The Lund Symmetric Fragmentation Function Bug Fix

The Pythia Collaboration

Original date: 16 May 2025
Last changed: 16 May 2025

The Lund symmetric fragmentation function (LSFF) is introduced to describe the sharing of momentum between hadrons in the fragmentation of a partonic system. For the issue at hand it is sufficient to consider a simple quark-antiquark pair moving out along the $\pm z$ axis, stretching out a string between them. The fragmentation can be described as a step-wise splitting off of a hadron from either end. The process should give the same result, on the average, whether hadrons are split off from the quark end towards the antiquark one, or the other way around. This requirement has a simple solution, the LSFF [1],

$$f(z) \propto \frac{1}{z}\,(1-z)^a\,\exp(-bm_\perp^2/z)\;, \tag{1}$$

where a and b are free parameters, and $m_\perp^2 = m^2 + p_\perp^2$ is the squared transverse mass of the produced hadron. The z variable is the lightcone fraction taken by the new hadron out of what remains from previous hadron production, where the lightcone amount is $E + p_z$ if moving from the positive end and $E - p_z$ if from the negative one. This nicely gives a probability distribution of the invariant time τ of string breaks

$$P(\Gamma) \propto \Gamma^a\,\exp(-b\Gamma)\;, \tag{2}$$

where $\Gamma = (\kappa\tau)^2$, with κ the string tension.

But this is not the most general form. It is possible to imagine that different flavours i have different characteristic production times, regulated by different a_i [2], *i.e.*

$$P_i(\Gamma) \propto \Gamma^{a_i}\,\exp(-b\Gamma)\;. \tag{3}$$

This leads to an extended shape of the LSFF

$$f(z) \propto \frac{1}{z}\,z^{a_i}\left(\frac{1-z}{z}\right)^{a_j}\,\exp(-bm_\perp^2/z) = \frac{(1-z)^{a_j}}{z^{1-a_i+a_j}}\,\exp(-bm_\perp^2/z)\;, \tag{4}$$

for a step from the old flavour i to the new flavour j. The main application has been to allow a larger a for diquarks, on the assumption that the production of two quarks could take a longer time than that of a single quark. The code also allows for a different a for strange quarks, but this possibility has seen no (little?) use, and is left aside.

The bug now found (by Tony Menzo; thanks a lot!) is a typo introduced already in the transition from Pythia 6 to Pythia 8, whereby the $(1-z)^{a_j}$ factor erroneously was encoded as $(1-z)^{a_i}$. This does then not affect the production of mesons, but only

that of baryons. An example is given in fig. 1, with $a_{\rm q} = a = 0.5$ for quarks ("light"), $a_{\rm qq} = a + a_{\rm extra} = 0.5 + 0.5 = 1.0$ for diquarks ("qq/s"), and $bm_\perp^2 = 0.8$. In the step from a quark to a diquark, producing a baryon, the bug gives a too large average z value, while a subsequent step from a matching antidiquark to a new antiquark gives too small a $\langle z \rangle$. The bug has tended to shift the distributions closer to the normal ones for light quarks, i.e. the attempts to use an $a_{\rm extra} > 0$ to obtain shifts for baryons has partly been nullified by the bug. That the shifts have not completely flipped sign is because the $f(z)$ spectra are modified also by the $z^{a_i}/z^{a_j} = z^{\pm a_{\rm extra}}$ factor.

The bug most directly affects baryon–antibaryon correlations, and also single baryon spectra. The two opposite-direction errors tend to cancel each other in a bigger picture, however, leaving multiplicity distributions and event shapes largely undisturbed. This is illustrated in the next section.

1 Comparisons

In this section we show some comparisons, how results with the default Monash tune [3] are affected by the bug fix, when no other changes are made to the code. Specifically, we keep $a =$ `StringZ:aLund` $= 0.68$, $b =$ `StringZ:bLund` $= 0.98$ GeV^{-2}, and $a_{\rm extra} =$ `StringZ:aExtraDiquark` $= 0.97$. Based on the argument above, we begin by the least critical distributions, and proceed towards more critical ones.

1.1 Generic event properties

Considering distributions where baryons are not identified as such, exemplified in fig. 2, any differences are marginal, as expected. Many other distributions have been studied, also LHC ones, so far without any major discrepancies.

1.2 Single baryon spectra

The scaled momentum spectra in fig. 3 show some improvement at the largest x_p values, but the bulk of the spectra are not all that much affected. This is as expected, since the shifts go in opposite directions and largely cancel. The one exception is near the string endpoints, since these are always quarks and never diquarks in $\rm e^+e^-$, so $a_{\rm extra}$ always wants to soften the endpoint baryons. In Monash the $a_{\rm extra}$ default value largely is driven by such considerations. The situation is different in the forward direction of pp collisions, where diquark fragmentation becomes important, but then also other aspects enter.

1.3 Baryon–antibaryon correlations

It is in baryon–antibaryon correlations that we expect the largest effects of the bug. As can be seen in fig. 4, the introduction of the bug fix indeed gives a significant shift, towards a worse description of data. This is unfortunate but not fatal, since also other parameters come into play here, such as the rate of the popcorn baryon production mechanism. Also, since now the shifts of $\langle z \rangle$ away from the default quark case become larger, a new

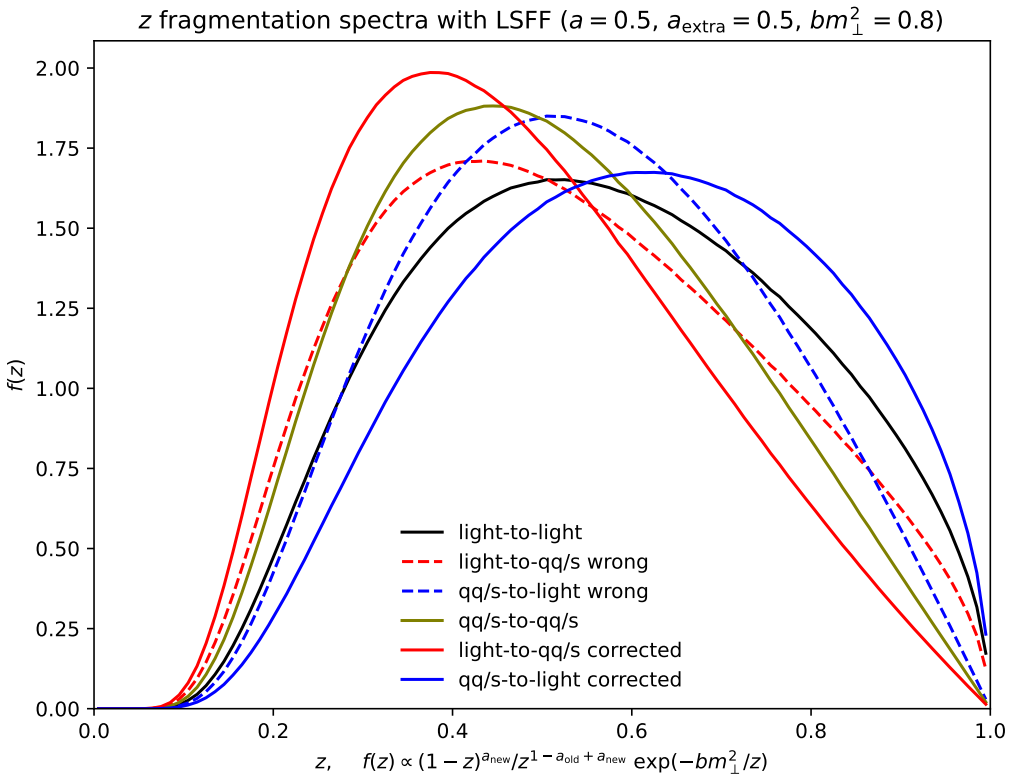

Figure 1: $f(z)$ distributions for a simple test case, comparing the old wrong behaviour and the new corrected one.

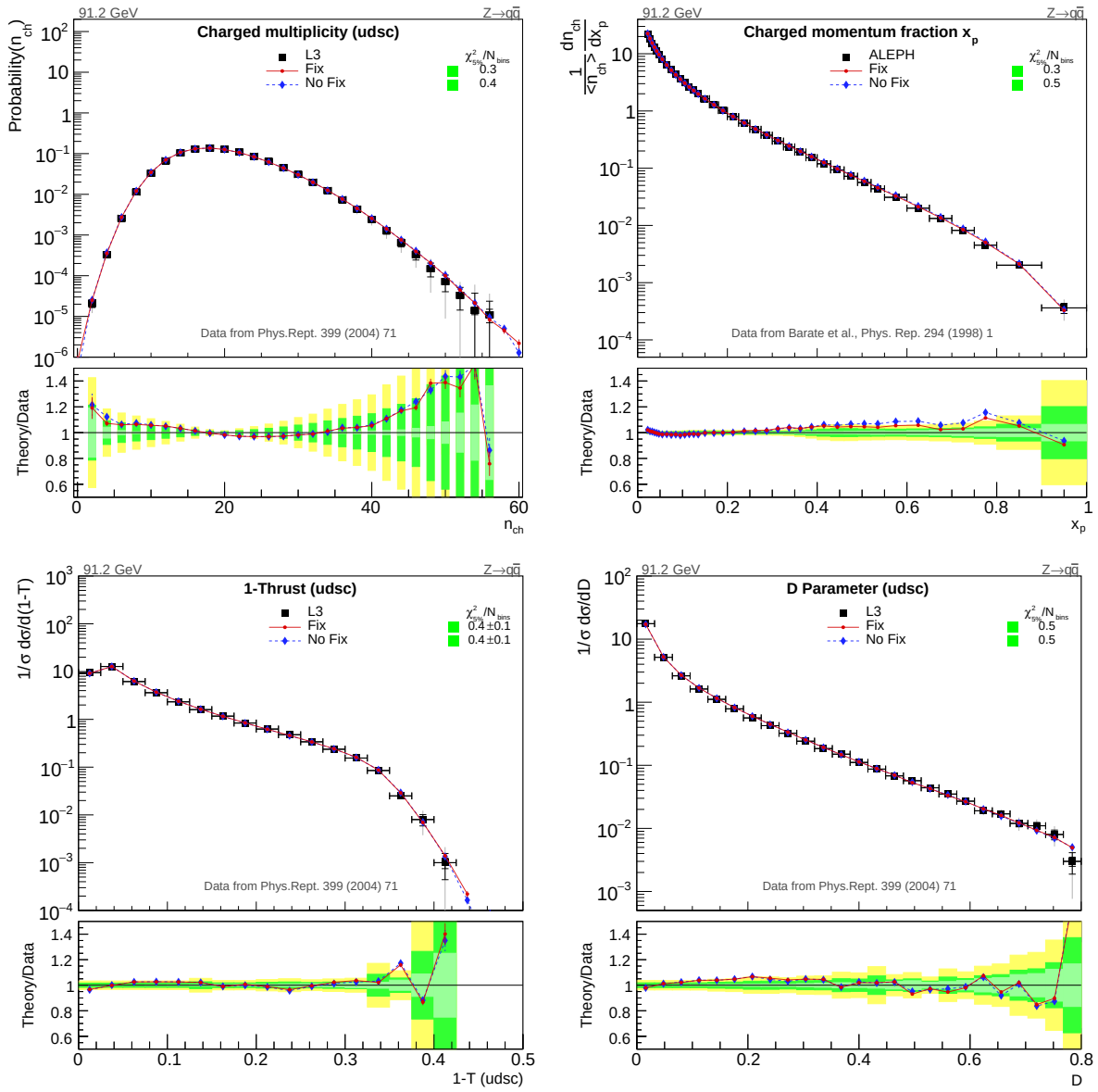

Figure 2: Some generic LEP event properties, compared with L3 [4] or ALEPH [7] data: charged multiplicity, charged-particle momentum fraction, Thrust and the D parameter.

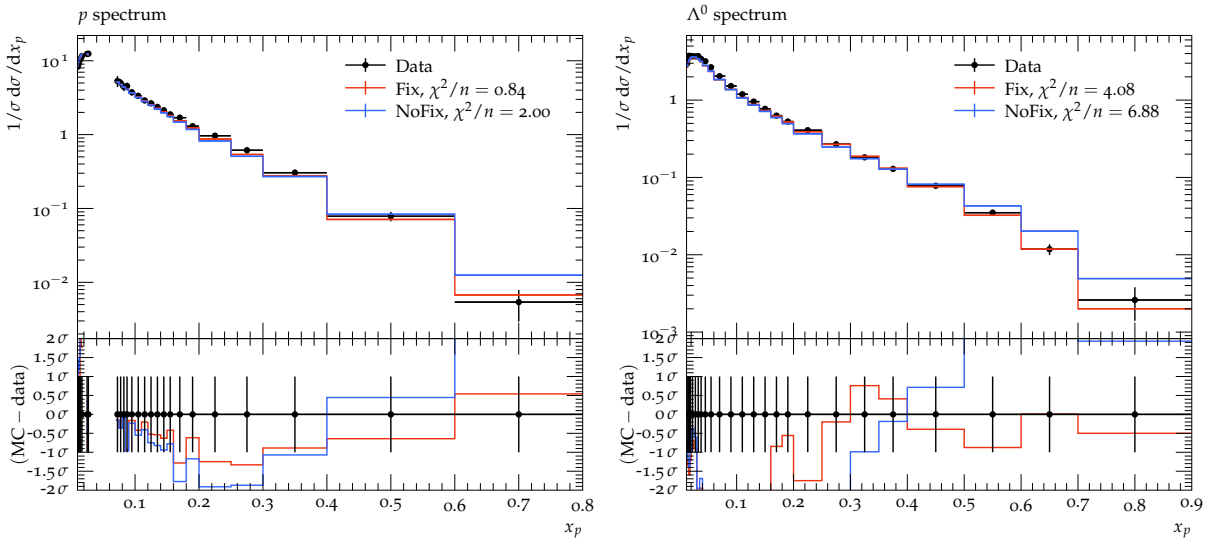

Figure 3: Proton and Λ scaled momentum spectra at LEP, compared with OPAL data [6].

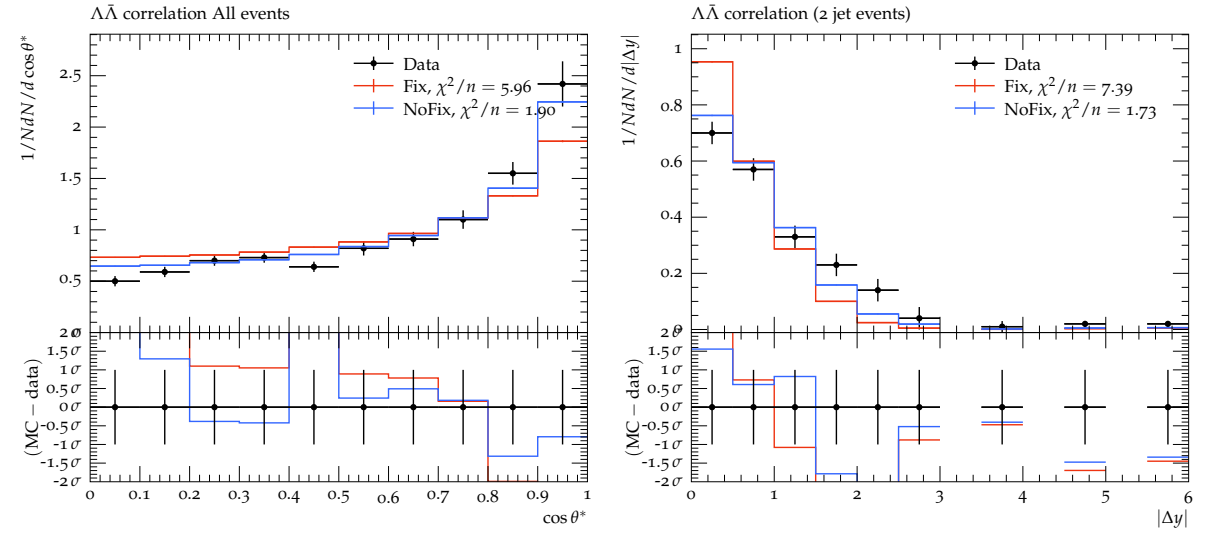

Figure 4: $\Lambda - \overline{\Lambda}$ correlations in angle and in rapidity at LEP, with and without the bug fix, compared with ALEPH data [7].

tune likely could achieve the same effect with a smaller $a_{\rm extra}$. One may note that these distributions were not used in the Monash tune, however.

2 Summary and Outlook

A serious bug has been found, affecting all previous PYTHIA 8 versions. Fortunately, overall effects are not as significant as one might fear at first glance, unless one is zooming in on baryon production. Nevertheless, existing tunes, such as Monash, no longer are optimal. Therefore new tunes should be produced, in a first instance ones targeting e^+e^- data. Such an effort has already started and results should be presented in a near future. For hadron collider tunes also newer data should be included, and is likely to be a larger driving force than the bug fix. Obviously this is a taller order, and will take longer.

References

[1] B. Andersson, G. Gustafson, G. Ingelman and T. Sjostrand, Phys. Rept. **97** (1983), 31-145 doi:10.1016/0370-1573(83)90080-7

[2] B. Andersson, G. Gustafson and B. Soderberg, Z. Phys. C **20** (1983), 317 doi:10.1007/BF01407824

[3] P. Skands, S. Carrazza and J. Rojo, Eur. Phys. J. C **74** (2014) no.8, 3024 doi:10.1140/epjc/s10052-014-3024-y [arXiv:1404.5630 [hep-ph]].

[4] P. Achard *et al.* [L3], Phys. Rept. **399** (2004), 71-174 doi:10.1016/j.physrep.2004.07.002 [arXiv:hep-ex/0406049 [hep-ex]].

[5] R. Barate *et al.* [ALEPH], Phys. Rept. **294** (1998), 1-165 doi:10.1016/S0370-1573(97)00045-8

[6] G. Abbiendi *et al.* [OPAL], Eur. Phys. J. C **13** (2000), 185-195 doi:10.1007/s100520050685 [arXiv:hep-ex/9808031 [hep-ex]].

[7] R. Barate *et al.* [ALEPH], Phys. Rept. **294** (1998), 1-165 doi:10.1016/S0370-1573(97)00045-8