11. Diagram of a qqq junction with quark endpoints q_i, q_j and q_z . Additional blue string pieces show Lund strings formed between quarks and adjacent gluons. Emitted gluons shown as orange extrusions.

Comparatively, it can be seen that for high energy gluons such as the one emitted from the q_z leg, the requisite kink is orientated significantly far from its source leg, and thus it will have a considerable effect on the overall motion of the junction in this region.

Pull-vectors become necessary when considering the change of energy of gluons over time. From Chapter 2, we can extrapolate equation (3.4) to show that over time, a gluon moving away from the junction origin will lose energy via equation (5.1):

$$\frac{dE_g}{dt} = -2\kappa . \quad (5.1)$$

Here the factor of 2 is obtained from the two string pieces of the gluon “kink” [20]. Once a gluon has exhausted all of its energy, it will no longer dominate the pull of the junction in its region. In the case of q_z in Figure 11 the quark leg would subsequently dominate.

In many processes, however, quarks emit multiple gluons, which would thus take the form of multiple connected kinks of the kind seen in the diagram. First, the gluon closest to the junction dominates the overall “pull”, then when this gluon depletes its energy, the junction will be pulled by the next-innermost, and so on, for all of the remaining gluons [5].

The motion of multiple gluons within the junction is evidently a complicated mechanism, and to simplify the process PYTHIA uses an object which, broadly, describes the average direction the junction is being pulled by each of the gluon string pieces. This is what is called a “pull-vector.”

For each of the gluons’ momenta p_i , PYTHIA defines the pull-vector as the weighted sum:

$$p_{pull} = \sum_{i=1}^n W_i p_i , \quad (5.2)$$

where i is the innermost gluon, $i + 1$ is the second-innermost, up to the n th gluon. The weights, W_i , of the pull-vector are chosen specifically such that gluons further from the junction origin (which “pull” less on the junction) are suppressed. This is represented by equation (5.3),

$$W_i = e^{-\sum_{j=1}^{i-1} E_j / E_{norm}} , \quad (5.3)$$

normalised to a free parameter E_{norm} [20]. The negative exponential dependence in equation (5.3) will ensure that when i is small, the resultant sum over j is small, and thus W_i is comparatively larger.

As the energies of the gluons E_j are frame-dependent, PYTHIA requires an iterative procedure to find them in the desired rest frame of the junction [20]. This can be prone to error, as seen in the next section.

5.2 The current method for reaching the Junction Rest Frame

The majority of the modifications to PYTHIA’s algorithms introduced in this report involve the *fragmentToJunction* method within the *StringFragmentation* class ².

Currently, the *fragmentToJunction* method uses an iterative procedure to boost systems into the Junction Rest Frame. These parton systems will start in their centre-of-mass frame, in which their effective leg 4-momenta are calculated using the pull-vectors, using equations (5.2) and (5.3) to perform a weighted sum over every parton on the leg.

The initial junction configuration defined by the momenta calculated from the pull-vectors will not be in the JRF. The “deviation” of this system from the JRF is quantified by the deviation of the legs’ orientation from the desired 120° topology. In the code this is calculated as the variable *errInCM*, which is the sum in quadrature of the cosine of the angle between each pair of legs, given by

$$errInCM = (\cos(\theta_{p_{0,1}}) + 0.5)^2 + (\cos(\theta_{p_{1,2}}) + 0.5)^2 + (\cos(\theta_{p_{0,2}}) + 0.5)^2 , \quad (5.4)$$

where $\theta_{p_{i,j}}$ is the angle between each of the legs. The 0.5 addition in each term ensures that in the “correct” case, when the system is in the JRF, and the angle between each vector is 120°, each term in *errInCM* will be 0. Likewise, any system where the angles are incorrect will yield a higher value of *errInCM*, making the measure a convenient judge of the relative “success” (in reaching the JRF) of a given system.

These leg momenta are then used in the *junctionRestFrame* method, which returns an estimate for the boost matrix required to boost the system into the Junction Rest Frame. This is achieved by numerically finding the root of equation (4.3) to give an estimate of one of the momenta $|p_1|$, and then substituting the value into the other permutations of f , f_{12} and f_{13} and analytically solving for $|p_2|$ and $|p_3|$. Any permutation of f_{jk} is monotonically decreasing if formulated as a function of $|p_i|$, so theoretically any bracket-based root finding algorithm should converge on the root, if it exists [20].

Currently, the algorithm in *junctionRestFrame* uses a simple bisection method for the first half of the root-finding iterations, and then the secant method for any subsequent iterations. Note that there may exist scope for improvement of this algorithm in the future: In the (possibly erroneous) case where the current function f_{jk} does not contain a root within a physically meaningful range (for example, if the only root that exists is for $|p_i| < 0$), then the algorithm would be expected to return the value of $|p_i|$ closest to $f_{jk} = 0$. While this eventuality will happen with the bisection method, it is not guaranteed with the secant method, which can be susceptible to abrupt variation in gradient throughout the root-finding function. Redesign of the algorithm to account for this discrepancy may prove useful, however, this was deemed outside of the scope of the work in this report.

²Details of the code for the old and new methods are shown in Appendix B.

Mode	Warning number	(%) of total events
Default (No CR)	10	0.25%
0	638	15.95%
2	397	9.93%
3	547	13.68%

Table 2. Number of non-convergence warnings generated from 4000 events of 13 TeV pp collisions, for the default Monash mode, and the three CR modes.

Once the three solution momenta p_i of the legs are known, the partons can then be boosted into their rest frame, and then boosted again into the Junction Rest Frame [20]. The overall boost to the JRF is then calculated and then output to the *fragmentToJunction* method.

This boost matrix is then applied to the current matrix *MtoJRF* designated to boost the system into the JRF. The entire previously mentioned process (from the start of *fragmentToJunction*) is then repeated until the boost matrix output from the method *junctionRestFrame* is within a tolerance of 10^{-5} from the identity matrix (so any subsequent boosts would have a negligible effect on the system). The method *fragmentToJunction* is called a maximum of 20 times before stopping automatically.

Lastly, the success of the algorithm is checked by comparing the orientation of the final parton system (which should be in the Junction Rest Frame) with the original orientation. This is achieved by comparing their angle measures as calculated from equation (5.4). If the final system error is larger than the initial, the algorithm is judged as unsuccessful in finding the Junction Rest Frame, and the original centre-of-mass frame is taken as the closest approximation to the JRF.

The identification and elimination of these errors in reaching the JRF is the main imperative of the work in this report.

5.3 Problems with JRF convergence

The aforementioned error in finding the Junction Rest Frame manifests as a warning in PYTHIA ³. The relative frequency by which these warnings arise in events is important, as it indicates one of two hypotheses: either the frequency of non-converging events is small, and thus a proper treatment of junction motion would not produce significantly different particle production spectra; alternatively, the frequency of non-converging events is high, and visible improvements to the spectra generated by PYTHIA will be evident.

To analyse the extent of this problem, four test cases were chosen. In each of these cases, the fraction of non-convergence warnings output in PYTHIA are shown in Table 2. All are sourced from proton-proton (pp) collisions at 13 TeV, with the generator run for 4000 events. The first case is taken without the consideration of CR effects (on the default Monash tune), whereas the rest are run with CR effects under *Mode 0*, *Mode 2* and *Mode 3*. The differences between modes will be discussed further later in this report, and are also shown in Appendix A.

In the default-tune case, it is evident that the number of events that trigger the warning is negligible, even at 4000 events it only arises 10 times, which corresponds to a relative frequency of 0.25%. Comparatively, the cases where CR is included have a substantial number of events where the system does not converge on the JRF, with the lowest for *Mode 2* at approximately 10% and the highest at *Mode 0* with 16%.

³Formally, *Warning : badconvergencejunctionrestframe*.

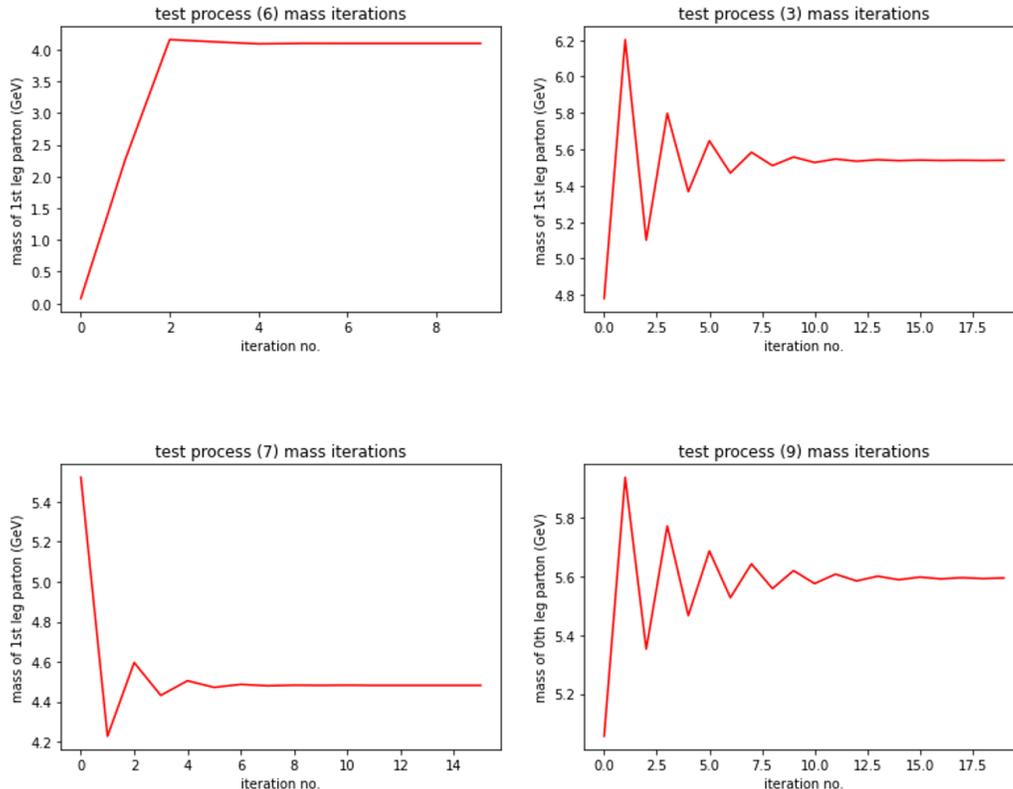


Figure 12. Collation of four of the test cases’ total mass of most massive leg, per iteration of the algorithm. This set shows the “incorrectly-converging” group.

For the spectra produced in the subsequent chapters of this report, that are produced from 1,000,000 events and require inclusion of CR effects, having at least 10% not converge correctly may have a considerable effect on the eventual results generated by PYTHIA. It is necessary then, to diagnose the underlying problem within PYTHIA’s JRF convergence algorithm.

To further understand the behaviour of the code, the state of the parton system was mapped out at each step of the algorithm’s iterating scheme. A simple and informative marker of this behaviour was the (total averaged) mass of the most massive leg in the system. For several test cases obtained from events generated for 8 TeV pp collisions, this quantity was tabulated over every iteration of *fragmentToJunction*.

In Figures 12 and 13 each sub-panel is showing the behaviour of an event that returned a non-convergence warning. Broadly analysing the results, we can observe two distinct problems with the algorithm’s behaviour, each of which is shown in a separate Figure.

5.3.1 The angle measure

Firstly, Figure 12 groups together the events containing a mass value that appears to converge correctly, yet the resultant system still returns the non-convergence warning. Observing test processes (6) and (7) the program does in fact converge on the Junction Rest Frame within the maximum number of allowed iterations (20). Similarly, while test processes (3) and (9) do not fully converge on a value

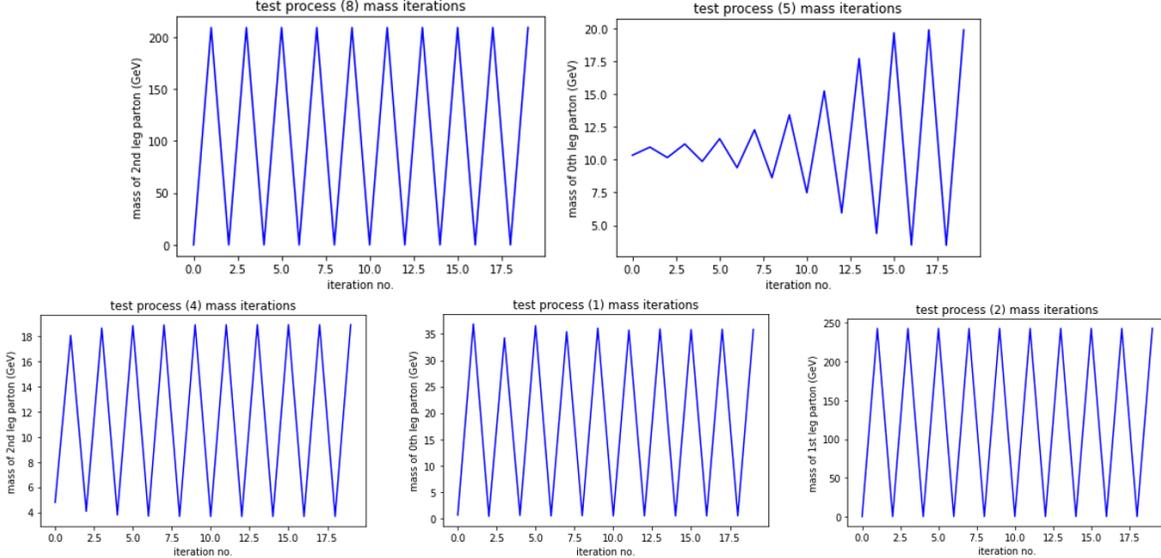


Figure 13. Collation of five of the test cases’ total mass of most massive leg, per iteration of the algorithm. This set shows the “Oscillating” group.

Process no.	$p_x(GeV)$	p_y	p_z	mass
3	0.000	0.000	0.001	5.540
6	0.000	0.000	-0.000	4.095
7	-0.000	0.000	0.000	4.482
9	0.000	0.001	-0.002	5.595

Table 3. For each of the incorrectly-converging test cases, the individual values of the 3-momentum of their final-state most massive leg, as well as the mass of that leg.

smaller than the desired minimum tolerance (10^{-5}), both sub-panels show the algorithm is visibly approaching a steady-state system as the iteration number increases. It is reasonable to extrapolate then that given a larger number of allowed iterations these test cases would likewise converge.

Analysing the final-state momenta of each of the aforementioned test cases’ most massive legs, an explanation arises as to why these events, which appear to be converging correctly, are not designated by the algorithm as reaching a frame that improves on the centre-of-mass frame.

For every process, each component of the 3-momentum of its most massive final leg is zero, or approximately zero in the cases of the near-converging processes (3) and (9). This is shown in Table 3. The state of such a system is that it has reached the *heavy quark rest frame*, a junction topology where the most massive leg has zero 3-momentum, as it has been transferred entirely to the other legs. The algorithm is expected to converge to this frame when the only root that exists for f_{jk} (equation 4.3) is negative, as the bisection method, in an attempt to “reach” this root, will reduce $|p_i|$ to 0.

In the current algorithm, a heavy quark rest frame solution is a problem due to the angle measure calculated in equation (5.4). If the final system’s deviation from 120° exceeds the deviation of the system in the C.O.M. frame, the algorithm is judged as unsuccessful in reaching the Junction Rest Frame and the warning is triggered. However, for the zero-momentum parton in the heavy quark rest

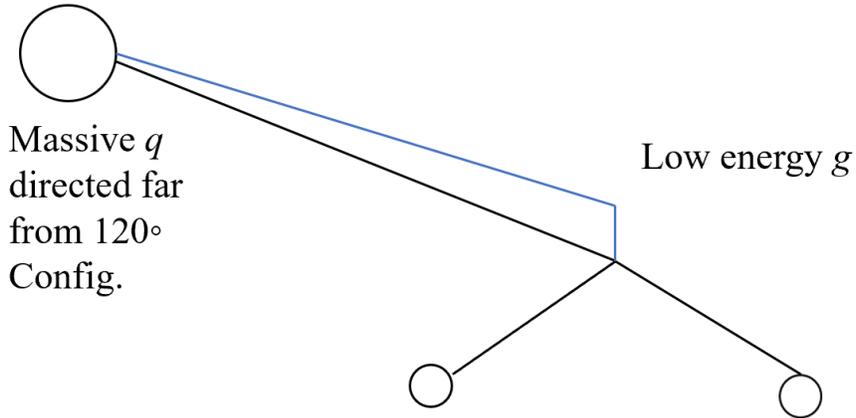


Figure 14. Junction configuration consisting of low energy gluon and very massive quark orientated in direction far from 120° configuration.

frame, the calculation of the angle between the legs is problematic.

Theoretically, the issue that arises here is that the angle between a vector and a zero-vector is ill-defined. In practice, due to the limited numerical precision of a numerically converging algorithm, the heavy quark leg will have small, but non-zero, momentum components, as can be seen in Table 3. PYTHIA will use this momentum vector as part of the system in which it calculates the angle measure in equation 5.4, and can decide to accept, or reject the new frame depending on the arbitrary “direction” of the heavy quark momentum vector. This renders the angle measure in equation 5.4 an unreliable metric of whether the system has converged correctly or not.

It is clear then that the algorithm of *fragmentToJunction* must be altered such that either the means of checking the ending frame reached against the C.O.M. frame can account for the heavy quark rest frame, or such that the check is no longer necessary within the algorithm and can be discarded. The latter was achieved with the new algorithm discussed in the next subsection.

5.3.2 The soft gluons

The second type of test cases that return non-convergence warnings are shown in Figure 13. For this category, the effective mass of the most massive leg simply oscillates between two values for each iteration. In each of these processes, the algorithm will clearly never converge on the Junction Rest Frame, and an updated treatment of these specific cases is required.

The hypothesised explanation for this behaviour involves the algorithm’s treatment of legs with soft (low-energy) gluons. Consider a hypothetical junction configuration where one leg possesses a very low energy gluon and a very massive quark, orientated in a very different direction, similar to what is shown in Figure 14.

When comparing the contributions of each of these partons, if the gluon is sufficiently soft the massive quark will make up the majority of the total centre-of-mass energy of the system (also called the squared invariant mass of the system), and will thus dominate the effective “pull” on the junction from the two partons. The expression for the COM energy E_{CM} is given by equation for n particles each with momentum p_i (5.5):

$$E_{CM}^2 = \left(\sum_i^n p_i^2 \right)^2. \quad (5.5)$$

As a gluon is massless if it has low energy its requisite p_i is small, and thus the larger 4-momentum of the massive quark will dominate the expression in equation (5.5).

Subsequently, the “effective” direction of the junction leg calculated by the pull-vector will approximately point in the direction of the massive quark, creating a system that requires a significant boost “back” to the desired 120° configuration. This will be applied as expected in an iteration of *fragmentToJunction*.

However, the large boost required to return to the correct configuration imparts significant energy to the other partons on the junction legs, specifically *the gluon*. Now the gluon has very high energy, it will dominate the COM energy and the massive quark will now exert a negligible influence on the junction. If the massive quark has a negligible effect, the junction is taken as effectively massless.

In this case, as stated in Chapter 4.3, an analytical solution to the JRF exists, so the *junction-RestFrame* method will use this to determine the boost required to return the now effectively massless system back to the 120° configuration. As the gluon is now the parton dominating the pull on the junction, the boost found will return the system back to the starting configuration with a low energy gluon, and now a massive quark again. From here, the previously mentioned process repeats, and we obtain the oscillatory behaviour visible in Figure 13.

The issue here is that the way the pull-vectors are constructed causes high-momentum partons to incorrectly skew the contributions of other partons on their leg. To account for this error, an updated treatment that considers each successive parton *individually* was introduced, and is outlined in the next section.

5.4 The iterative-removal algorithm

As a solution to the problems with convergence outlined in the previous subsection, a new algorithm called the *iterative-removal algorithm* was used.

The crux of the iterative-removal algorithm is that it now calculates the boost to the Junction Rest Frame through a successive procedure of considering the effect of each individual parton on each leg. It functions as follows.

The algorithm will start with the three closest partons on each leg, and then boost to the JRF of just this sub-system, finding the requisite junction velocity for this boost. It will then identify the leg with the smallest 3-momentum.

The momentum Δp from the initial smallest-momentum leg will then be subtracted from the other two (larger) legs. The junction leg with the smallest 3-momentum is updated to the subsequent parton on that leg. These new momenta create a new 3-parton system. At this point the process will then repeat, until the algorithm reaches the endpoint of one of the legs. If the endpoint parton is massless, then the algorithm will stop at this point, as the COM frame of the system does not exist. If it is massive, then the algorithm will also calculate the required boost velocity for the system with the endpoint parton. The basic steps of this process are shown in Figure 15.

For each iteration of this procedure, the junction velocity calculated is stored. To find the resultant, overall junction velocity of the entire system, each individual junction velocity is averaged in an inverse-exponential weighting scheme similar to the original pull-vector.

Equation 5.1 shows that the energy of a parton decreases with time, and thus the effect of its “pull” on the junction decreases with time. Thus, the larger the proper time τ of each parton, the

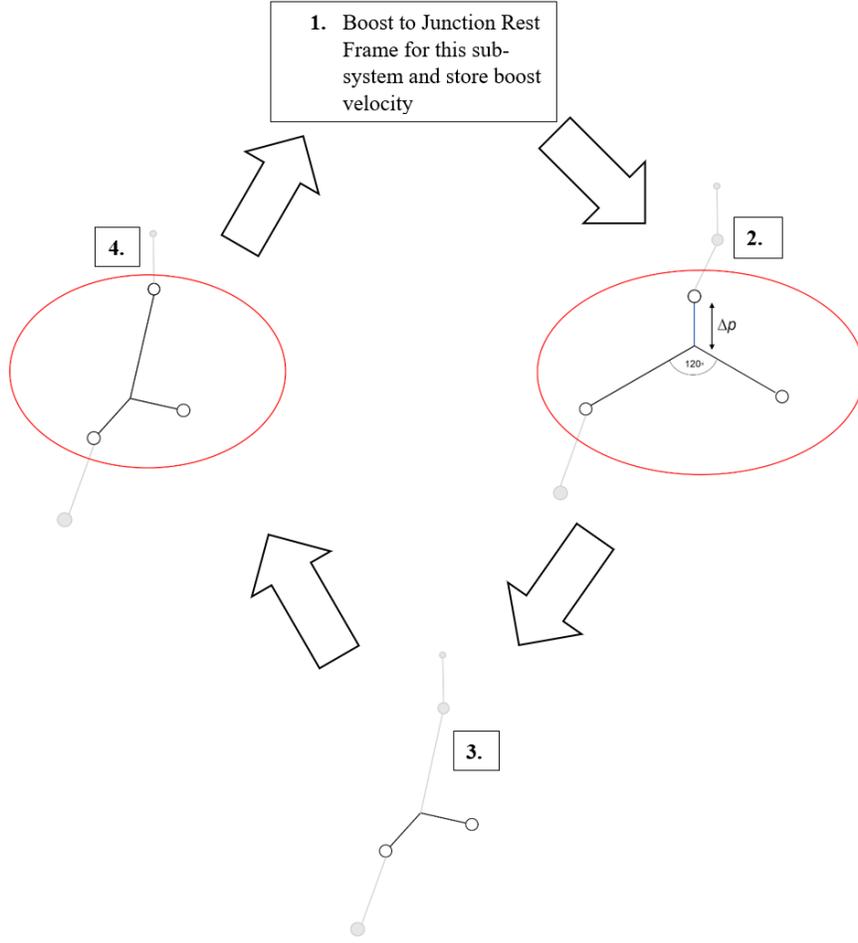


Figure 15. Schematic of the four key steps in an iteration of the iterative-removal algorithm:

1. Innermost 3 partons on each leg are selected as the current subsystem, and algorithm boosts to their JRF
2. Smallest-momentum leg is identified
3. Momentum Δp subtracted from other legs
4. System moves to select the next parton for the sub-system from the depleted leg.

further it has moved from the junction origin, and the more it should be suppressed in the averaging scheme.

A general expression for the weighting scheme of the n junction velocities is given by equation (5.6):

$$v_{tot} = \sum_{i=1}^n \frac{v(\tau_i) [\exp(-c\tau_{i-1}) - \exp(-c\tau_i)]}{1 - \exp(-\sum_{j=1}^n c\tau_j)}, \quad (5.6)$$

where v_{tot} is the overall, final junction velocity and $v(\tau_i)$ is each individual junction velocity. c is a normalisation constant, given by an approximate hadronisation energy scale $p_H = 2$ GeV (designated in the code as *pNormJunction*).

It is worth noting that although the above method removes the requirement for the use of a pull-vector in the calculation of the junction velocity, a pull-vector is still used within PYTHIA to designate the direction of the “fictitious leg” created in order to fragment each leg of the junction as a $q\bar{q}$ string. Further testing of the new algorithm beyond the scope of this report may be useful to determine whether this is a significant source of error.

An additional correction was also added to *fragmentToJunction* that ensured, in the case where the algorithm reaches a situation like that displayed in Table 3, the system is reset to the heavy quark rest frame, to avoid the issue discussed previously in Chapter 5.3.1. This correction only occurs when one of the legs of the system is judged to have a significantly small momentum, and the angle between the remaining legs exceeds 120° .

The primary advantage of the iterative-removal algorithm over the current one is that the process of moving to each parton individually is NOT frame-dependent. To observe whether this new scheme is successful, as well as analyse any new results that might arise from this differing treatment of gluons, an array of test cases were prepared to compare the old and new schemes, shown in the next section.

6 Comparison of Old and New-scheme events

6.1 The Dalitz plot

A Dalitz plot is a two-dimensional visual representation of the three-body phase space of three bodies in particle physics. It was first invented by Richard Dalitz in his analysis of the τ -meson [25]. While they are typically used to represent quantities within 3-body decays, for this report they will be used to analyse the properties of the three partons within junctions.

The essence of a Dalitz plot is that it fully describes the behaviour of a three-body system through only two kinematic variables. While a three-body system, generically, is described by 12 independent parameters (four components for each of the three four-momenta in the system), imposing the requirement of four-momentum conservation reduces the number of variables to 8. Furthermore, if the masses of the bodies within the system, this reduces the number of parameters to 5. Finally, 3 of the parameters correspond to “Euler Angles” that simply change the global orientation of the system and have no effect on its internal behaviour. Thus, a three-body system can be completely described by only two-parameters [26].

Traditionally, the parameters chosen for a Dalitz plot will be the squared invariant masses of two of the bodies i and j , given by equation (6.1):

$$m_{ij}^2 = (p_i + p_j)^2 = p_i^2 + p_j^2 + 2p_i p_j . \quad (6.1)$$

6.2 Plot generation method

For the analysis within this report, just the last term from equation (6.1) $2p_i p_j$ is treated as a variable in the plots. This is because each of the squared momenta p_i^2 are simply the squared mass of each of the partons on the legs, which will be fixed for each individual Dalitz plot.

The four-momentum product term is also normalised to the square of the total centre-of-mass energy of the system E_{CM}^2 , found from equation (5.5).

This enables the axes of each of the Dalitz plots to be standardised (spanning values $\in [0, 1]$) for all parton system configurations that are tested.

To analyse the behaviour of both the old algorithm within PYTHIA, and the new iterative-removal algorithm, Dalitz plots were created with axes of double the normalised four-product of the first and second junction leg momenta, and the product of the second and third leg momenta: $\frac{2p_i p_j}{E_{CM}^2}$.

6.2.1 The starting momenta configuration

From these inputs, it is possible to uniquely generate a four-momentum for each of the system legs. An outline of the process of converting the normalised four-products into four-momenta is given as follows.

Defining the variable s_{ij} as two times the non-normalised four-product of momenta p_i and p_j ,

$$s_{ij} \equiv 2p_i p_j , \quad (6.2)$$

in the C.O.M. frame of the system we can develop a generic expression for each of the three legs' energy [27],

$$E_i = \frac{E_{CM}^2 - s_{jk} + m_i^2}{2E_{CM}} . \quad (6.3)$$

Combining this expression with Einstein's energy momentum relation $E^2 = p^2 + m^2$, we can obtain the magnitude of the momentum for each leg p_i . Furthermore, in the C.O.M. frame, the relative angle between two of the legs can be derived [27]:

$$\cos \theta_{ij} = \frac{2E_i E_j + m_i^2 + m_j^2 - s_{ij}}{2|\mathbf{p}_i| |\mathbf{p}_j|} . \quad (6.4)$$

Fixing the system in the xz plane, allows us from these values to derive explicit expressions for each of the momenta of the three legs:

$$p_1 = (0, 0, |\mathbf{p}_1|, E_1) , \quad (6.5)$$

$$p_2 = (-|\mathbf{p}_2| \sin \theta_{12}, 0, |\mathbf{p}_2| \cos \theta_{12}, E_2) , \quad (6.6)$$

and

$$p_3 = (-|\mathbf{p}_3| \sin \theta_{13}, 0, |\mathbf{p}_3| \cos \theta_{13}, E_3) . \quad (6.7)$$

6.2.2 The values of the Dalitz plot

These derived four momenta were then used to fully define the starting system which was input into PYTHIA. PYTHIA was then run once on its default settings, and the resultant boost velocity calculated to boost from the starting configuration into the JRF was recorded. Subsequently, the velocities obtained from each of the different inputs of the four-products were used as the values for the contours within the Dalitz plots.

Several corrections were required to ensure the plotting algorithm was successful over all input values. Firstly, in the case when the starting 3-parton system possesses two legs oriented very close together (i.e. the angle between them is small), PYTHIA judges the relative effects of each individual leg as insignificant and instead fragments the system as a diquark-quark string topology. When this occurred, the diquark velocity was instead taken as the equivalent junction velocity for the Dalitz plot.

In the regions of the grid where the input values did not yield a physically meaningful system (for example, when the derived energy values were smaller than the starting masses), the resultant "velocity" value for the Dalitz plot was set to 1, to ensure correct limiting behaviour of the plot contours at the edges of the physical region. To check whether a given momentum configuration was physically meaningful, we made use of the determinant of the Gram matrix. For a set of vectors $\{p_1, \dots, p_n\}$ the

Gram matrix is defined as the $n \times n$ matrix of each of the inner products of the vectors: $\langle p_i p_j \rangle$ [28]. If the determinant of the Gram matrix is positive, the vectors that formed its components are known to comprise a vector space.

For the purposes of our analysis, the Gram determinant can be used to determine whether a given set of momenta products s_{ij} refer to a set of momenta $\{p_1, p_2, p_3\}$ that describe a system within the bounds, of real, physical phase space. The corresponding Gram matrix for the momentum vectors with generic indices $\{i, j, k\}$ is given by the matrix,

$$\begin{bmatrix} p_i p_i & p_i p_j & p_i p_k \\ p_j p_i & p_j p_j & p_j p_k \\ p_k p_i & p_k p_j & p_k p_k \end{bmatrix}$$

which, discarding the constant $1/2$ factor, is equivalent to the matrix:

$$\begin{bmatrix} s_{ii} & s_{ij} & s_{ik} \\ s_{ji} & s_{jj} & s_{jk} \\ s_{ki} & s_{kj} & s_{kk} \end{bmatrix}$$

An expression for the determinant of this matrix is given by [29]:

$$\Delta_3 = \frac{1}{4}(s_{ij}s_{jk}s_{ik} - s_{ij}^2 m_k^2 - s_{jk}^2 m_i^2 - s_{ik}^2 m_j^2 + 4m_i^2 m_j^2 m_k^2) . \quad (6.8)$$

After the momenta of test configurations were derived, the above determinant was calculated and checked to be positive. If it was not positive, the starting configuration was deemed unphysical.

This plot-generation program facilitated a simple and effective method of observing the behaviour of both of PYTHIA's algorithms in a highly controlled environment. In the next sections, analysis will be performed on these generated Dalitz plots for a range of starting parameters, to observe the effects of considering various factors and inputs, and to compare the success of PYTHIA's old and new algorithms.

6.3 Compendium of test cases without gluons

To begin with, starting configurations without gluons were run through PYTHIA, in order to analyse the problem of the angle-measure in isolation. This allowed an important conclusion to be quickly drawn: when considering gluonless starting configurations, the no-convergence warning is *never* triggered. The algorithm was tested for this conclusion, considering a range of starting configurations of different masses, as well as for a system rotated by an arbitrary angle.

The explanation for this phenomenon is not yet conclusive, yet is likely that the method for numerically finding the solution in *junctionRestFrame* always converges from the same direction. Thus, if the system reaches the heavy quark rest frame, although the heavy quark momentum should, theoretically, be a zero-vector, the residual vector generated from the root-finding algorithm is always orientated in a direction such that the angles measured between it and its adjacent legs are considered an improvement on the COM frame. This is supported by the observation that the value of the f_{jk} function from equation (4.3) for the final system found from each configuration is *always negative*.

Evidently, further study of this issue may be required to explain the behaviour of the old algorithm in these cases, however this will likely be unnecessary once the new algorithm is finalised (and the angle-measure is removed altogether).

Regardless of this result, analysing the test cases where no change between the algorithms is expected is still useful to demonstrate the base functionality of the new iterative-removal algorithm. Thus, as a beginning point of comparison, test cases containing no gluons were first generated.

OLD

NEW

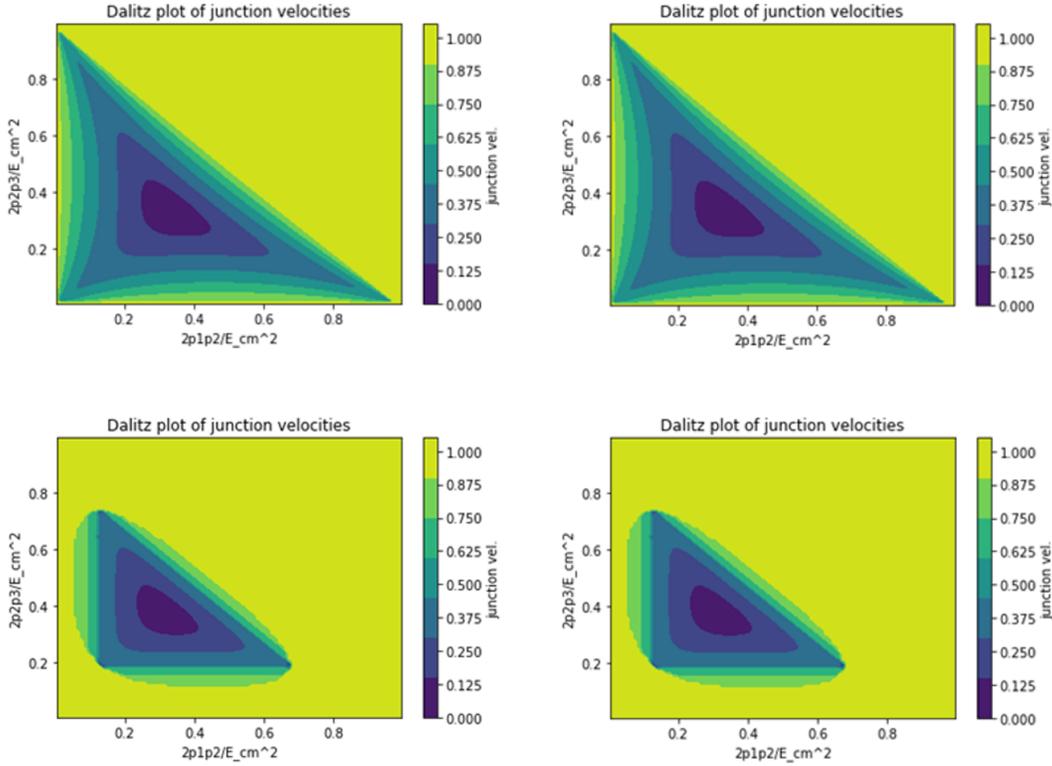


Figure 16. Table of Dalitz plots formed from two test cases: One with all massless leg endpoints, and one with one 1.5 GeV mass per leg. The first column was generated from the old PYTHIA algorithm, the second from the new algorithm. Each contour plot is a 100×100 point grid, generated in Python.

Figure 16 shows a comparison of two separate test cases, each run through the old and new algorithm of PYTHIA. The first row of the Figure shows the velocity distribution obtained from a purely massless starting configuration. The second row is comprised of velocities calculated from a starting configuration consisting of a single charm quark (1.5 GeV mass) on each junction leg. In both cases the C.O.M. energy was fixed at 10 GeV.

Comparing the old and new algorithms in Figure 16, it is evident that the requisite phase spaces of each shown in the Dalitz plots are visibly identical. Each corresponding point with the same input values is the same in both algorithms, for both test cases, to a precision of 10^{-6} (this is the maximum absolute difference between two corresponding points). This conclusion is also reached from the plots of many other starting configurations with different mass endpoints.

It is thus reasonable to conclude that the iterative-removal algorithm has been successful at reproducing the same results previously obtained from PYTHIA’s old algorithm, but through a new method. This outcome is expected, as the old algorithm was observed to always converge for systems without gluons. Overall, the comparison in this chapter does work as an adequate litmus test of the

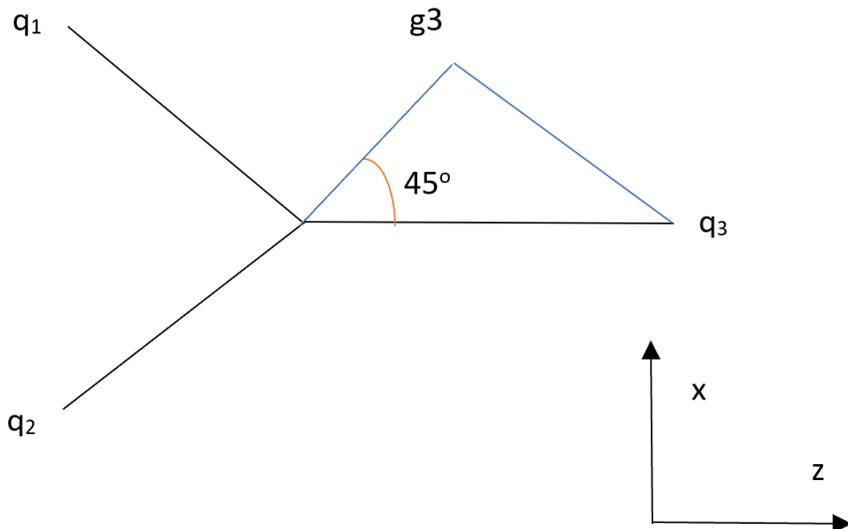


Figure 17. Diagram of the starting 1-gluon setup, located 45° from the positive z axis.

functionality of the new algorithm, and confirms it possesses no new issues in reaching the JRF that were not present in the old PYTHIA.

6.4 Compendium of test cases containing gluons

The focus of the improvements to PYTHIA’s modelling concern the new treatment of gluons through the iterative-removal algorithm. To visualise the results of the modifications to the old algorithm, a compendium of test cases was created, each highlighting the effect of varying a particular parameter within the starting configuration.

6.4.1 Gluon energy variation

To begin with, we start by adding a single gluon to the basic 3-quark system, colour-connected to the parton pointing in the z direction. The gluon is located at a polar angle of 45° (measured from the positive z direction). A basic schematic of the starting configuration is shown in Figure 17. Note the string-leg vector lengths are only for visual reference, and their length is not representative of their magnitude.

The goal of this subsection is to analyse the effect of varying the energy of the gluon in the system. Three test cases for both algorithms are included: A gluon with energy 0.25 GeV, 2 GeV and 10 GeV. All other parameters are fixed, with the endpoint masses each at 0.33 GeV, and the C.O.M. energy at 10 GeV. A collation of the results of these three cases is shown in Figure 18.

Figure 18 now shows clear visible differences between the results of the old and new algorithms. For the low energy gluon case, as expected both systems exhibit behaviour very similar to the gluonless cases in the previous sub-chapter, as the effect of the gluon relative to the centre-of-mass energy is comparatively negligible. Nevertheless, the results of the new algorithm show apparent discontinuities within the spectrum of junction velocities on the upper edge of the range of $2p_1p_2/E_{CM}^2$ approximately $\in [0.75, 0.85]$. A plot better highlighting this region of discontinuity is shown by Figure 19, that plots the absolute difference between the velocity values found from the old and new algorithms. Note the non-zero differences in the lower right-hand corner.

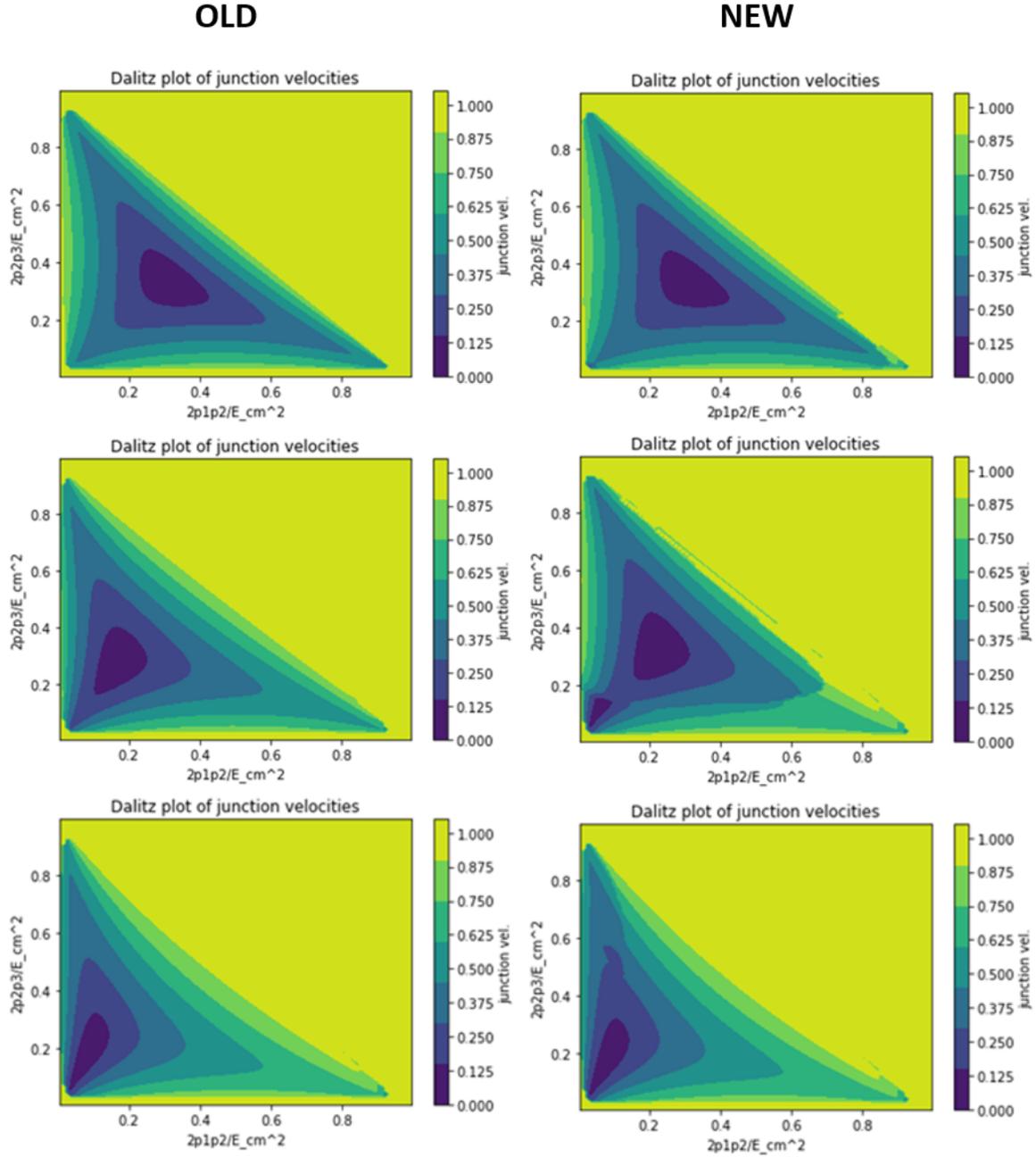


Figure 18. Array of Dalitz plots generated for starting configuration with 1 gluon, located at polar angle 45° . Each is a 100×100 point grid. First row shows system with gluon energy 0.25 GeV, second with 2 GeV, third with 10 GeV. Left column is the old algorithm, right is the new algorithm.

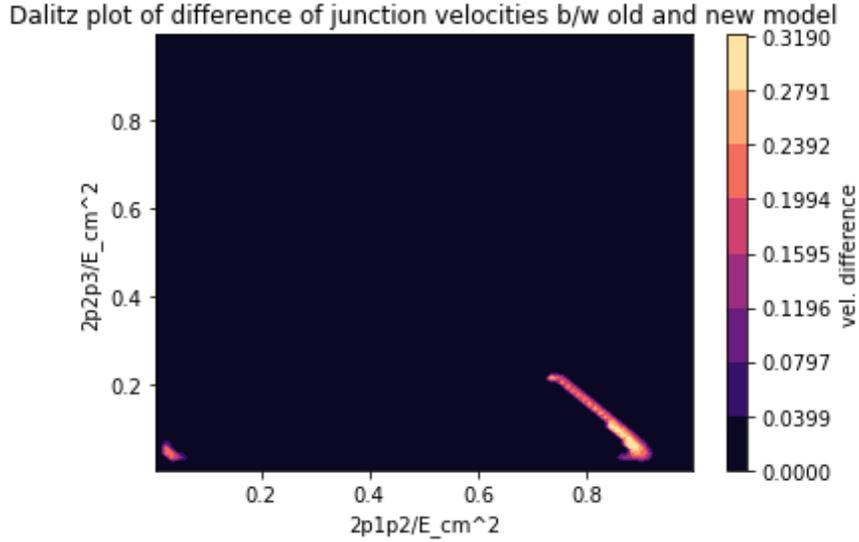


Figure 19. Dalitz plot of the absolute difference in velocities output from the new and old algorithm, for the 0.25 GeV gluon test case.

Differences between the new and old models are expected given the new treatment of gluons in the iterative-removal algorithm, however it is the discontinuities in the new algorithm Dalitz plots that suggest error. For example, observing the grid point $(0.825, 0.125)$, which yields a junction velocity of 0.61; there is a considerable difference between it and the value of its adjacent grid point, $(0.825, 0.135)$ which has a velocity of 0.93. This manifests visually as a clear gap in the otherwise smooth contours of the Dalitz plot.

For the sake of brevity hereafter, as in each Dalitz plot the x and y axes correspond to the same normalised four-product terms, when referring to these product terms, they will be dubbed x ($2p_1p_2/E_{CM}^2$) or y ($2p_2p_3/E_{CM}^2$).

Analysing the starting configurations of the two previously mentioned grid points reveals nothing that would potentially motivate such a large step in junction velocity, as the angles between each leg, and leg momenta are very similar in both cases. The two configurations can be seen in Figures 20 and 21. These diagrams are approximately to scale, with the relative angles between each of the leg vectors specified, as well as their length directly corresponding to the magnitude of the momentum 3-vectors for each leg.

Thus, this discontinuity in the new algorithm’s 0.25 GeV gluon panel is unlikely to be physically motivated, and may correspond to a deficiency present within the current implementation of the new system.

Unfortunately, the possibly incorrect behaviour of the algorithm at the edges continues into the 2 GeV gluon case in the second row. While both the old and new algorithms present broadly the same overall results, there are two key differences: The “band” of very high junction velocities that has developed near the diagonal edge of the Dalitz shape, and also an additional low velocity region appearing in the top left of the new algorithm’s dalitz plot.

Both new plot artefacts are definitely not likely to be physically motivated. The region along the diagonal edge actually possesses a thin “strip” where the junction velocity is actually *lower* than the

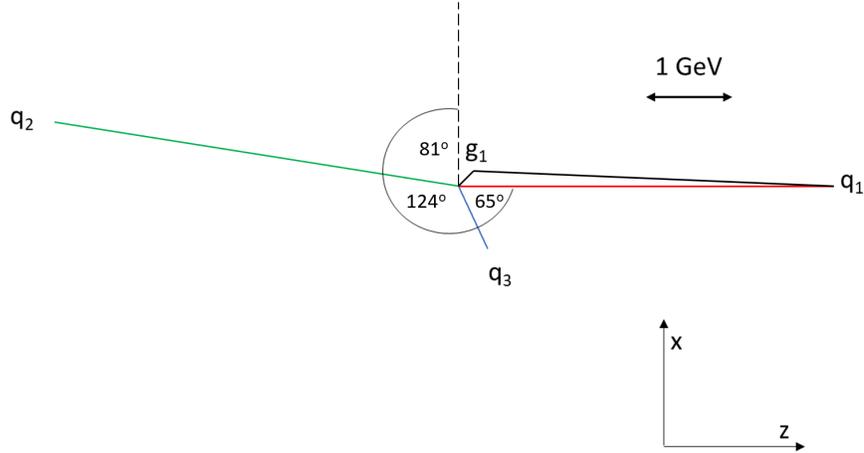


Figure 20. Scaled diagram of the starting junction configuration used for $(x, y) = (0.825, \mathbf{0.125})$. Yields a junction velocity of 0.61. Gluon $g1$ is fixed at a 45° angle from the positive z direction.

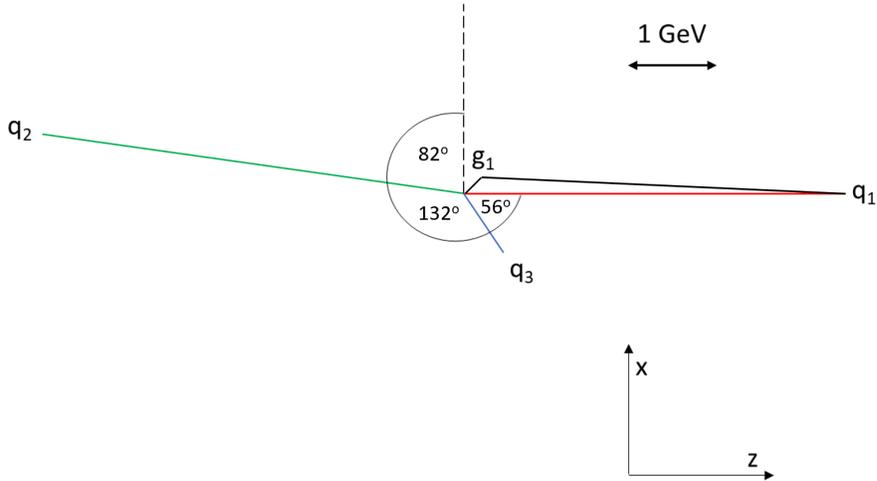


Figure 21. Scaled diagram of the starting junction configuration used for $(x, y) = (0.825, \mathbf{0.135})$. Yields a junction velocity of 0.93. Gluon $g1$ is fixed at a 45° angle from the positive z direction.

area directly below it. This is likely incorrect behaviour, as each Dalitz plot should smoothly tend to junction velocity = 1 at its edges.

Potentially more problematic is the very low velocity region in the bottom left corner for the approximate range of $x \in [0.01, 0.1]$ and $y \in [0.05, 0.3]$. The minimum point of this low-velocity region occurs at $(x, y) = (0.065, 0.145)$, with a velocity value of 0.032. The starting configuration for this particular point is given by Figure 22.

As the velocity values plotted correspond to the required boost velocity to move from the cases' starting configuration (in the lab frame) to the configuration corresponding to the Junction Rest Frame, it is expected that regions in the Dalitz plots with low velocity are a result of starting configurations that are close to the desired “Mercedes” configuration; these starting configurations require only small

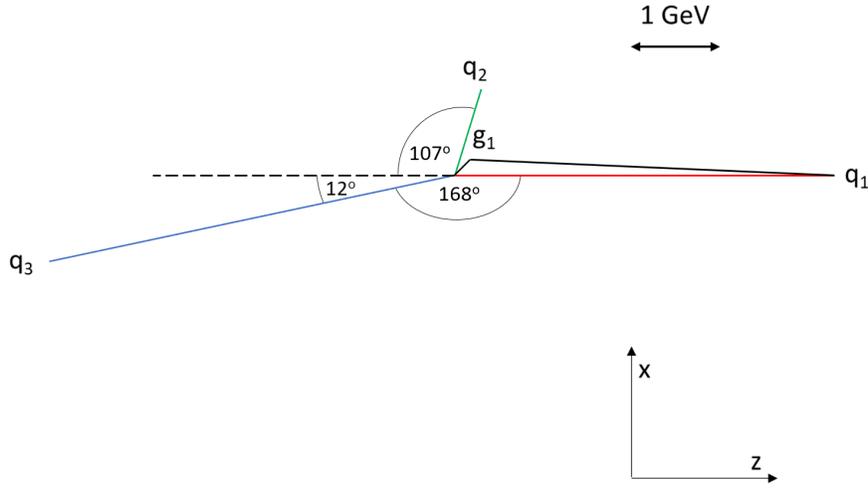


Figure 22. Scaled diagram of the starting junction configuration used for $(x, y) = (0.065, 0.145)$. Yields a junction velocity of 0.032. Gluon $g1$ is fixed at a 45° angle from the positive z direction.

boosts to correctly reorient the legs for the JRF. This is why the “additional” low-velocity region in the new algorithm’s plot is likely erroneously generated: observing Figure 22, the angles between the starting legs are quite large, and would likely require a considerable boost in order to move to a frame where each angle is 120° .

Lastly, the final row in Figure 18, corresponding to the 10 GeV gluon, has no visibly incorrect features within its new algorithm plot. There is some variation between the results of the two algorithms, in the slightly extended low-velocity region for $x \approx 0.1$, y approximately $\in [0.55, 0.6]$, the cause of which may be worth additional investigation.

Overall, however, we see that it is likely that additional work is required to refine the iterative-removal algorithm, where gluons are considered, as the discontinuities and physically unmotivated plot artefacts within Figure 18 suggest some unaddressed sources of error.

6.4.2 Gluon position variation

These sources of error unfortunately only become more evident in results obtained when considering different gluon positions. For this subsection, the gluon energy was fixed at 1 GeV, however its position was varied. Specifically, three cases were considered: The gluon at a polar angle (defined from the positive z axis) of 20° , 90° and 160° . The results of these test cases are shown in Figure 23.

The results from the first row of Figure 23 are predictably both very similar between the two algorithms, and also very similar to the gluonless case. This is because the angle between the gluon’s string “kink” and the leg pointing in the z direction is only 20° , so it does not “pull” the junction in a significantly different direction.

This is not the case, however for the 90° and 160° cases, where the two algorithms produce extremely different results. Firstly, present in the results of the new algorithm is the previously mentioned “band” of high velocity values seemingly superimposed onto the plots. In fact, both the contours within the 90° and 160° new algorithm plots seem to “continue” around these bands: for example in the 160° plot, it appears that the region of $x \in [0, 0.2]$ and $y \in [0, 0.2]$ seems a very natural

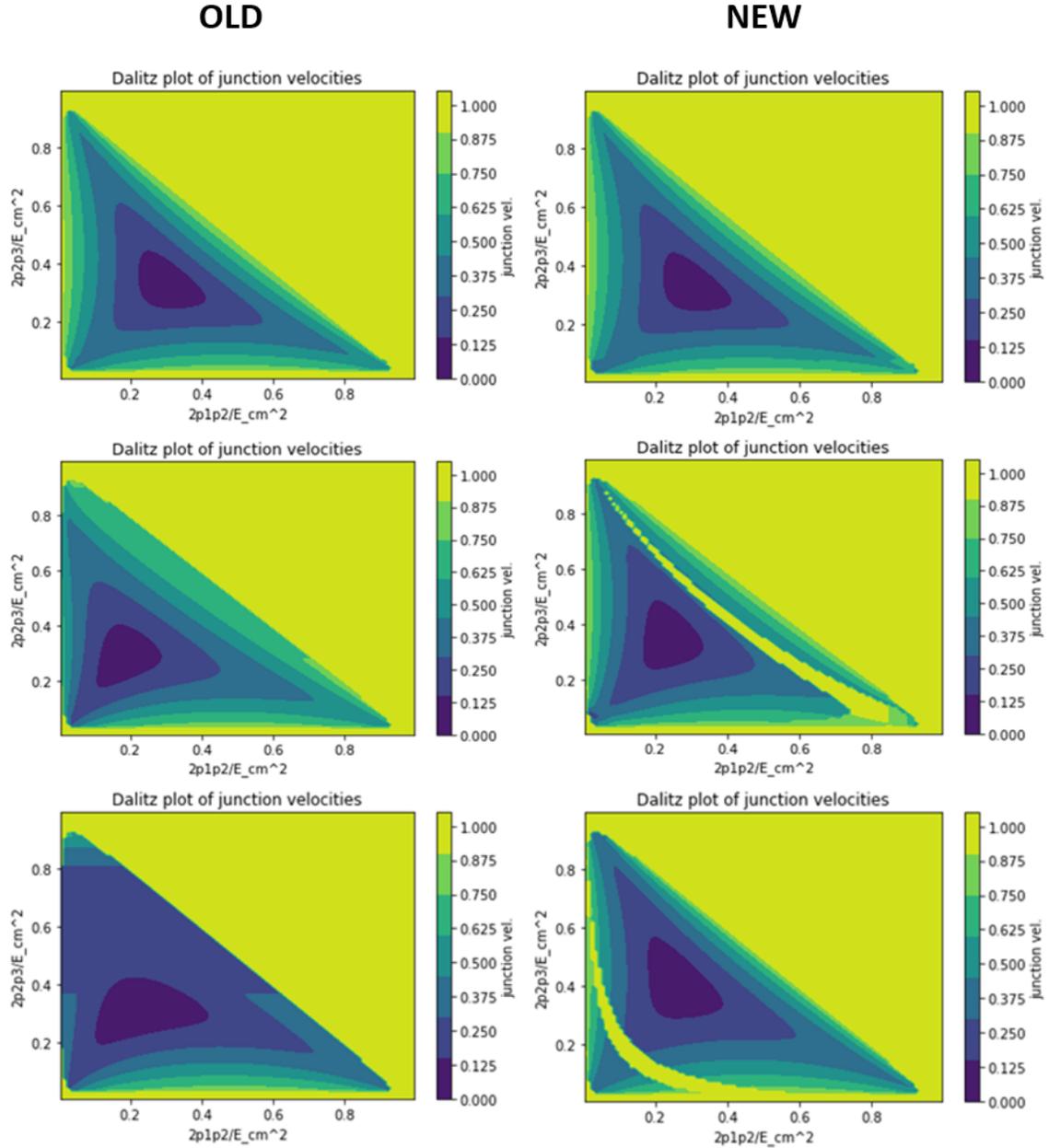


Figure 23. Array of Dalitz plots generated for starting configuration with 1 gluon, with energy 1 GeV. Each is a 100×100 point grid. First row shows system with gluon located at polar angle 20° , second at 90° , third at 160° . Left column is the old algorithm, right is the new algorithm.

extrapolation of the behaviour of the contours for larger x and y above the high-velocity band. Simply, it is very clear that these bands present within the new algorithm’s results are unphysical.

Of encouraging note, however is the comparative success of the new algorithm in displaying more realistic contours, when the band structures are ignored. Particularly in the 160° case, the old algorithm struggles to produce correct results; the large low velocity region from $x \in [0.4, 0.8]$ and $y \in [0, 0.6]$ is very unlikely to be physically motivated. In contrast, the contours of the new plot show a much more reasonable distribution of results with no large near-constant velocity regions outside of the bands.

To conclude this chapter, the results of the iterative-removal algorithm from the listed test cases suggest it may require further refinement before it can consistently produce physically meaningful results when gluons are added. However, there are still promising signs that the iterative-removal algorithm is worth pursuing further, as its extant issues are comprised primarily of isolated plotting artefacts such as unmotivated discontinuities in the results - which should be easily resolved. Without these, it does appear to yield a better treatment of gluons than the old algorithm, in particular when considering systems with gluons located at positions significantly far from the other legs of the junctions. It is for this reason that it would be informative to present some basic analyses of particle production spectra generated from PYTHIA under the new algorithm, to confirm the extent to which it will alter the experimental predictions made by the program. This is done in the next chapter.

7 Predictions from the New Model

7.1 Current successes of the QCD-CR model

7.1.1 CR introduction

The introduction of colour reconnection effects into PYTHIA’s modelling (the QCD-CR model) has yielded predictions with substantial and varied experimental support.

Processes utilising CR were first introduced into PYTHIA as a means of refining the predictions of the mass of the top quark. When modelling events, the effects of non-perturbative QCD proved difficult to quantify, yet made up a significant fraction of the uncertainty within determinations of the top quark mass, judged to be approximately ± 0.5 GeV. Predictions from new “tunes” (specific sets of input parameters used for modelling events) that included CR effects closely matched data from the Tevatron, and thus CR was judged to be a relevant and useful tool for modelling non-perturbative QCD processes [30].

7.1.2 Generator parameters

To yield events with potentially significant input from CR effects, PYTHIA is first set to include *soft*, *inelastic*, *non-diffractive* processes. It is useful to define these terms as these processes form the basis of all the spectra made in this chapter.

Soft QCD corresponds to interactions where there is low momentum transfer between partons (as opposed to “hard” collisions, where momentum is less “dissipated”). Soft interactions play a major role in high-energy collisions, but cannot be described through perturbative-QCD, and thus require a theoretical framework such as CR to model such processes [31].

As discussed in Chapter 2.1, inelastic collisions “break up” the partons within the event into their requisite substructures. This is important when studying processes involving sub-nucleonic particles such as quarks, that are generated from collisions of larger particles like protons. All spectra created in this chapter are formed from pp collisions.

Lastly, diffraction in QCD can be readily understood through the analogy with the more well-known optical diffraction. In optics, waves can “bend” when passing through a gap with size comparable to their wavelength, due to different parts of the wave interfering with each other. When considering QCD, a similar process also occurs with the scattering from particles: when two protons, for example, collide, their sub-components can interfere with each other. This can lead to either or both of them raising to an excited state, which decays into only a few very energetic particles that undergo less central particle interactions. Thus, when analysing the effect of colour reconnections in particle collisions, diffractive events can be ignored.

As briefly discussed in Chapter 5.3, the current iteration of PYTHIA 8 contains three different “modes” of CR effects that it can implement. Each mode comprises a set of slightly different values for each of the parameters input into PYTHIA. The range of values for all modes are shown in full in Appendix A.

The primary difference between the three CR modes concerns their treatment of the time-dilation effects on CR dipoles. In order for colour reconnections to be possible between two string pieces, they must occur in the amount of time between when the strings form, and when they start to hadronise (i.e. break up). This has to also take into account the effects of time-dilation caused from each of the relative boosts of each dipole. Two strings can be defined as being in *causal contact*, allowing colour reconnections to occur, through the condition [23]:

$$\frac{\gamma c}{m_{string} r_{had}} < C_{time} , \quad (7.1)$$

where γ is the relative boost factor, C_{time} is a tunable parameter, and r_{had} is a fixed constant given from the hadronisation time scale.

Subsequently, PYTHIA has three different methods for allowing this process to occur. The first method, dubbed *Mode 0*, does not consider time-dilation constraints. The second method, *Mode 2*, is the “strict” method, requiring both string pieces to be causally connected as per equation (7.1). Lastly, *Mode 3*, is the “loose” method, requiring only one of the dipoles to be causally connected. Generally, Mode 2 is used as the standard tuning for implementing CR effects into PYTHIA, and will be used primarily in the generation of the spectra in this chapter [23].

While most of the individual parameters within each mode were not changed from their values as tabulated in the Appendix, the effect of varying the parameter *StringFlav:probQQ1toQQ0join* was tested. Currently, it is set to the vector (0.0275, 0.0275, 0.0275, 0.0275). This parameter describes the degree to which the formation of spin-1 diquarks within junctions are suppressed (with respect to spin-0). When the values of each mode parameter were first finalised (2015), there were few measurements from *pp* colliders of spin-1 and heavy baryon states, so no reference as to the actual suppression of these diquarks from experiment. Thus, the values for *StringFlav:probQQ1toQQ0join* were taken as the same suppression factor for spin-1 diquarks in ordinary string breaking with light quarks [23].

This choice may very well be not experimentally motivated, however, and to observe the effect of varying this parameter in the potential case of the appearance of new, supportive, experimental data, a new “sub-mode” was created for analysis within this report. In addition to generating spectra within PYTHIA using the conventional Mode 2, results were also obtained using the same mode, but with all of the values within the *StringFlav:probQQ1toQQ0join* vector set to 1. The differences between the spectra generated from this sub-mode, and the regular Mode 2, will also provide an estimate as to the strength of this parameter in affecting the resultant particles generated.

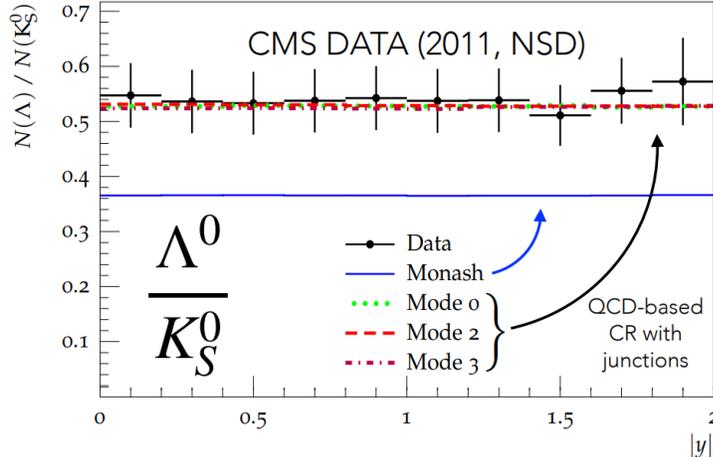


Figure 24. CMS data of Λ^0 baryon (containing uds) number with respect to K_s^0 meson (K-short, containing $\frac{d\bar{s}-s\bar{d}}{\sqrt{2}}$) number, against absolute rapidity $|y|$ [32].

7.1.3 Current progress

Introduction of colour reconnection effects have seen wide success in providing a reliable mechanism to explain the discrepancies in baryon-meson ratios observed in a number of LHC experiments.

We see from data obtained from CMS in 2011 that the QCD-CR model provides a much better fit to the experimental results than the model without CR implemented. The data shown in Figure 24 shows a substantial increase in the expected $\frac{\Lambda^0}{K_s^0}$ fraction, and this increase is well described by the predictions from the QCD-CR model [32]. Note that the “Monash” prediction within the Figure simply corresponds to the results obtained from PYTHIA without employing colour reconnections.

Additionally, the QCD-CR model has seen significant experimental support in observations of Λ_b (containing udb) asymmetry from the LHCb. The Λ_b asymmetry refers to the difference in relative Λ_b^0 and $\bar{\Lambda}_b^0$ cross-sections, with them calculated relative to the total cross section of both baryons. The asymmetry A is given by

$$A = \frac{\sigma(\Lambda_b^0) - \sigma(\bar{\Lambda}_b^0)}{\sigma(\Lambda_b^0) + \sigma(\bar{\Lambda}_b^0)}. \quad (7.2)$$

Once again, the success of implementing the effects of colour reconnections can be readily seen in Figure 25, also again especially at low values of p_\perp [33].

Lastly, predictions from the QCD-CR model have also been very successful at matching results of $\frac{\Lambda_c^+}{D^0}$ production ratios, where here Λ_c^+ is the “charmed” Λ baryon containing udc quarks and D^0 is the D-meson containing $c\bar{u}$. Measurements of p-Pb collisions by ALICE at $\sqrt{s} = 5.02$ TeV and also of pp collisions at $\sqrt{s} = 13$ TeV have both demonstrated strong experimental support for the effects of colour reconnections on Λ_c^+ production, particularly at low values of $p_\perp < 20$ GeV/c [34][35]. Fits of each tuning of the QCD-CR model to both data sets are shown in Figure 26.

The wide experimental support for the predictions obtained from the QCD-CR model strongly indicate that it is a theoretical framework worth pursuing further; at its current stage it is already very successful in accurately modelling a range of observed increases in baryon production. Thus, with the more physically motivated improvements to the treatment of gluons promised by the iterative-removal

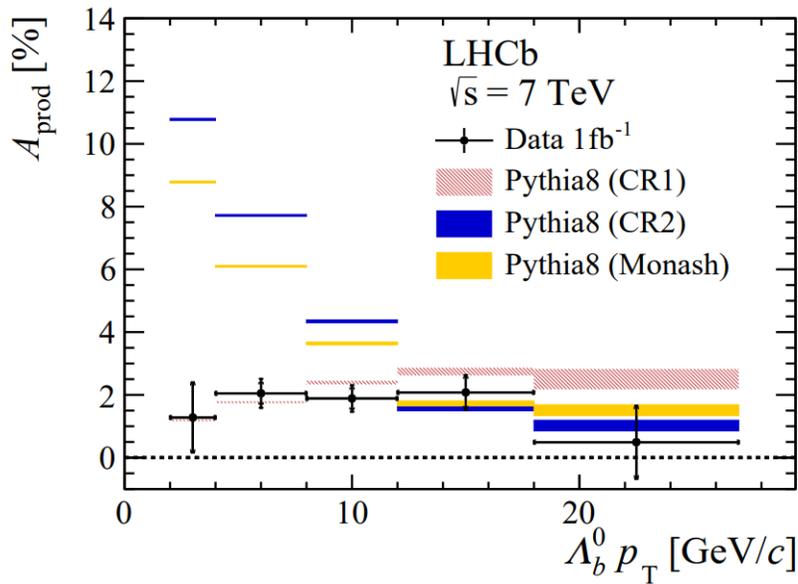


Figure 25. LHCb data of Λ_b asymmetry against p_{\perp} [33]

algorithm, the potential variations to PYTHIA’s predictions introduced from this new scheme are very much worth investigating.

7.2 New scheme modelling

For this report, the focus of the predictions made with PYTHIA’s new algorithm was on the particle production ratios generated from pp collisions at 13 TeV, allowing them to be readily compared with the previously discussed experimental results obtained from ALICE [35].

Each spectra was generated using a simple counting algorithm, that ran a number of events of 13 TeV pp collisions, and incremented a counter every time one of the desired particles, or its anti-particle, appeared in the event. The only constraint on the statistics was the decision to only count particles present with absolute rapidity ($|y| \leq 0.5$), as per ALICE’s data [35]. Broadly, the rapidity of a particle is a measure of its “relativistic velocity”, given by equation (7.3):

$$y = \frac{1}{2} \ln \left(\frac{E + p_z}{E - p_z} \right), \quad (7.3)$$

where E is the energy of the particle and p_z is its momentum along the beam axis [9].

7.2.1 Mode 2 predictions

The two baryon production ratios most easily comparable with our predictions were the Λ_c^+/D_0 and the Σ_c/D_0 . Here Σ_c refers to the group of charmed sigma baryons, encompassing Σ_c^0 (containing ddc quarks), Σ_c^+ (udc) and Σ_c^{++} (uuc). The experimental data from ALICE for both of these ratios was overlaid onto the predictions from PYTHIA for both the new and old algorithms. These comparisons are shown in Figures 27 and 28.

The histogram shown in Figures 27 and 28 both use mode 2, as it is deemed to be the most accurate within PYTHIA and “conventional” mode used for the QCD-CR model. To generate the

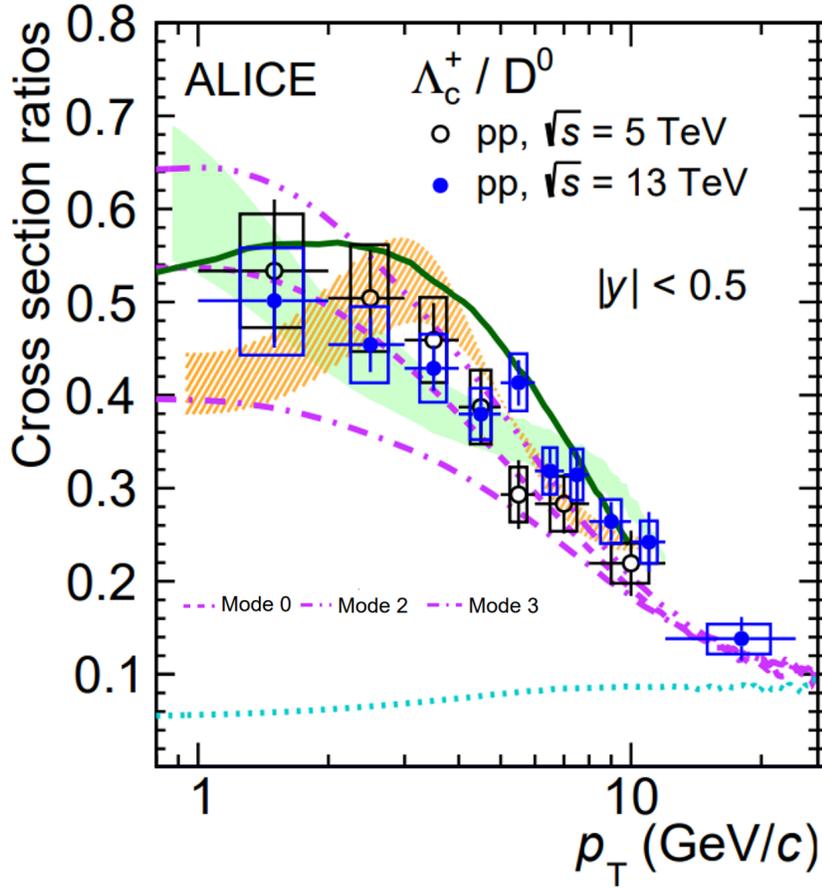


Figure 26. ALICE data of $\frac{\Lambda_c^+}{D^0}$ production ratio obtained from both $\sqrt{s} = 5.02$ and 13 TeV experiments [35].

figures, individual histograms were created displaying the number of Λ_c^+ , Σ_c and D_0 registered from the 4,000,000 events. The vertical error in each bin was taken as the standard statistical error, $\pm\sqrt{n}$, where n is the number of counts in each bin of the histogram. To produce the ratios shown, the count values in the datasets for each particle were divided by each other, and their requisite error was also propagated.

Observing the two figures, it is somewhat problematic to draw any solid conclusions. The spectra generated from the new and old models are very different, which reinforces the importance of the revisions made in our new model: a different treatment of gluons does appear to greatly impact the results obtained from PYTHIA’s modelling.

Specifically, the new algorithm displays a much more pronounced inverse-exponential-like shape in both Figures, with it tending towards zero for high p_\perp and seemingly only reaching a point of inflection at very low p_\perp , approximately the 0.5-1 GeV bin. In comparison, the “rate of decay” of the old algorithm’s spectra is far less apparent, and it appears to reach a local maximum in the region of 1-2 GeV in both plots.

The slower rate of decay of the old algorithm causes its $\frac{\Lambda_c^+}{D_0}$ spectra to quite closely align with the spectra generated from ALICE, matching nearly all data points within one standard deviation of

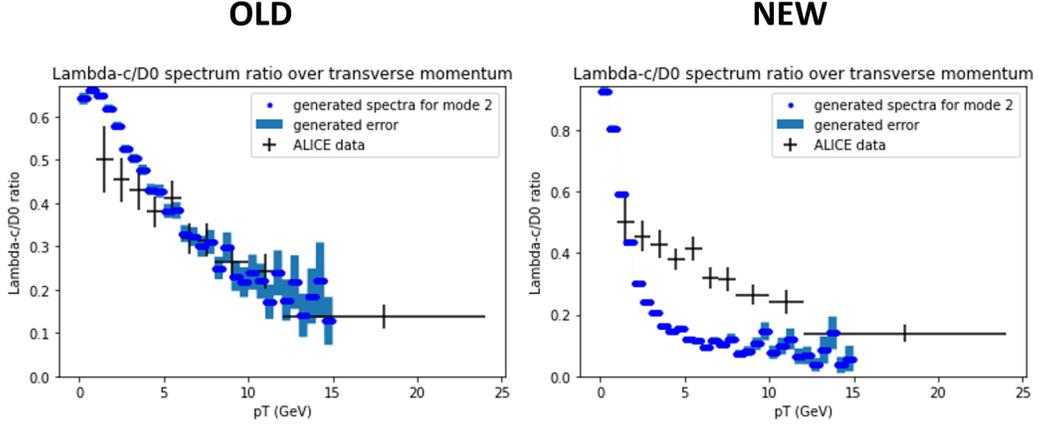


Figure 27. Comparison of old and new spectra of $\frac{\Lambda_c^+}{D_0}$ ratio, against p_\perp . Generated from PYTHIA, running over 4,000,000 events. Width of each point corresponds to the width of each histogram bin (0.5 GeV), height the standard statistical error propagated for each particle. Overlaid is the data obtained from ALICE, with systematic and statistical uncertainties added in quadrature.

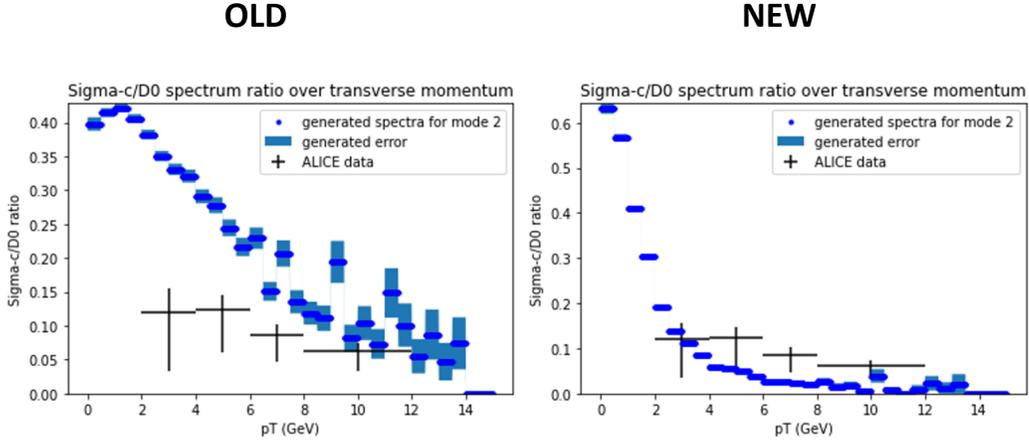


Figure 28. Comparison of old and new spectra of $\frac{\Sigma_c}{D_0}$ ratio, against p_\perp . Generated from PYTHIA, running over 4,000,000 events. Width of each point corresponds to the width of each histogram bin (0.5 GeV), height the standard statistical error propagated for each particle. Overlaid is the data obtained from ALICE, with systematic and statistical uncertainties added in quadrature.

uncertainty. In comparison, the behaviour suggested from the new algorithm is very different than what is predicted by the data, displaying what could be interpreted as an almost completely different relationship.

Regarding the $\frac{\Sigma_c}{D_0}$ spectra, exactly the opposite is true. The more linear-like behaviour of the old algorithm causes its predictions to far exceed that which have been observed from the data, however the sharp decay of the new algorithm causes its predicted spectra to fairly closely align with ALICE, again within one standard deviation of uncertainty for all but one data point.

A possible source of error in these predictions may arise from the specific means by which the Λ_c^+

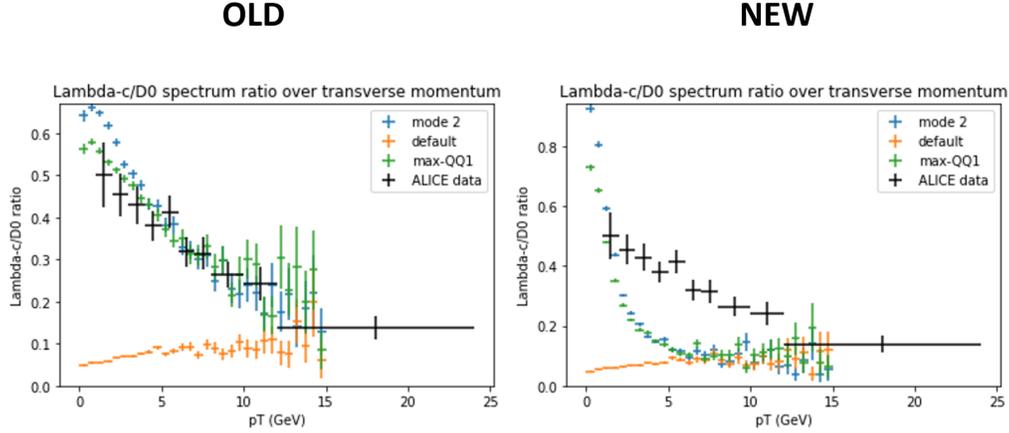


Figure 29. Comparison of old and new spectra of $\frac{\Lambda_c^+}{D_0}$ ratio, against p_\perp . Results generated from mode 2, max-QQ1 case and default all shown. Data from ALICE also overlaid. Histogram bins shown as simple errorbars so plot is more comprehensible.

were counted. Every Σ_c baryon encountered when reading from the event record was included in the count used for the spectrum. This neglects the possibility that some Σ_c may decay into the Λ_c^+ , in addition to the Λ_c^+ produced from the primary pp collision. In contrast, the data from ALICE only measured “prompt” Σ_c baryons, particles only produced either in the primary interaction or not as a decay product from a long-lived ($> 3 \times 10^{-11}$ seconds) previous particle [36]. Accounting for this discrepancy would likely yield a slightly lower $\frac{\Lambda_c^+}{D_0}$ spectrum generated from PYTHIA, which in the case of the old algorithm would potentially better align with ALICE, but in the new algorithm is unlikely to improve its predictions.

Clearly, it can not be concluded that the new algorithm as it currently exists is not comprehensively successful at generating spectra that more closely match with experiment. The results for the $\frac{\Sigma_c}{D_0}$ ratios are promising, and do align better than the old algorithm, however the $\frac{\Lambda_c^+}{D_0}$ predictions present a significant deviation from the experimental data, to an extent that is quite probably the result of intrinsic error present within the new algorithm. Once again, it is likely that further refinement of PYTHIA’s new model is required.

7.2.2 Comparison of modes

Also tested in this report is the effect of varying the previously mentioned *StringFlav:probQQ1toQQ0join* parameter (hereafter “QQ1” for brevity). To analyse the “strength” of this parameter in influencing PYTHIA’s results, as before we generated Figures of the $\frac{\Lambda_c^+}{D_0}$ and $\frac{\Sigma_c}{D_0}$ ratios comparing the spectra obtained from PYTHIA’s conventional mode 2 tuning, the “maximum QQ1 case” with all of its components set to 1, and the default, non-CR, case as a baseline point of reference. The results of these are shown in Figures 29 and 30.

The results obtained from Figures 29 and 30 are quite promising. The overall effect of varying the QQ1 parameter causes the overall values of the ratio at each p_\perp to decrease. However in the old algorithm for both baryon production spectra, this decrease results in a much better alignment with the data: For the $\frac{\Lambda_c^+}{D_0}$ ratio, the predictions from the *max-QQ1* case now agree with every data point

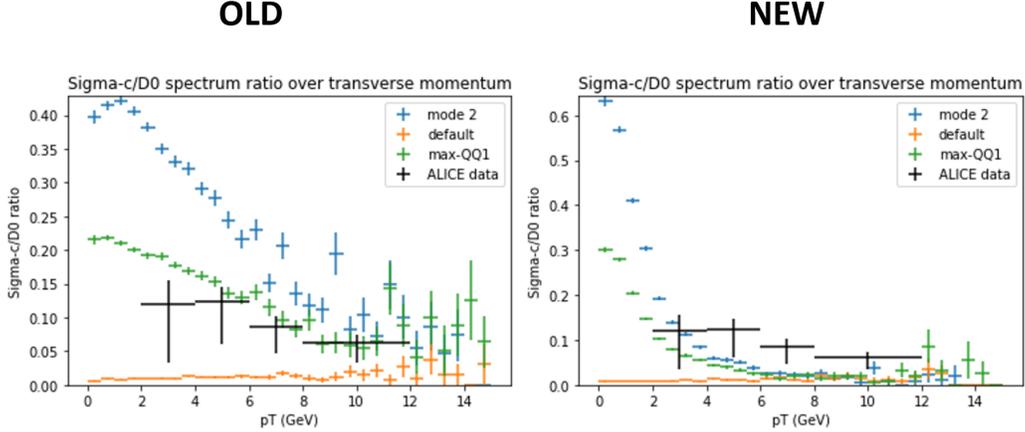


Figure 30. Comparison of old and new spectra of $\frac{\Sigma_c}{D_0}$ ratio, against p_\perp . Results generated from mode 2, max-QQ1 case and default all shown. Data from ALICE also overlaid. Histogram bins shown as simple errorbars so plot is more comprehensible.

generated from ALICE within a single deviation of uncertainty. Likewise for the $\frac{\Sigma_c}{D_0}$ spectrum, the *max* – *QQ1* predictions align with each of the data points except one within one standard deviation of uncertainty, a significant improvement on the results generated from the standard tuning of mode 2.

While the decrease in ratio values does slightly further increase the discrepancy between the predicted spectra from the new algorithm and the data, as previously stated this discrepancy may well have arisen from existing error within the new model, and the revaluation of the QQ1 parameter has done little to alleviate this issue. Thus, it seems likely that a higher-valued QQ1 parameter may very well better agree with the data, than a low-valued one, and the modification of PYTHIA’s standard mode 2 tuning to account for this may be an initiative worth further pursuing.

7.3 Additional predictions: Other hadron spectra

To provide a brief outlook beyond the scope of this report, a few potential areas of investigation will be discussed. Firstly, while data on Λ_c^+ and Σ_c over p_\perp are well-tabulated by ALICE, the spectra of other hadrons, in particular heavy baryons, have not been as comprehensively documented.

A charmed hadron less represented within the literature is the Σ_c^* (“Sigma-star”), a group of three baryons with the same constituent quarks as the three Σ_c baryons, but with total spin 3/2, unlike the Σ_c baryons with 1/2. Predictions from PYTHIA of the $\frac{\Sigma_c^*}{D_0}$ ratio spectrum for each mode are shown in Figure 31.

Of interest in Figure 31 is how, unlike for the $\frac{\Lambda_c^+}{D_0}$ or $\frac{\Sigma_c}{D_0}$ spectra, the effect of increasing the QQ1 parameter *greatly* increases the overall yield of Σ_c^* produced, by approximately 400% for both the old and new algorithm at low p_\perp . An accurate measurement of this baryon production ratio would thus be instrumental in conclusively demonstrating whether a high QQ1 parameter is the better predictor of experiment. Given the wide difference in both algorithms between the generated spectra from mode 2 and from the *max* – *QQ1* case, data closely aligning to either prediction would provide further much-needed experimental evidence regarding the effect of this parameter.

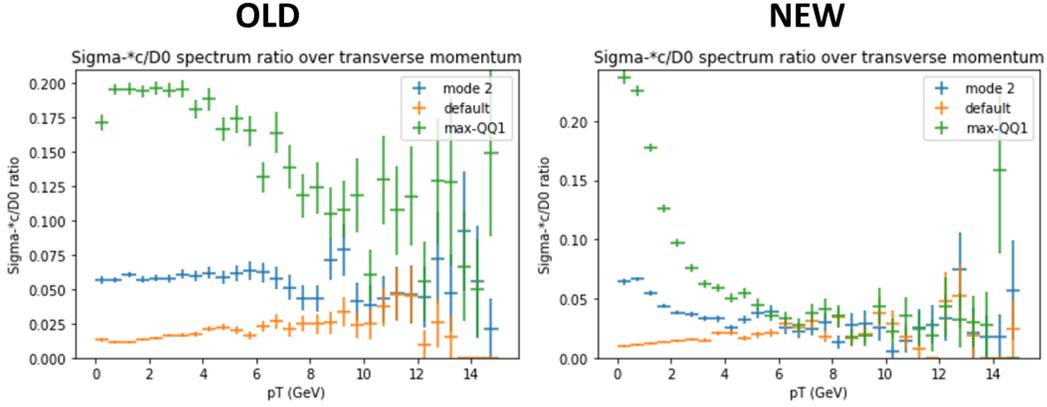


Figure 31. Comparison of old and new spectra of $\frac{\Sigma_c^*}{D_0}$ ratio, against p_\perp . Results generated from mode 2, max-QQ1 case and default all shown.

An additional area of potential future research might involve production spectra of heavier hadrons such as those containing b quarks. Due to their larger mass, even at rest these quarks have energies far larger than the QCD energy scale Λ_{QCD} and their analysis can thus be used as a means of probing the behaviour of perturbative QCD models at low momenta [37].

As baryons and mesons containing b quarks are also much harder to produce given the mass of the b , developing a well-functioning system for statistically predicting their production rates may prove very useful in further studies of such hadrons. Figure 32 compiles three different production ratios generated from both algorithms. The first $\frac{\Lambda_b^0}{B_\pm}$, where Λ_b^0 is a baryon containing udb , and B_\pm , a meson containing $u\bar{b}$. The second ratio is $\frac{\Sigma_b}{B_\pm}$, where Σ_b corresponds to a group of baryons containing Σ_b^+ (uub), Σ_b^0 (udb) and Σ_b^- (ddb). The last ratio $\frac{\Sigma_b^*}{B_\pm}$ also contains the Σ_b^* baryons, with the same constituent quarks as the Σ_b but with total spin $3/2$.

Most interesting to note about Figure 32 is how the behaviour in each of the individual spectra mimics the behaviour of each of the corresponding “charmed” baryons. For example, like the $\frac{\Lambda_c^+}{D_0}$ spectra, the $\frac{\Lambda_b^0}{B_\pm}$ predictions also show a “shallower” decay rate with respect to p_\perp in the old model, but a steeper inverse-exponential like behaviour in the new model. As before, the predictions generated in the $max - QQ1$ case yield a slightly smaller ratio at low p_\perp in both algorithms.

The general behaviour of the results shown in the second and third row of Figure 32 can also be seen to be very similar to the corresponding spectra generated for the “charmed” Σ baryons in Figures 28 and 31. This is potentially a very useful result: It may suggest that both the charm and heavy baryon production ratios can be theoretically described by very similar models, which may in turn present an opportunity to make predictions of these heavy hadron spectra with greater ease.

7.4 Multiplicity dependence

Another potential area of further investigation could involve the study of how different parameters in the pp collision could affect the generated spectra. In this subsection some preliminary results are presented showing the dependence of the spectra on the total *charged multiplicity* of the products formed from the collision, however many other constraints may also be worth studying.

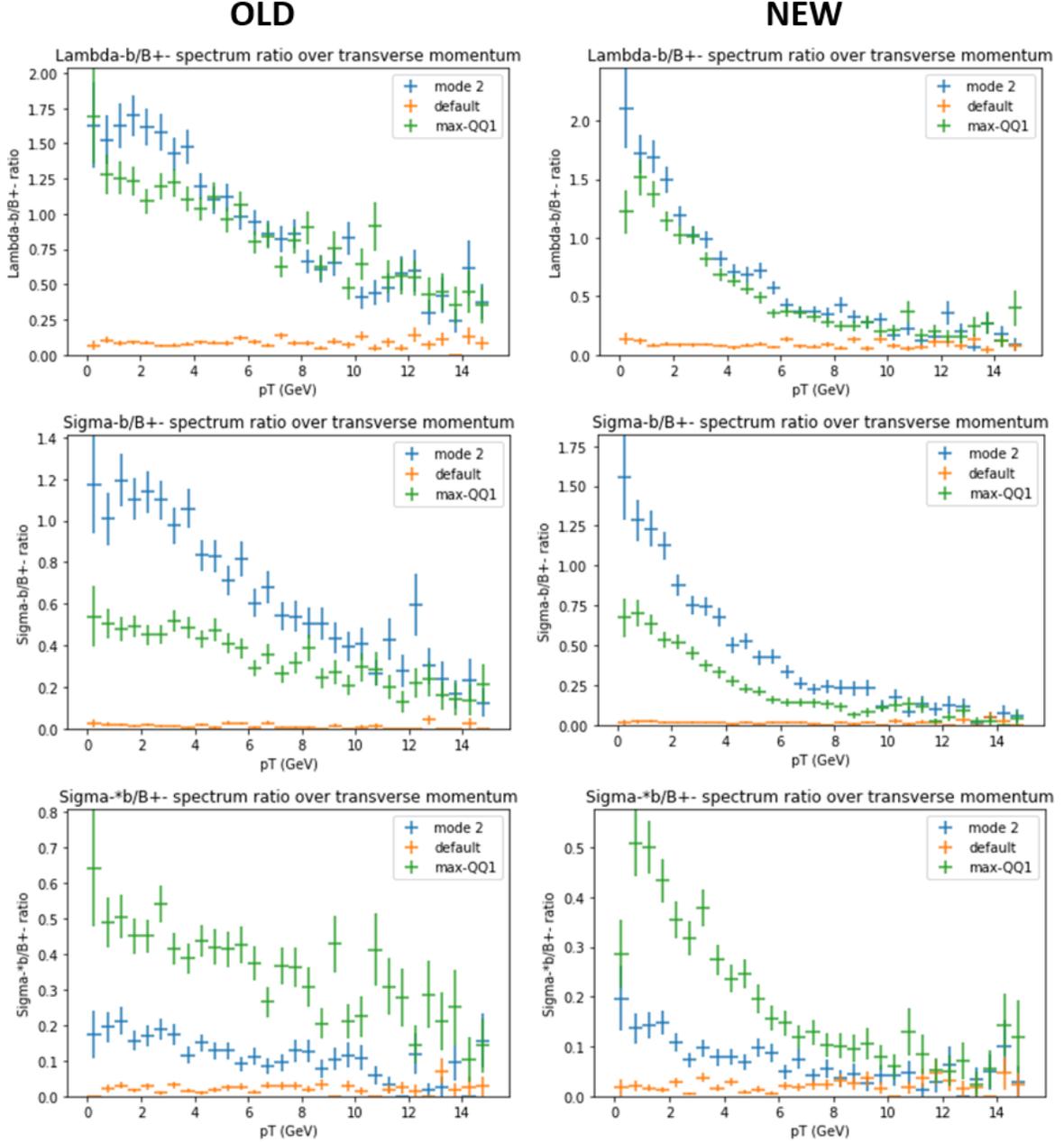


Figure 32. Comparison of old and new spectra of $\frac{\Lambda_b^0}{B_{\pm}^0}$ ratio, $\frac{\Sigma_b}{B_{\pm}}$ and $\frac{\Sigma_b^*}{B_{\pm}^*}$ all against p_{\perp} . Results generated from mode 2, max-QQ1 case and default all shown.

The charged multiplicity simply refers to the number of final-state (not decaying into another particle), charged particles present in each event. It is these particles that are actually measured in experimental particle collisions, and thus testing whether the number observed affects the amount of resultant baryon production may be very useful.

To show the variation of the spectra with charged multiplicity, the previously shown histograms are divided into three separate cases each. The first case considers the “low-range multiplicity” events, and only counted hadrons produced in events with charged multiplicity < 10 . The second case considers the “mid-range multiplicity”, counting particles with charged multiplicity > 10 but < 30 , and the final case considers the “high-range multiplicity” particles from events with charged multiplicity > 60 . In calculation of the charged multiplicity, only partons with absolute pseudorapidity $|\eta| < 0.5$ were counted, so the value of the charged multiplicity matched the constraint of $|y| < 0.5$ within the previous spectra.

Pseudorapidity is an approximation to rapidity in the case where the jet mass is small with respect to the jet energy and $p_z \approx E \cos \theta$. θ is the polar angle with respect to the beam axis. Rearranging using trigonometric identities, pseudorapidity $|\eta|$ is defined by equation (7.4). The constraint on pseudorapidity was required for counting the charged particles as their mass was not determined.

$$\eta = \frac{1}{2} \ln (\cot^2 (\theta/2)) . \quad (7.4)$$

Figure 33 compiles the results of the low, mid and high-range multiplicity histograms for both the old and new algorithms within PYTHIA, again plotting the generated $\frac{\Lambda_c^+}{D_0}$ ratio against p_\perp .

The results of Figure 33 are in line with what could be reasonably expected. While there is some minor variation in the overall shape of the individual “curves” on each figure, the slow-decay rate of the $\frac{\Lambda_c^+}{D_0}$ ratio in the old algorithm, and the more prominent decay rate of it in the new algorithm, which were discussed in the previous chapter, are still prominent within each range of charged multiplicity.

There are two main differences between the spectra generated at each multiplicity range. One is the variation of high- p_\perp error: some panels, such as the new algorithm mid-range multiplicity have relatively minimal error even at the highest p_\perp bins. Others, like the new algorithm low-range multiplicity panel, have in their high p_\perp data points an error so large its resultant errorbars span more than $4\times$ the height of the original “peak” at low p_\perp . As the error in the histograms is simply the standard statistical error (\sqrt{n}), its larger value is due to a higher number of particles counted in each bin. This larger yield is likely due to the low-range charged multiplicity events appearing more frequently than those with higher multiplicity, as would likely be expected for a large sample of events.

Secondly, the increased relative yield of Λ_c^+ baryons for higher multiplicity ranges is apparent in both the old and new algorithms. While the low-range multiplicity panel “peaks” at a ratio of approximately 0.45, the mid-range peaks at ~ 0.75 and the high range at ~ 1 . This is likely because higher multiplicities result in more MPIs, which allow more CR interactions, and thus the potential for more junctions to form and create baryons. As before, this conclusion agrees with what would be intuitively expected from a comparison of events of different charged multiplicity.

Ultimately, the analysis of the effect on the predictions of production spectra from the variation of other parameters within the events may well be an area of investigation worth pursuing. Observation of these effects within experiment would also likely prove informative in furthering our understanding of the more precise processes by which these hadrons form.

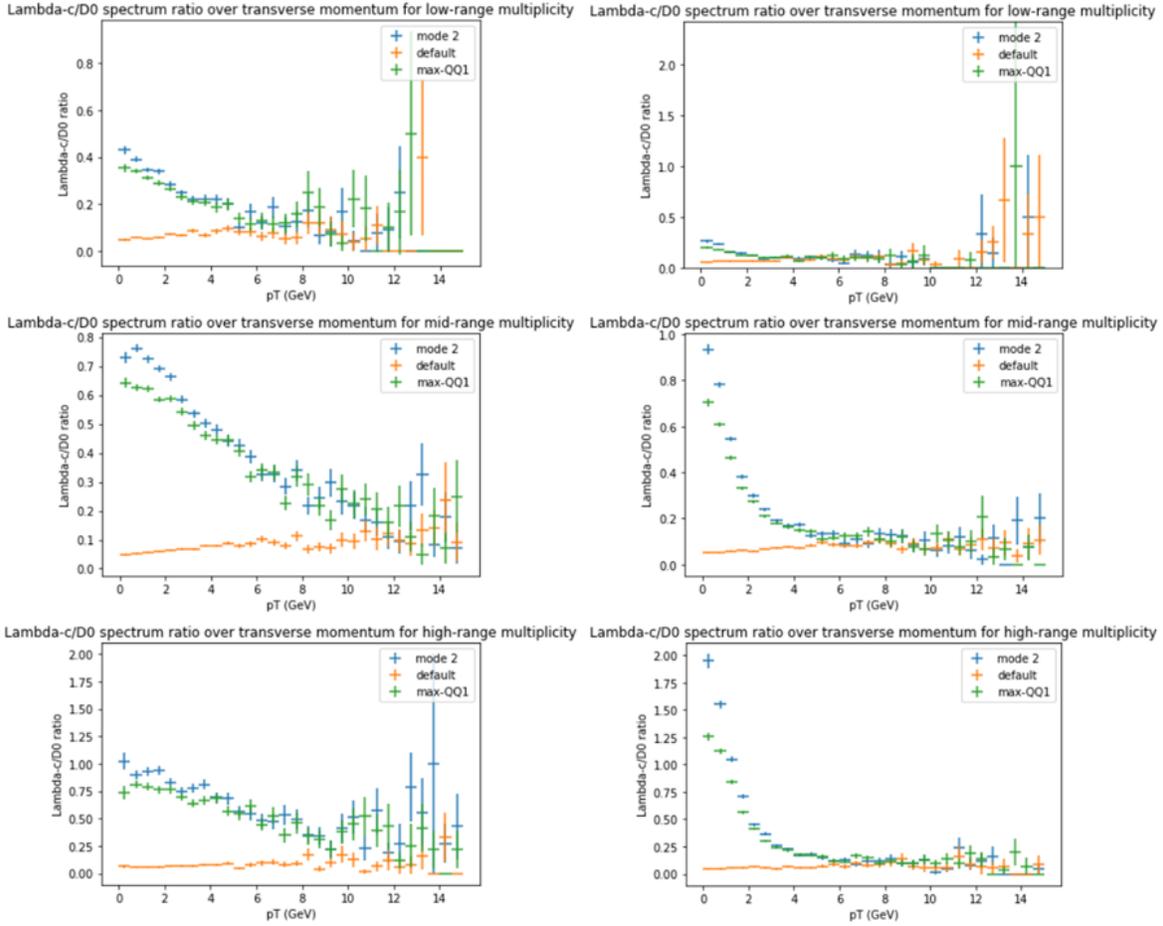
OLD**NEW**

Figure 33. Comparison of old and new spectra of $\frac{\Lambda_c^+}{D_0}$ ratio against p_{\perp} . Individual panels included for the case of low, mid and high-range charged multiplicity. Results generated from mode 2, max-QQ1 case and default all shown in each panel.

8 Conclusion

A number of the new developments implemented into the QCD-CR model have yielded promising results in improving PYTHIA's treatment of gluons within string junctions. The errors in its current pull-vector algorithm that resulted in approximately 10% of events generating incorrectly have theoretically been rectified with the new iterative-removal algorithm. This new algorithm presents a more physically meaningful and intuitive method of fragmenting junctions containing gluons, and, once finalised, is very likely to consistently produce more accurate predictions for hadron production.

There remains further work to be completed on the iterative-removal algorithm, however. Many of the controlled tests performed on it highlighted behaviour for certain starting systems that was unlikely to be physically motivated, such as large resultant junction velocity discontinuities between similar systems, or certain configurations where the new algorithm likely does not reach a junction velocity

that would be expected given the input systems. In evaluating the new algorithm, the accuracy of PYTHIA's predictions of particle production spectra is difficult to determine, as there is no conclusively improved alignment of the predictions with existing experimental data.

Thus, it is clear that the improvements to the QCD-CR model are very much still in need of refinement before they can be reliably implemented into PYTHIA. However, once finalised, they will hopefully allow for more accurate theoretical modelling of many important parton production processes, and further develop our understanding of hadronisation through the Lund String Model.

9 Acknowledgements

I would like to extend my everlasting thanks to my supervisor Peter Skands, who managed somehow to facilitate my increasingly incessant demands for assistance and consultation: Without him it is indubitable this project would have been impossible on a multitude of levels. A great thank you is also due PhD student Javira Altmann, who was responsible for the invention and physical design of the new iterative-removal algorithm, and must have possessed a great deal of patience to be able to put up with my constant questions and inquiries about its function.

References

- [1] M. Gell-Mann, *A schematic model of baryons and mesons*, *Physics letters* **8** (1964) 214.
- [2] G. Zweig, *An $SU(3)$ model for strong interaction symmetry and its breaking. Version 2*, pp. 22–101. 2, 1964.
- [3] D. J. Griffiths, *Introduction to elementary particles*, Physics textbook. Wiley-VCH, Weinheim, 2nd, rev. ed., 2010.
- [4] C. Bierlich, *Rope Hadronization, Geometry and Particle Production in pp and pA Collisions*, Ph.D. thesis, Lund U., 2017.
- [5] C. Bierlich et al., *A comprehensive guide to the physics and usage of PYTHIA 8.3*, [2203.11601](#).
- [6] M. Bahr et al., *Herwig++ Physics and Manual*, *Eur. Phys. J. C* **58** (2008) 639 [[0803.0883](#)].
- [7] SHERPA collaboration, *Event Generation with Sherpa 2.2*, *SciPost Phys.* **7** (2019) 034 [[1905.09127](#)].
- [8] M. H. Seymour and M. Marx, *Monte Carlo Event Generators*, in *69th Scottish Universities Summer School in Physics: LHC Physics*, pp. 287–319, 4, 2013, DOI [[1304.6677](#)].
- [9] M. Thomson, *Modern particle physics*. 2013.
- [10] G. L. Kane, *Modern elementary particle physics*. Addison Wesley, Advanced Book Program, Redwood City, Calif., 1987.
- [11] C. Bierlich et al., *PYTHIA 8 online manual*, 2022.
- [12] CMS collaboration, *Measurement of the top quark mass using events with a single reconstructed top quark in pp collisions at $\sqrt{s} = 13$ TeV*, *JHEP* **12** (2021) 161 [[2108.10407](#)].
- [13] E. Eichten, K. Gottfried, T. Kinoshita, J. Kogut, K. D. Lane and T. M. Yan, *Spectrum of charmed quark-antiquark bound states*, *Physical review letters* **34** (1975) 369.
- [14] P. Skands, *Introduction to QCD*, in *Theoretical Advanced Study Institute in Elementary Particle Physics: Searching for New Physics at Small and Large Scales*, pp. 341–420, 2013, DOI [[1207.2389](#)].
- [15] S. Ferreres-Solé and T. Sjöstrand, *The space-time structure of hadronization in the lund model*, *The European physical journal. C, Particles and fields* **78** (2018) 983.

- [16] C. Bierlich, S. Chakraborty, G. Gustafson and L. Lönnblad, *Hyperfine splitting effects in string hadronization*, *The European physical journal. C, Particles and fields* **82** (2022) 1.
- [17] C. B. Duncan and P. Skands, *Fragmentation of Two Repelling Lund Strings*, *SciPost Phys.* **8** (2020) 080 [[1912.09639](#)].
- [18] J. Schwinger, *On gauge invariance and vacuum polarization*, *Physical review* **82** (1951) 664.
- [19] C. Bierlich, G. Gustafson, L. Lönnblad and A. Tarasov, *Effects of Overlapping Strings in pp Collisions*, *JHEP* **03** (2015) 148 [[1412.6259](#)].
- [20] T. Sjöstrand and P. Z. Skands, *Baryon number violation and string topologies*, *Nucl. Phys. B* **659** (2003) 243 [[hep-ph/0212264](#)].
- [21] P. Bartalini et al., *Multi-Parton Interactions at the LHC*, **11**, 2011, [1111.0469](#).
- [22] U. Egede, T. Hadavizadeh, M. Singla, P. Skands and M. Vesterinen, *The role of multi-parton interactions in doubly-heavy hadron production*, [2205.15681](#).
- [23] J. R. Christiansen and P. Z. Skands, *String Formation Beyond Leading Colour*, *JHEP* **08** (2015) 003 [[1505.01681](#)].
- [24] CMS collaboration, *Evidence for collectivity in pp collisions at the LHC*, *Phys. Lett. B* **765** (2017) 193 [[1606.06198](#)].
- [25] R. H. Dalitz, *On the analysis of tau-meson data and the nature of the tau-meson*, *Phil. Mag. Ser. 7* **44** (1953) 1068.
- [26] PARTICLE DATA GROUP collaboration, *Review of Particle Physics*, *Chin. Phys. C* **40** (2016) 100001.
- [27] W. T. Giele, D. A. Kosower and P. Z. Skands, *A simple shower and matching algorithm*, *Phys. Rev. D* **78** (2008) 014026 [[0707.3652](#)].
- [28] P. Erdős, *The Mathematics of Paul Erdős*, Algorithms and Combinatorics, 14. 1st ed., 1997.
- [29] A. Gehrmann-De Ridder, M. Ritzmann and P. Z. Skands, *Timelike Dipole-Antenna Showers with Massive Fermions*, *Phys. Rev. D* **85** (2012) 014013 [[1108.6172](#)].
- [30] D. Wicke and P. Z. Skands, *Non-perturbative QCD Effects and the Top Mass at the Tevatron*, *Nuovo Cim. B* **123** (2008) S1 [[0807.3248](#)].
- [31] PARTICLE DATA GROUP collaboration, *Review of Particle Physics*, *PTEP* **2020** (2020) 083C01.
- [32] CMS collaboration, *Strange Particle Production in pp Collisions at $\sqrt{s} = 0.9$ and 7 TeV*, *JHEP* **05** (2011) 064 [[1102.4282](#)].
- [33] LHCb collaboration, *Observation of a $\Lambda_b^0 - \bar{\Lambda}_b^0$ production asymmetry in proton-proton collisions at $\sqrt{s} = 7$ and 8 TeV*, *JHEP* **10** (2021) 060 [[2107.09593](#)].
- [34] ALICE collaboration, *Λ_c^+ production in pp and in p-Pb collisions at $\sqrt{s_{NN}}=5.02$ TeV*, *Phys. Rev. C* **104** (2021) 054905 [[2011.06079](#)].
- [35] ALICE collaboration, *Measurement of Prompt D^0 , Λ_c^+ , and $\Sigma_c^{0,++}(2455)$ Production in Proton-Proton Collisions at $\sqrt{s} = 13$ TeV*, *Phys. Rev. Lett.* **128** (2022) 012001 [[2106.08278](#)].
- [36] LHCb collaboration, *Measurement of prompt charged-particle production in pp collisions at $\sqrt{s} = 13$ TeV*, *JHEP* **01** (2022) 166 [[2107.10090](#)].
- [37] ALICE collaboration, *Charm and beauty production and hadronization with the ALICE experiment*, **11**, 2022, [2211.03720](#).

A Parameter tunings of each mode

Shown here is a compilation of the value of each parameter, within each CR mode of PYTHIA. The “Monash” mode is simply the default mode.

Parameter	Monash	Mode 0	Mode 2	Mode 3
StringPT:sigma	= 0.335	= 0.335	= 0.335	= 0.335
StringZ:aLund	= 0.68	= 0.36	= 0.36	= 0.36
StringZ:bLund	= 0.98	= 0.56	= 0.56	= 0.56
StringFlav:probQQtoQ	= 0.081	= 0.078	= 0.078	= 0.078
StringFlav:ProbStoUD	= 0.217	= 0.2	= 0.2	= 0.2
StringFlav:probQQ1toQQ0join	= 0.5, 0.7, 0.9, 1.0	= 0.0275, 0.0275, 0.0275, 0.0275	= 0.0275, 0.0275, 0.0275, 0.0275	= 0.0275, 0.0275, 0.0275, 0.0275
MultiPartonInteractions:pT0Ref	= 2.28	= 2.12	= 2.15	= 2.05
BeamRemnants:remnantMode	= 0	= 1	= 1	= 1
BeamRemnants:saturation	-	= 5	= 5	= 5
ColourReconnection:mode	= 0	= 1	= 1	= 1
ColourReconnection:allowDoubleJunRem	= on	= off	= off	= off
ColourReconnection:m0	-	= 2.9	= 0.3	= 0.3
ColourReconnection:allowJunctions	-	= on	= on	= on
ColourReconnection:junctionCorrection	-	= 1.43	= 1.20	= 1.15
ColourReconnection:timeDilationMode	-	= 0	= 2	= 3
ColourReconnection:timeDilationPar	-	-	= 0.18	= 0.073

B Coding syntax of old and new algorithms

Details of the JRF finding section of *fragmentToJunction* within the OLD algorithm. Coding done in C++.

```

// Fragment off two of the string legs in to a junction.
bool StringFragmentation::fragmentToJunction(Event& event) {

    // Identify range of partons on the three legs.
    // (Each leg begins with an iParton[i] = -(10 + 10*junctionNumber + leg),
    // and partons then appear ordered from the junction outwards.)
    int legBeg[3] = { 0, 0, 0};
    int legEnd[3] = { 0, 0, 0};
    int leg = -1;
    // PS (4/10/2011) Protect against invalid systems
    if (iParton[0] > 0) {
        infoPtr->errorMsg("Error in StringFragmentation::fragment"
            "ToJunction: iParton[0] not a valid junctionNumber");
        return false;
    }
    for (int i = 0; i < int(iParton.size()); ++i) {
        if (iParton[i] < 0) {
            if (leg == 2) {
                infoPtr->errorMsg("Error in StringFragmentation::fragment"
                    "ToJunction: unprocessed multi-junction system");
                return false;
            }
            legBeg[++leg] = i + 1;
        }
        else legEnd[leg] = i;
    }

    // Iterate from system rest frame towards the junction rest frame (JRF).
    RotBstMatrix Mstep;
    MtoJRF.reset();
    MtoJRF.bstback(pSum);
    Vec4 pWTinJRF[3];
    int iter = 0;
    double errInCM = 0.;

    do {
        ++iter;
        // Find weighted sum of momenta on the three sides of the junction.
        for (leg = 0; leg < 3; ++leg) {
            pWTinJRF[leg] = 0.;
            double eWeight = 0.;
            for (int i = legBeg[leg]; i <= legEnd[leg]; ++i) {
                Vec4 pTemp = event[ iParton[i] ].p();
                pTemp.rotbst(MtoJRF);
                pWTinJRF[leg] += pTemp * exp(-eWeight);
                eWeight += pTemp.e() / eNormJunction;
                if (eWeight > EJNWEIGHTMAX) break;
            }
        }

        // Store original deviation from 120 degree topology.
        if (iter == 1) errInCM = pow2(costheta(pWTinJRF[0], pWTinJRF[1]) + 0.5)
            + pow2(costheta(pWTinJRF[0], pWTinJRF[2]) + 0.5)
            + pow2(costheta(pWTinJRF[1], pWTinJRF[2]) + 0.5);

        // Check numerical instabilities in boost, use rest frame if it fails.
        if ( (pWTinJRF[0] + pWTinJRF[1]).m2Calc() < M2MINJRF
            || (pWTinJRF[0] + pWTinJRF[2]).m2Calc() < M2MINJRF
            || (pWTinJRF[1] + pWTinJRF[2]).m2Calc() < M2MINJRF ) {
            infoPtr->errorMsg("Warning in StringFragmentation::fragmentTo"
                "Junction: Negative invariant masses in junction rest frame");
            MtoJRF.reset();
            MtoJRF.bstback(pSum);
            break;
        }
    } while (errInCM > 0.);
}

```

```

}
// Find new JRF from the set of weighted momenta.
Mstep = junctionRestFrame( pWTinJRF[0], pWTinJRF[1], pWTinJRF[2]);
// Fortran code will not take full step after the first few
// iterations. How implement this in terms of an M matrix??
MtoJRF.rotbst( Mstep );
} while (iter < 3 || (Mstep.deviation() > CONVJNREST && iter < NTRYJNREST) );

// If final deviation from 120 degrees is bigger than in CM then revert.
double errInJRF = pow2(costheta(pWTinJRF[0], pWTinJRF[1]) + 0.5)
+ pow2(costheta(pWTinJRF[0], pWTinJRF[2]) + 0.5)
+ pow2(costheta(pWTinJRF[1], pWTinJRF[2]) + 0.5);

if (errInJRF > errInCM + CONVJNREST){
  infoPtr->errorMsg("Warning in StringFragmentation::fragmentTo"
    "Junction: bad convergence junction rest frame");
  MtoJRF.reset();
  MtoJRF.bstback(pSum);
}

```

Details of the JRF finding section of *fragmentToJunction* within the NEW algorithm.

```

// Identify range of partons on the three legs.
// (Each leg begins with an iParton[i] = -(10 + 10*junctionNumber + leg),
// and partons then appear ordered from the junction outwards.)
int legBeg[3] = { 0, 0, 0};
int legEnd[3] = { 0, 0, 0};
int leg = -1;
// PS (4/10/2011) Protect against invalid systems
if (iParton[0] > 0) {
    infoPtr->errorMsg("Error in StringFragmentation::fragment"
        "ToJunction: iParton[0] not a valid junctionNumber");
    return false;
}
for (int i = 0; i < int(iParton.size()); ++i) {
    if (iParton[i] < 0) {
        if (leg == 2) {
            infoPtr->errorMsg("Error in StringFragmentation::fragment"
                "ToJunction: unprocessed multi-junction system");
            return false;
        }
        legBeg[++leg] = i + 1;
    }
    else legEnd[leg] = i;
}

// Make list of partons on each junction leg.
vector<vector<int>> iLeg;
for (int i = 0; i < 3; ++i) {
    vector<int> partonList;

    // Store event index of partons on given leg, from junction outwards.
    for (int l = legBeg[i]; l < legEnd[i] + 1; ++l) {

        // Split massive gluons into its mother gluons.
        if (event[iParton[l]].idAbs() == 21 && event[iParton[l]].m2() > 1e-4) {
            int motherParton = event[ iParton[l] ].mother1();

            // Make sure partons are listed in correct colour connected order.
            if (event[ motherParton ].statusAbs() == 73) {
                int iMum1 = event[ motherParton ].mother1();
                int iMum2 = event[ motherParton ].mother2();
                int lastParton = iParton[l+1];
                if ( int(partonList.size()) > 0 ) lastParton = partonList.back();
                bool colConnected = ( event[lastParton].acol() == event[iMum1].col()
                    || event[lastParton].col() == event[iMum1].acol() );
                if ( int(partonList.size()) > 0 ) {
                    iMum1 = iMum2;
                    iMum2 = event[ motherParton ].mother1();
                }
                if ( colConnected ) {
                    partonList.push_back( iMum2 );
                    partonList.push_back( iMum1 );
                } else {
                    partonList.push_back( iMum1 );
                    partonList.push_back( iMum2 );
                }
            }
            else partonList.push_back( iParton[l] );
        }
        iLeg.push_back(partonList);
    }
}

// Define variables for first partons on each leg.
int iNow[3] = {0, 0, 0};
Vec4 pLeg[3];
pLeg[0] = event[ iLeg[0][iNow[0]] ].p();

```

```

pLeg[1] = event[ iLeg[1][iNow[1]] ].p();
pLeg[2] = event[ iLeg[2][iNow[2]] ].p();
int idNow[3];
idNow[0] = event[ iLeg[0][iNow[0]] ].id();
idNow[1] = event[ iLeg[1][iNow[1]] ].id();
idNow[2] = event[ iLeg[2][iNow[2]] ].id();

for (int i = 0; i < 3; ++i) {
    if (idNow[i] == 21) pLeg[i] = 0.5 * pLeg[i];
}

// Find average JRF by stepping through partons on each junction leg.
int iMin, iSmall;
double pSmall;
double pSumJRF = 0.;
double weightSum = 0.;
bool findWeight = true;
bool endParton[3] = { false, false, false};
bool lastJRF = false;
vector<Vec4> listJRF;
vector<double> weightJRF;
vector< vector<Vec4> > pVec;
pVec.resize(3);

// Find JRF for partons until momentum exceeds PJNWEIGHTMAX.
do {
    // Find boost to JRF of given partons.
    Vec4 vJunNow = junctionRestFrame(pLeg[0], pLeg[1], pLeg[2],
        idNow[0], idNow[1], idNow[2]);
    listJRF.push_back( vJunNow );

    // Find smallest energy leg in given JRF. Check for at rest endpoints.
    iMin = 0;
    int iRest = -1;
    for (int i = 0; i < 3; ++i) {
        pLeg[i].bst(vJunNow);
        if ( pLeg[i].pAbs() < pLeg[iMin].pAbs() ) iMin = i;
        if ( pLeg[i].pAbs() < 1e-4 && iNow[i] == int(iLeg[i].size()) - 1 )
            iRest = i;
    }

    // Find smallest energy leg that is not an endpoint at rest.
    iSmall = iMin;
    if (iRest > -1) {
        vector<int> iList;
        for (int i = 0; i < 3; ++i)
            if ( i != iRest ) iList.push_back(i);
        if ( pLeg[iList[0]].pAbs() < pLeg[iList[1]].pAbs() )
            iSmall = iList[0];
        else iSmall = iList[1];
        pSmall = pLeg[iSmall].pAbs();
    } else pSmall = pLeg[iMin].pAbs();

    // Define pull vectors w.r.t. the lab frame.
    Vec4 pTemp;
    for (int i = 0; i < 3; ++i) {
        pTemp = pLeg[i];
        pTemp.bstback(vJunNow);
        pTemp.e( pTemp.pAbs() );
        pTemp /= pTemp.e();
        pVec[i].push_back(pTemp);
        if (iNow[i] == int(iLeg[i].size()) - 1 ) endParton[i] = true;
    }

    // Check how many legs we've reached the end parton for.

```

```

int endCounter = 0;
bool inRestFrame = false;
for (int i = 0; i < 3; ++i) {
    if (endParton[i] == true) {
        ++endCounter;
        if (pLeg[i].pAbs() < 1e-4) inRestFrame = true;
    }
}
// If at multiple endpoints and in the rest frame of a parton,
// stop the frame finding procedure here.
if (endCounter > 1 && inRestFrame) lastJRF = true;

// Update momenta of each leg and boost back to lab frame.
for (int i = 0; i < 3; ++i) {

    // Subtract used up momenta from legs not iMin or iSmall.
    if (i != iMin && i != iSmall) {

        double m2Now = event[ iLeg[i][iNow[i]] ].m2();
        double idFac = (idNow[i] == 21) ? 2. : 1.;
        pTemp = idFac * pLeg[i];
        pLeg[i] = pTemp - idFac * pSmall * pTemp / pTemp.pAbs();
        pLeg[i].e( sqrt( m2Now + pLeg[i].pAbs2( ) ) );
        pLeg[i] /= idFac;
        pLeg[i].bstback(vJunNow);

        // Protect against small energy massless partons.
        if ( pLeg[i].e() < EMINJRF ) {

            // If not at endpoint, step to next parton on leg.
            if (iNow[i] != int(iLeg[i].size()) - 1) {
                iNow[i] = iNow[i] + 1;
                idNow[i] = event[ iLeg[i][iNow[i]] ].id();
                pLeg[i] = event[ iLeg[i][iNow[i]] ].p();
                if (idNow[i] == 21) pLeg[i] = 0.5 * pLeg[i];

                // Else stop frame finding here.
            } else lastJRF = true;
        }
    }

    // Special treatment if the short leg is an endpoint parton.
    } else if ( iNow[i] == int(iLeg[i].size()) - 1 ) {

        // If massless endpoint, stop frame finding.
        double m2Now = event[ iLeg[i][iNow[i]] ].m2();
        // TODO: change pSmall to subtract till energy of mCon
        if ( m2Now < M2MINJRF ) lastJRF = true;

        // Else update momentum to be at rest.
        pLeg[i] = {0., 0., 0., 0.};
        pLeg[i].e( sqrt( m2Now ) );
        pLeg[i].bstback(vJunNow);

        // Else update pLeg of the short leg to the next parton on the leg.
    } else {
        iNow[i] = iNow[i] + 1;
        idNow[i] = event[ iLeg[i][iNow[i]] ].id();
        pLeg[i] = event[ iLeg[i][iNow[i]] ].p();
        if (idNow[i] == 21) pLeg[i] = 0.5 * pLeg[i];
    }
}

// Check for collinear partons with updated momenta. If so, stop here.
for (int i = 0; i < 3; ++i) {
    int j = i + 1;
    if (i == 2) j = 0;
}

```

```

// If two massive partons have same rest frame, cannot find further JRF.
double m2i = event[ iLeg[i][iNow[i]] ].m2();
double m2j = event[ iLeg[j][iNow[j]] ].m2();
if ( m2i > 0 && m2j > 0 ) {
    Vec4 pLegI = pLeg[i];
    pLegI.bstback(pLeg[j]);
    if ( pLegI.pAbs() < 1e-4 ) lastJRF = true;

// Check for partons that are massless and collinear.
// TODO: diquark treatment
} else if ( costheta( pLeg[i],pLeg[j] ) == 1 && m2i == 0 && m2j == 0 ) {
    lastJRF = true;
    infoPtr->errorMsg("Error in StringFragmentation::fragment"
        "ToJunction: cannot find JRF for collinear partons");
}
}

// Ensure weightings (in JRFs) do not exceed PJNWEIGHTMAX.
// If using JRF that is not 120 degrees topology, stop frame finding.
double pSumSaved = pSumJRF;
if (pSumJRF + pSmall > PJNWEIGHTMAX || lastJRF) {
    weightJRF.push_back( PJNWEIGHTMAX - pSumSaved );
    pSumJRF = PJNWEIGHTMAX;
} else {
    weightJRF.push_back( pSmall );
    pSumJRF += pSmall;
}

// Multiply weighting by gamma factor.
weightJRF.back() = weightJRF.back() / sqrt( 1 - vJunNow.pAbs2() );

// Define multiplicative factor in exponential weighting by pNormJunction.
if (pSumJRF > pNormJunction && findWeight) {
    weightSum += ( pNormJunction - pSumSaved )
        / sqrt( 1 - vJunNow.pAbs2() );
    findWeight = false;
} else if (pSumJRF < pNormJunction) weightSum += weightJRF.back();
} while (pSumJRF < PJNWEIGHTMAX);

// Find average junction velocity using exponential weighting.
double weightSoFar = 0.;
Vec4 vJunction( 0., 0., 0., 0.);
for (int i = 0; i < int(weightJRF.size()); ++i) {
    vJunction += listJRF[i] * ( exp(- weightSoFar / weightSum)
        - exp(- (weightJRF[i] + weightSoFar) / weightSum ) );
    weightSoFar += weightJRF[i];
}
vJunction /= (1 - exp(- weightSoFar / weightSum));

MtoJRF.reset();
MtoJRF.bst(vJunction);

```