Reducing the scaling complexity for high-multiplcity event generation with AmpliCol

Rikkert Frederix[1]*, Timea Vitos[2,3]†

[1] *Department of Physics, Lund University, Sölvegatan 14A, SE-223 62, Lund, Sweden*
[2] *Department of Physics and Astronomy, Uppsala University, Box 516, 751 20, Uppsala, Sweden*
[3] *Institute for Theoretical Physics, ELTE Eötvös Loránd University, Pázmány Péter sétány 1/A, H-1117 Budapest, Hungary*

Abstract

The scaling of event generation with particle number is a challenge in particle physics. In this work, we target the efficient generation of realistic LHC processes with increasing number of jets in the final state. The main idea relies on a previously developed two-step approach, in which unweighted events are generated at leading-colour accuracy, and then in a second step reweighted to full-colour accuracy, thus remaining efficient in the integration step, but capturing the full-colour accuracy in the hard scattering process. We compare the computation time for four benchmark LHC processes: multi-jet, $t\bar{t}$ plus jets, ZZ plus jets and Drell-Yan plus jets. The results show that the combined timing of generating a fixed number of unweighted events at full-colour accuracy is moderate and scales exponentially with the particle multiplicities we consider, vastly outsourcing the factorial growth in conventional approaches.

*E-mail: `rikkert.frederix@fysik.lu.se`
†E-mail: `timea.vitos@physics.uu.se`

Contents

1 Introduction

The current procedure for precise predictions and measurements of particle physics collisions relies heavily on computational simulations. Although the general-purpose collision simulators have been around for 20-30 years [1–3], there have been numerous developments the past decades targeting various aspects of the computations, both in the width of the targeted processes and in their computational efficiencies. One still standing problem for event generators is the multiplicity handling: the computations are too slow or not even feasible when the final state multiplicity (often in terms of jets) in the collisions in the hard scattering reaches a (mostly algorithm-dependent) limit. The alternative has long been the usage of parton showers for the generation of multiple jets, with the obvious limitation of the range of validity to the soft and collision regions.

The main computational challenge for these multi-jet events is due to the complex structure of quantum chromodynamics (QCD). The integrand in the computation of the cross section grows factorially with the external particle number, resulting in a highly inefficient integration due to the increasing integrand evaluation time. In conventional tools, this results in a break-down of the execution feasibility of the computation at some multiplicity numbers. Many works have contributed to improvements in this area, including the improvement of the phase-space integration [4–15], sampling of colour and helicity configurations [13, 16–22], and recursion relations instead of conventional diagram-based computation of the matrix-elements [16, 23]. Another approach is the reduction of the integrand by introducing a truncation in the colour expansion, either at leading-colour (LC), leading to a linear scaling in the number of terms with the external multiplicity, or at the next-to-leading colour (NLC) order yielding a polynomial scaling [24, 25]. There have been works targeting the efficient event generation of high-multiplicity events, including COMIX [13] and Pepper [26].

One possible solution to the problem of increasing complexity in event generation was presented in our previous work Ref. [27]. We suggest there a two-step approach for generating events at leading-order (tree-level) in the Standard Model coupling expansion. The first step constitutes a conventional phase-space integration, in which the amplitudes are computed with off-shell recursion relations [16, 23], and the precision in the colour expansion is truncated in the first order, the LC approximation. This truncation results in a much more efficient evaluation of the matrix-elements, as the factorial complexity is reduced to a exponential complexity. In this manner, unweighted LC events are generated, covering all of the targeted phase-space, following a LC integrand in the importance sampling. In the second step, these unweighted events are passed through a reweighting module, upon which the full-colour (FC) matrix-elements are evaluated for each event, and with a secondary unweighting, the new set of FC events are obtained.

In this work, we present results for hadronic collisions, by properly dealing with partonic subprocesses in an efficient manner in the standalone, Fortran-based computer code which we dub AmpliCol. We present results for four benchmark processes and compare the scaling behaviours for increasing jet multiplicity. The paper is structured in the following way: in Section 2 we present the methodology by first briefly summarising the two-step event generation and discussing the phase-space integration. In Section 3 we present our findings, and finally summarise and present an outlook in Section 4.

2 Method

The methodology of the present work follows closely that presented in Ref. [27]. In the following we present the brief summary of the method and highlight the main new features. We refer the interested reader to the mentioned reference for further details of the method.

2.1 Overview of the two-step event generation

In the phase-space integral of the cross section, the main bottleneck is the highly complex and peaked matrix-element in the integrand. Suppressing the parton density functions, flux factor and additional factors coming from colour and helicity averages, the integral can be rewritten as a double sum over dual amplitudes $\mathcal{A}_i$ and corresponding colour factors C_{ij} [16, 28], as

$$\sigma_{\mathrm{FC}} \sim \int |\mathcal{M}|^2 \mathrm{d}\Phi = \int \sum_{i,j} \mathcal{A}_i C_{ij} \mathcal{A}_j^* \mathrm{d}\Phi, \tag{1}$$

where the i, j indices label colour orderings of the external QCD particles. The only terms contributing at leading-colour in this colour sum are those with identical colour order, $i = j$ and so the LC-truncated cross section becomes

$$\sigma_{\mathrm{LC}} \sim \int \sum_{i} \mathcal{A}_i C_{ii} \mathcal{A}_i^* \mathrm{d}\Phi = \sum_i C_{ii} \int |\mathcal{A}_i|^2 \mathrm{d}\Phi. \tag{2}$$

This rewriting of the cross section is used in the proposed algorithm for the integration step: the integrand is approximated at the LC accuracy, as the square of the dual amplitudes, using multi-channeling for each of the orderings. This alleviates a fast integrand evaluation and efficient phase-space integration. The obtained events which pass the generation cuts are unweighted, leading to the sample of LC unweighted events.

In the second step, these LC events are passed through a reweighting algorithm, in which each event is multiplied with the reweight factor

$$r^{\mathrm{LC}\to\mathrm{FC}} = \frac{|\mathcal{M}|^2}{\sum_i C_{ii} |\mathcal{A}_i|^2}, \tag{3}$$

which is a correction factor for obtaining the FC weight. This correction factor is evaluated using the specific kinematics and helicity of each event, in this way obtaining a FC-accurate (weighted) event sample. Finally, a secondary unweighting can be performed, resulting in the FC event generation which finalizes the two-step algorithm for the efficient event generation.

2.2 Phase-space integration

For the standard integration of a generic partonic process

$$ab \to 12\dots n \tag{4}$$

we use the phase-space parametrisation introduced in Ref. [29] and used in Ref. [27]. This parametrisation is based on a $2 \to 3$ building block of consecutive momenta generations, following the colour ordering of the particles and maps all relavant variables of the maximally-helicity-violating (MHV) amplitudes [30] to integration variables. The phase space integration is split to generate two sets of particles: those between particle a and b, and those between b and a. The particle kinematics in the two sets are then generated in a way that follows the peak structure of the MHV amplitudes, and therefore also for helicity summed amplitudes, and hence results in an efficient phase-space sampling.

For the full LHC process including hadrons and jets, all possible subprocesses (treating the b-quark in the requested flavour-scheme) contributing to the requested collision are grouped in common phase-space order blocks. The given colour order of each block is then passed the phase-space generation described in the above, generating the kinematics in the specified order for the particles. The kinematics is then assigned for the different subprocesses in the block. With this method, one reduces overflow of phase-space samplings, improving further the efficiency of the algorithm.

2.3 Matrix-elements and subprocesses

The dual amplitudes are computed using the off-shell recursion relations, including all particles in the Standard Model. The main contributions to the partonic processes are, whenever present, the all-gluon processes. The subprocesses with quark lines are subdominant, and decrease in dominancy as the number of quark lines grow. Hence, we include the two-quark-line processes as standard to all the hadronic processes, however, the three-quark-line subprocesses are not included in the standard computation. We do, however, an analysis on their impact on the computation time, by including them on the level of the integration (generation of LC events).

Mention color singlet multi-channeling?

Mention the decomposition of SF processes?

Benchmarked against known results.

3 Results

We use the above method for the event generation of four benchmark LHC processes at center-of-mass energy of $\sqrt{s} = 14$ TeV:

$$\begin{aligned} &pp \to nj, \\ &pp \to t\bar{t} + (n-2)j, \\ &pp \to ZZ + (n-2)j, \\ &pp \to e^+e^- + (n-1)j, \end{aligned} \tag{5}$$

with the multiplicity number n varied in the range $n \in [2, 6]$. In all processes we use a 5-flavour scheme and set the following cuts on the jets, whenever applicable:

$$p_T(j) > 30 \text{ GeV} \quad , \quad \eta(j) < 6.0 \quad , \quad \Delta R(j_1, j_2) > 0.4. \tag{6}$$

In the Drell-Yan process, a cut on the leptonic invariant mass of $m(e^+e^-) > 50$ GeV is used in order to avoid the photon-mediated singularity, and no further cuts on the leptons are placed. We set the renoralisation and factorisation scales equal to the Z boson mass, and use the NNPDF23nlo set [31] for the initial state parton distributions. We perform all the computations on a single core of an Intel i7-8700K CPU running at 3.70GHz and extract timings from the runs, focussing on how the runtime scales with the multiplicity number n.

The total time t_{tot} for the generation of unweighted events at FC accuracy in the two-step approach consists of

$$t_{\text{tot}} = t_{\text{int}} + t_{\text{rwgt}} \quad . \tag{7}$$

Here, the integration time t_{int} is for the generation of unweighted LC accurate events. This includes the setup of the integration grids, finding the upper bound of the event weights, and the unweighting of the generated events. At most a fraction of 0.1% of the cross section, the so-called overweight fraction see e.g. [**?**], is truncated to improve unweighting efficiency, but typically this number is between 0.05 to 0.07% for the results presented in this work. The time for reweighting t_{rwgt} (plus secondary unweighting) is required to adjust the weights such that they capture the FC corrections. One must take into consideration the loss of events in the secondary unweighting step and generate the LC event number accordingly, such that the final number of FC events is the required event sample. We consider this fractional loss in the timings of the total time. In the following we compare these various times for the different processes.

In Fig. 1 we show the main results of this work: the time to generate $N = 10^5$ unweighted events at FC accuracy on a single CPU core. It is shown for the four sample benchmark processes for varying jet multiplicity. The plot includes the total time t_{tot} required to generate these events (blobs with error bars) and also separately the reweighting time t_{rwgt} for each of the processes (triangles), showing clearly that this step is negligible in the fraction of the total time for all processes considered. In addition, a fit of an exponential curve is made to the total time with respect to the multiplicity n. The fit indicates an exponential growth for all processes with a base of around 5 (with a slight variance between process types, each shown in the figure).

Move the following to an appendix?

In our secondary unweighting procedure, i.e. in the reweighting of the LC events to FC, we follow the usual procedure of first finding the maximum weight among a sample of events w_{max} and then sampling from the pool of events with the acceptance probability

$$u_i^{\text{eff}} = \frac{w_i}{w_{\text{max}}}, \tag{8}$$

where w_i is given by the reweight factor, $r^{\text{LC}\to\text{FC}}$. The fraction of events retained in this procedure is the secondary unweighted efficiency. In Fig. 2 (right left) we plot the secondary unweighting efficiency for the processes considered. This efficiency remains above approximately 60% for all processes and multiplicities, indicating on average a sample size of roughly 1.7 larger at LC accuracy to generate the required number of FC events.

Even though the secondary unweighting efficiency is large, one may consider keeping the weighted events instead. A way of assessing the statistical power of a weighted sample of size N with weights w_i is by evaluating the Kish effective sample size (ESS), defined by

$$f_{\text{ESS}} = \frac{1}{N}?\frac{\left(\sum_{i=1}^{N} w_i\right)^2}{\sum_{i=1}^{N} w_i^2}. \tag{9}$$

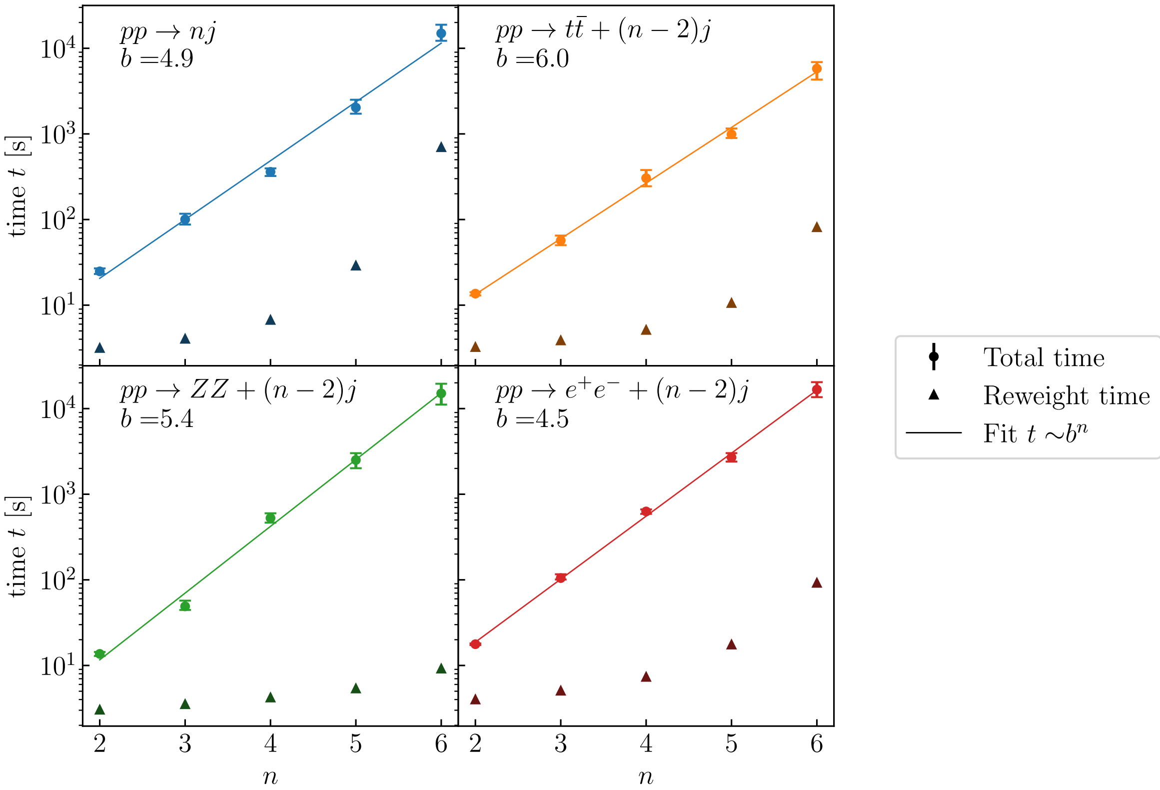

Figure 1: Computational time (in seconds) for the four benchmark processes. Shown is the total computation time (blob with error bar) for 10^5 unweighted events at FC accuracy (including the generation time of the LC events, the reweighting time, and accounting for the secondary unweighting loss), and separately the reweighting time (triangle). An exponential curve is fitted to the total timings and portrayed as a line in the plots, with the base b of the exponential indicated in each of the four processes.

This measure evaluates the spread of the weights, reaching a value of 1 for a sample of unweighted events (equal weights). A few outliers from an otherwise almost equal-weight event sample do not impact this effective size greatly, while one outlier will greatly decrease the unweighting efficiency defined in eq. 8. We show the ESS values for the four benchmark processes for varying multiplicity n in the left(right)-hand plot of Fig. 2. Even for the highest multiplicities at $n = 6$, all of these processes remain at an ESS value of beyond 98%, with the multi-jet processes scaling the best with the multiplicity increase. This suggests that keeping the weighted events and thereby reducing the effective sample size by less than 2% significantly outperforms the unweighting the events, except in cases where post-processing (parton shower, hadronisation, detector simulation, etc.) is particularly time consuming.

4 Discussion and outlook

In this work we presented the program AmpliCol, which performs the two-step event generation of high-multiplicity processes, with an exponential increase in computation time (at least up to moder-

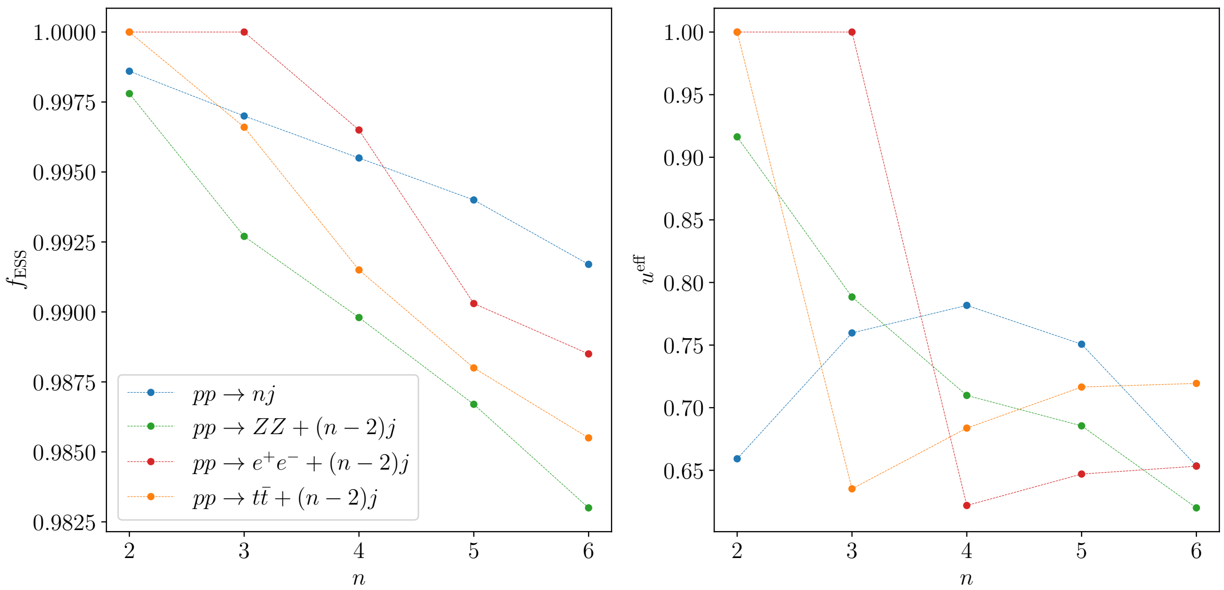

Figure 2: Effective sample size f_{ESS} for increasing jet multiplicity (left) and the unweighting efficiency u^{eff} (right) for the four benchmark processes considered.Can we put unweighting efficiency on the left, and ESS on the right?)

ately high multiplicities) rather than the conventional factorial growth with conventional event generators. We illustrated the computation time for four benchmark processes with various multiplicities. The AmpliCol program will in the near future be interfaced with the MadGraph5_aMC@NLO user-friendly matrix-element generator [14], improving the automated computation of multi-jet processes, reaching multiplicities which are currently out of reach within that framework.

The current version of the program works at tree-level in the perutrbative expansion, and we leave the extension of the code for NLO computations for future work.

Acknowledgements

The work of R.F. is supported by the Swedish Research Council under project number 202004423. The work of T.V. is supported by the Swedish Research Council under project number VR:2023-00221. **Update if needed**

References

[1] T. Sjöstrand, S. Ask, J.R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C.O. Rasmussen, P.Z. Skands, Comput. Phys. Commun. **191**, 159 (2015). DOI 10.1016/j.cpc.2015.01.024

[2] M. Bahr, et al., Eur. Phys. J. C **58**, 639 (2008). DOI 10.1140/epjc/s10052-008-0798-9

[3] T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert, J. Winter, JHEP **02**, 007 (2009). DOI 10.1088/1126-6708/2009/02/007

[4] G.P. Lepage, J. Comput. Phys. **27**, 192 (1978). DOI 10.1016/0021-9991(78)90004-9

[5] G.P. Lepage, J. Comput. Phys. **439**, 110386 (2021). DOI 10.1016/j.jcp.2021.110386

[6] A. van Hameren, Acta Phys. Polon. B **40**, 259 (2009)

[7] R. Kleiss, R. Pittau, Comput. Phys. Commun. **83**, 141 (1994). DOI 10.1016/0010-4655(94)90043-4

[8] T. Ohl, Comput. Phys. Commun. **120**, 13 (1999). DOI 10.1016/S0010-4655(99)00209-X

[9] F. Maltoni, T. Stelzer, JHEP **02**, 027 (2003). DOI 10.1088/1126-6708/2003/02/027

[10] C.G. Papadopoulos, Comput. Phys. Commun. **137**, 247 (2001). DOI 10.1016/S0010-4655(01)00163-1

[11] F. Krauss, R. Kuhn, G. Soff, JHEP **02**, 044 (2002). DOI 10.1088/1126-6708/2002/02/044

[12] W. Kilian, T. Ohl, J. Reuter, Eur. Phys. J. C **71**, 1742 (2011). DOI 10.1140/epjc/s10052-011-1742-y

[13] T. Gleisberg, S. Hoeche, JHEP **12**, 039 (2008). DOI 10.1088/1126-6708/2008/12/039

[14] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.S. Shao, T. Stelzer, P. Torrielli, M. Zaro, JHEP **07**, 079 (2014). DOI 10.1007/JHEP07(2014)079

[15] E. Bothmann, et al., SciPost Phys. **7**(3), 034 (2019). DOI 10.21468/SciPostPhys.7.3.034

[16] F.A. Berends, W.T. Giele, Nucl. Phys. B **306**, 759 (1988). DOI 10.1016/0550-3213(88)90442-7

[17] F. Caravaglios, M. Moretti, Phys. Lett. B **358**, 332 (1995). DOI 10.1016/0370-2693(95)00971-M

[18] F. Caravaglios, M.L. Mangano, M. Moretti, R. Pittau, Nucl. Phys. B **539**, 215 (1999). DOI 10.1016/S0550-3213(98)00739-1

[19] P. Draggiotis, R.H.P. Kleiss, C.G. Papadopoulos, Phys. Lett. B **439**, 157 (1998). DOI 10.1016/S0370-2693(98)01015-6

[20] M.L. Mangano, M. Moretti, F. Piccinini, R. Pittau, A.D. Polosa, JHEP **07**, 001 (2003). DOI 10.1088/1126-6708/2003/07/001

[21] C. Duhr, S. Hoeche, F. Maltoni, JHEP **08**, 062 (2006). DOI 10.1088/1126-6708/2006/08/062

[22] O. Mattelaer, K. Ostrolenk, Eur. Phys. J. C **81**(5), 435 (2021). DOI 10.1140/epjc/s10052-021-09204-7

[23] R. Britto, F. Cachazo, B. Feng, Nucl. Phys. B **715**, 499 (2005). DOI 10.1016/j.nuclphysb.2005.02.030

[24] R. Frederix, T. Vitos, JHEP **12**, 157 (2021). DOI 10.1007/JHEP12(2021)157

[25] S. Badger, B. Biedermann, P. Uwer, V. Yundin, Comput. Phys. Commun. **184**, 1981 (2013). DOI 10.1016/j.cpc.2013.03.018

[26] E. Bothmann, T. Childers, W. Giele, S. Höche, J. Isaacson, M. Knobbe, SciPost Phys. **17**(3), 081 (2024). DOI 10.21468/SciPostPhys.17.3.081

[27] R. Frederix, T. Vitos, JHEP **12**, 201 (2024). DOI 10.1007/JHEP12(2024)201

[28] M.L. Mangano, S.J. Parke, Phys. Rept. **200**, 301 (1991). DOI 10.1016/0370-1573(91)90091-Y

[29] E. Byckling, K. Kajantie, Phys. Rev. **187**, 2008 (1969). DOI 10.1103/PhysRev.187.2008

[30] S.J. Parke, T.R. Taylor, Phys. Rev. Lett. **56**, 2459 (1986). DOI 10.1103/PhysRevLett.56.2459

[31] R.D. Ball, et al., Nucl. Phys. B **867**, 244 (2013). DOI 10.1016/j.nuclphysb.2012.10.003

[32] K. Danziger, T. Janßen, S. Schumann, F. Siegert, SciPost Phys. **12**, 164 (2022). DOI 10.21468/SciPostPhys.12.5.164

[33] C. Gao, S. Höche, J. Isaacson, C. Krause, H. Schulz, Phys. Rev. D **101**(7), 076002 (2020). DOI 10.1103/PhysRevD.101.076002