Nov 2020

CASCADE3:
A Monte Carlo event generator based on TMDs

H. Jung[1], S. Baranov[2], A. Bermudez Martinez[1], F. Hautmann[3,4], A. Lelek[3], A. Lipatov[5], M. Malyshev[5], M. Mendizabal[1], S. Taheri Monfared[1], A.M. van Kampen[3], Q. Wang[1,6]

[1]DESY, Hamburg, FRG
[2]Lebedev Physics Institute, Russia
[3]Elementary Particle Physics, University of Antwerp, B 2020 Antwerp
[4]RAL, Chilton OX11 0QX and University of Oxford, OX1 3NP
[5]SINP, Moscow State University, Russia
[6]School of Physics, Peking University

Abstract

The CASCADE3 Monte Carlo event generator based on Transverse Momentum Dependent (TMD) parton densities is described. Hard processes can be generated internally within k_t-factorization (as in previous versions) or read in via LHE event files, generated within collinear factorization via the packages aMC@NLO, or generated with off-shell kinematics. A TMD initial state parton shower is generated which follows the distributions from the TMDs.

PROGRAM SUMMARY

Title of Program: CASCADE3 3.0.2-beta11

Computer for which the program is designed and others on which it is operable: any with standard Fortran 77 (gfortran)

Programming Language used: FORTRAN 77

High-speed storage required: No

Separate documentation available: No

Keywords: QCD, TMD parton distributions.

Method of solution: Since measurements involve complex cuts and multi-particle final states, the ideal tool for any theoretical description of the data is a Monte Carlo event generator which generates initial state parton showers according to Transverse Momentum Dependent (TMD) parton densities, in a backward evolution. The evolution follows the DGLAP evolution equation exactly as used for the determination of the TMD.

Restrictions on the complexity of the problem: Any LHE file (with on-shell or off-shell) initial state partons can be processed.

Other Program used: PYTHIA (*version* > 6.4) for hadronization, TMDLIB as a library for TMD parton densities BASES/SPRING 5.1 for integration (supplied with the program package).

Download of the program: `http://www.desy.de/~jung/cascade`

Unusual features of the program: None

1 Introduction

The simulation of processes for high energy hadron colliders has been improved significantly in the past years by automation of next-to-leading order (NLO) calculations and the matching of the hard processes to parton shower Monte Carlo event generators which also include a simulation of hadronization. Among those are the MADGRAPH5_aMC@NLO [1] generator based on the MC@NLO [2–5] or the POWHEG [6,7] method for the calculation of the hard process. The results these packages are then combined with either the HERWIG [8] or PYTHIA [9] packages for parton showering and hadronization. Different jet multiplicities can be combined at the matrix element level and then merged with special procedures, like the MLM merging [10] for LO processes, the FxFx [11] or MiNLO method [12] for merging at NLO. While the approaches of matching and merging matrix element calculations and parton showers are very successful, two ingredients important for high energy collisions are not (fully) treated: while the matrix elements are calculated with collinear dynamics, the inclusion of initial state parton showers results in a net transverse momentum of the hard process; the special treatment of high energy effects (small x) is not included.

The CASCADE3 Monte Carlo event generator, developed originally for small x processes based on the CCFM [13–16] evolution equation, has been extended to include the full kinematic range (not only small x) by applying the newly developed Parton Branching (PB) Transverse Momentum Dependent (TMD) parton densities [17,18]. The initial state radiation is fully described and determined by the TMD density, similarly to the case of the CCFM gluon density, but now available for all flavor species, including quarks and gluons at small and large x and scale μ.

With the developments in determination of transverse momentum dependent (TMD) parton densities [17,18], it is natural to develop a scheme, where the initial parton shower follows exactly the TMD parton density and where either collinear (on-shell) of k_t-dependent (off-shell) hard process calculations can be combined. In order to be flexible and to use the latest developments in automated matrix element calculations of hard process at higher order in the strong coupling α_s, events available in the Les Houches Event (LHE) file format [19], which contains all the information of the hard process including the color structure, can be read into CASCADE3.

In this report we describe the new developments in CASCADE3 for a full PBTMD parton shower and the matching of TMD parton densities to collinear hard process calculations. We also mention features of the small x mode of CASCADE3.

2 The hard process

The cross section of two hadrons A and B can be written as a convolution of the partonic cross section of partons a and b: $a + b \to X$ and the density of partons a (b) inside the hadrons A (B):

$$\sigma(A+B \to Y) = \int dx_a \int dk_{t\,a}^2 \int dx_b \int dk_{t\,b}^2 \, \mathcal{A}(x_a, k_{t\,a}, \mu)\, \mathcal{A}(x_b, k_{t\,b}, \mu)\, \sigma(a+b \to X)\,, \quad (1)$$

where $x_a(x_b)$ are the longitudinal momentum fractions of partons $a(b)$ and $k_{t\,a}(k_{t\,b})$ are their transverse momenta, and $\sigma(a+b \to X)$ is the partonic cross section, and μ is the factorization scale of the process. The final state Y contains the partonic final state X and the recoils from the parton evolution and hadron remnants.

In the following we discuss separately on-shell as well as off-shell partonic processes.

2.1 On-shell processes

The hard processes in collinear factorization (with on-shell initial partons, without transverse momenta) can be calculated by standard automated methods like MADGRAPH5_aMC@NLO [1] at NLO accuracy. The matrix element processes are calculated with collinear parton densities (PDF), as provided by LHAPDF [20]. However, when the hard process is to be combined with a TMD parton density, as described later, the integral over $k_t$of the TMD density must agree with the collinear (k_t-integrated) density; this feature is guaranteed by construction for the PB-TMDs, which are also available as integrated PDFs in LHAPDF format.

When transverse momenta of the initial partons from TMDs are to be included to the hard scattering process, which was originally calculated under the assumption of collinear initial partons, care has to be take that energy and momentum are still conserved. The procedure adopted in CASCADE3 is the following: for each initial parton, a transverse momentum is assigned according to the TMD density, and this system is rotated and booted to its center-of-mass frame. Since the initial state partons have transverse momentum, they acquire a virtuality. The momenta of the incoming partons are given in Sudakov representation, with $p^{(A)}(p^{(B)})$ being the four-momenta of the incoming particles:

$$a = x_a p^{(A)} + \bar{x}_b p^{(B)} + k_{t,a}$$
$$b = \bar{x}_a p^{(A)} + x_b p^{(B)} + k_{t,b}$$

with $x_{a,b}, \bar{x}_{a,b}$ being the light-cone momentum fractions of partons a, b.

The energy and longitudinal component of the initial momenta $p_{a,b}$ are recalculated taken this virtuality into account, by [21]:

$$E_{a,b} = \frac{1}{\sqrt{2\hat{s}}} \left(\hat{s} \pm (Q_b^2 - Q_a^2) \right) \quad (2)$$

$$p_{z\,a,b} = \pm \frac{1}{2\sqrt{\hat{s}}} \sqrt{(\hat{s} + Q_a^2 + Q_b^2)^2 - 4Q_a^2 Q_b^2} \quad (3)$$

where Q_1^2 and Q_2^2 are the virtualities of parton $1, 2$ after the transverse momentum is assigned. The final partons of the hard system are rotated and boosted to their center-of-mass frame. Then the whole system of initial and final state partons is boosted and rotated back to its original system. This procedure is similar to the procedure applied in standard parton showers like PYTHIA, when a transverse momentum is created from the shower.

The TMD $\mathcal{A}(x, k_t, \mu)$ depends on the factorization scale μ as well as on the transverse momentum k_t: the factorization scale μ is calculated in the hard process and is the same as used in the evaluation of the collinear parton density, the transverse momentum $k_t$is limited by the so called *shower scale* which is the scale up to which the parton shower is allowed to contribute. Technically the factorization scale μ is calculated within CASCADE3 (see parameter `lhescale`) as it is not directly accessible from the LHE file, while the *shower scale* is given by `SCALUP`. With this choice of the *shower scale* it is guaranteed that the the TMD and later the parton shower does not generate transverse momenta which would violate the collinear factorization ansatz.

The advantage of using TMDs from the beginning is that the kinematics are fixed, independent of simulating explicitly the partons from the parton shower. For inclusive processes, for example inclusive Drell-Yan processes, the details of the hadronic final state generated by a parton shower do not matter, but the only net effect of the transverse momentum distribution. The parton shower, as described below, follows closely the transverse momentum distribution of the TMD and thus does not change any kinematic distribution after the TMD is included.

All hard processes, which are available in MADGRAPH5_aMC@NLO can be used within CASCADE3. The treatment of mulitjet merging is described in Section 7.

2.2 Off-shell processes

Several processes with off-shell matrix elements are implemented in CASCADE3 as listed in Tab. 1, and described in detail in [33].

However, many more processes are accessible via the automated matrix element calculator KATIE [34] or PEGASUS [35] for off-shell kinematics for the initial state partons. The events from the hard process are read into the CASCADE3 package via LHE files. For processes generated with KATIE no further corrections need to be performed and the event can be directly passed to the showering procedure, described in the next section.

When using off-shell processes, BFKL or CCFM type parton densities should be used, in order to allow for transverse momenta, k_t, which can be larger that the transverse momentum of any of the partons of the hard process. Until now, only gluon densities obtained from CCFM [13–16] or BFKL [36–38] are available, thus limiting the advantages of using off-shell matrix elements to gluon induced processes.

3 Initial State Parton Shower based on TMDs

3.1 PB TMD evolution and parton shower

The parton shower, which is described here, follows consistently the parton evolution of the TMDs. By this we mean that the splitting functions P_{ab}, the order in α_s, the scale in the calculation of α_s as well as the kinematic restrictions applied are identical in both the parton shower and the evolution of the parton densities.

Lepto(photo)production	process	IPRO	Reference
	$\gamma^* g^* \to q\bar{q}$	10	[22]
	$\gamma^* g^* \to Q\bar{Q}$	11	[22]
	$\gamma^* g^* \to J/\psi g$	2	[23–26]
Hadroproduction			
	$g^* g^* \to q\bar{q}$	10	[22]
	$g^* g^* \to Q\bar{Q}$	11	[22]
	$g^* g^* \to J/\psi g$	2	[26]
	$g^* g^* \to \Upsilon g$	2	[26]
	$g^* g^* \to \chi_c$	3	[26]
	$g^* g^* \to \chi_b$	3	[26]
	$g^* g^* \to J\psi J\psi$	21	[27]
	$g^* g^* \to h^0$	102	[28]
	$g^* g^* \to ZQ\bar{Q}$	504	[29,30]
	$g^* g^* \to Zq\bar{q}$	503	[29,30]
	$g^* g^* \to Wq_iQ_j$	514	[29,30]
	$g^* g^* \to Wq_iq_j$	513	[29,30]
	$qg^* \to Zq$	501	[31]
	$qg^* \to qg$	10	[32]
	$gg^* \to gg$	10	[32]

Table 1: *Processes included in* CASCADE3. *Q stands for heavy quarks, q for light quarks.*

A backward evolution method, as now common in Monte Carlo event generators, is applied for the initial state parton shower, evolving from the large scale of the matrix-element process backwards down to the scale of the incoming hadron. However, in contrast to the conventional parton shower, which generates a transverse momentum of the initial state partons during the backward evolution, the transverse momentum of the initial partons of the hard scattering process is fixed by the TMD and the parton shower does not change the kinematics. The transverse momenta during the cascade follow the behavior of the TMD. The hard scattering process is obtained as described in section 2.

The backward evolution of the initial state parton shower follows very closely the description in [21,33,39,40]. The evolution scale μ is selected from the hard scattering process, as described either in Section 2.1 or directly from the calcualtion in Section 2.2. In case of on-shell matrix elements, the transverse momentum of the hardest parton in the parton shower evolution is limited by the *shower-scale*, as described in Section 2.1.

Starting with the hard scale $\mu = \mu_i$, the parton shower algorithm searches for the next scale μ_{i-1} at which a resolvable branching occurs. This scale μ_{i-1} is selected from the Sudakov form factor Δ_S making use of the TMD densities $\mathcal{A}_a(x', k_t', \mu')$ which depend on the longitudinal momentum fraction $x' = xz$ of parton a, its transverse momentum k_t' probed at a scale μ' (see also [33]). The Sudakov form factor Δ_S for the backward evolution is given by (see fig. 1 left):

$$\Delta_S(x, \mu_i, \mu_{i-1}) = \exp\left[-\int_{\mu_{i-1}}^{\mu_i} \frac{d\mu'}{\mu'} \frac{\alpha_s(\tilde{\mu}')}{2\pi} \sum_a \int dz P_{a\to bc}(z) \frac{x'\mathcal{A}_a(x', k_t', \mu')}{x\mathcal{A}_b(x, k_t, \mu')}\right] \quad (4)$$

which describes the probability that parton b remains at x with transverse momentum k_t when evolving from μ_i to $\mu_{i-1} < \mu$. Please note, that the argument in α_s is $\tilde{\mu}'$ and depends on the ordering condition as discussed later. [1]

In the parton shower language, the selection of the next branching comes from solving the Sudakov form factor eq.(4) for μ_{i-1}. However, to solve the integrals in eq.(4) numerically for every branching would be too time consuming, instead the veto-algorithm [21,41] is applied. The selection of μ_{i-1} and the branching splitting z_{i-1} follows the standard methods [21].

The splitting function P_{ab} as well as the argument $\tilde{\mu}$ in the calculation of α_s is chosen exactly as used in the evolution of the parton density. In a parton shower one treats "resolvable" branchings, defined via a cut in $z < z_M$ in the splitting function to avoid the singular behavior of the terms $\frac{1}{1-z}$, and branchings with $z > z_M$ are regarded as "non-resolvable" and are treated similarly as virtual corrections: they are included in the Sudakov form factor Δ_S. This is exactly the same as in the determination of TMD densities within the PB or CCFM methods: a resolvable branching is defined via a cut $z < z_M$. The longitudinal momentum fraction $x_{i-1} = \frac{x_i}{z_{i-1}}$ is calculated by generating z_{i-1} according to the splitting function. With z_{i-1} and μ_{i-1} all variables needed for a collinear parton shower are obtained.

[1]In equation eq.(4) ordering in μ is assumed, if angular ordering, as in CCFM [13–16], is applied then the ratio of parton densities would change to $\frac{x'\mathcal{A}_a(x',k_t',\mu'/z)}{x\mathcal{A}_b(x,k_t,\mu')}$ as discussed in [33].

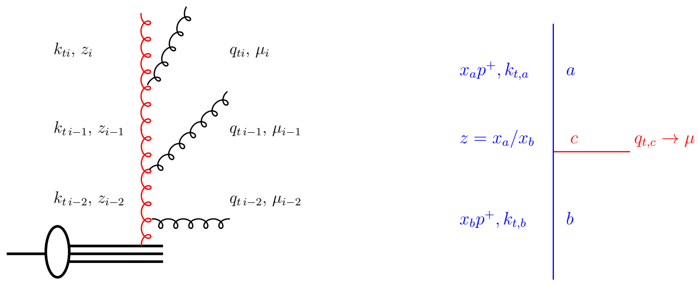

Figure 1: Left: Schematic view of a parton branching process. Right: Branching process $b \to a + c$.

The calculation of the transverse momentum k_t is sketched in fig. 1 right. The transverse momentum $q_{t\,i}$ can be obtained by giving a physical interpretation to the evolution scale μ_i (see fig. 1 right), and $q_{t\,i}$ can be calculated in case of angular ordering (μ is associated with the angle of the emission) in terms of the angle Θ of the emitted parton wrt the beam directions $q_{t,c} = (1-z)E_b \sin\Theta$:

$$\mathbf{q}_{t,i}^2 = (1-z)^2 \mu_i^2 \quad . \tag{5}$$

Once the transverse momentum of the emitted parton q_t is known, the transverse momentum of the propagating parton can be calculated from

$$\mathbf{k}_{t\,i-1} = \mathbf{k}_{t\,i} + \mathbf{q}_{t\,i-1} \tag{6}$$

with a uniformly distributed azimuthal angle ϕ is assumed for the vector components of $\mathbf{k}$ and $\mathbf{q}$. The generation of the parton momenta is performed in the center-of-mass frame of the collision (in contrast to conventional parton showers, which are generated in different partonic frames).

The whole procedure is iterated until one reaches a scale $\mu_{i-1} < q_0$ with q_0 being a cut-off parameter, which can be chosen to be the starting evolution scale of the TMD. However, it turns out that during the backward evolution the transverse momentum k_t can reach large values, even for small scales μ_{i-1}, because of the random ϕ distribution. On average the transverse momentum decreases, but it is of advantage to continue the parton shower evolution to a scale $q_0 \sim \Lambda_{qcd} \sim 0.3$ GeV, to allow emissions to share the transverse momenta generated.

3.2 CCFM parton evolution and parton shower

The CCFM parton evolution and corresponding parton shower follows a similar approach as described in the previous section and in detail also in Refs. [33,39,40]. The main difference to the PB-TMD shower are the splitting functions with the non-Sudakov form factor Δ_{ns} The original CCFM splitting function $\tilde{P}_g(z, q, k_t)$ for branching $g \to gg$ is given by (neglecting finite terms as they are not obtained in CCFM at the leading infrared accuracy (cf p.72 in [15]):

$$\tilde{P}_g(z,q,k_t) = \frac{\bar{\alpha}_s(q(1-z))}{1-z} + \frac{\bar{\alpha}_s(k_t)}{z}\Delta_{ns}(z,q,k_t), \tag{7}$$

where the non-Sudakov form factor Δ_{ns} is defined as:

$$\log \Delta_{ns} = -\bar{\alpha}_s(k_t) \int_0^1 \frac{dz'}{z'} \int \frac{dq^2}{q^2} \Theta(k_t - q)\Theta(q - z' q_t)\,, \tag{8}$$

with $q_t = \sqrt{\mathbf{q}_t^2}$ defined in Eq.(5) and k_t being defined in Eq.(6).

4 The TMD parton densities

In the previous versions of CASCADE3 the TMD densities where part of the program. With the development of TMDLIB [42] there is easy access to all available TMDs, they can be selected, as before, via `PartonDensity` with a value > 100000. For example the TMDs from the parton branching method [17,18] are selected via `PartonDensity=102100 (102200)` for PB-NLO-HERAI+II-2018-set1 (set2).

Please note, that the features of the TMD parton shower are only fully available for the PB-TMD sets and the CCFM shower clearly need CCFM parton densities.

The parameter `MaxFactor` is used to set the scale factor for the maximum weight when generating the transverse momentum for onshell partons; this factor should be 1 but can be adjusted to ensure that only a small fraction of events have weights larger than the maximum (in the output check: `CAS_LHEREAD wt>wtmax`).

5 Final state parton showers

The final state parton shower uses the parton shower routine `PYSHOW` of PYTHIA. Leptons in the final state, coming for example from Drell-Yan decays, can radiate photons, which are also treated in the final state parton shower. Here the method from `PYADSH` of PYTHIA is applied, with the scale for the QED shower being fixed at ?????.

The default scale for the QCD final state shower is $\mu^2 = 2 \cdot (m_{1\,\perp}^2 + m_{2\,\perp}^2)$ (`ScaleTimeShower=1`), with $m_{1(2)\,\perp}$ being the transverse mass of the hard parton 1(2). Other choices are possible: $\mu^2 = \hat{s}$ (`ScaleTimeShower=2`) and $\mu^2 = 2 \cdot (m_1^2 + m_2^2)$ (`ScaleTimeShower=3`). In addition a scale factor can be applied: `ScaleFactorFinalShower`$\times\mu^2$ (default: `ScaleFactorFinalShower=1`).

6 Uncertainies

Uncertainties of QCD calculations mainly arise from missing higher order corrections, which are estimated by varying the factorization and renormalization scales up and down by typically a factor of 2. The scale variations are performed when calculating the matrix elements and are stored as additional weights in the LHE file, which are then passed directly to the HEPMC output file for further processing.

The uncertainties coming from the PDFs can also be calculated as additional weigh factors during the matrix element calcualtion. However, when using TMDs, additional uncertainties arise from the transverse momentum distribution of the TMD. The PB-TMDs come with uncertainties from the experimental uncertainties as well as from model uncertainties, as discussed in Ref. [43]. These uncertainties can be treated and applied as additional weight factors with the parameter `Uncertainty_TMD = 0`.

7 Multi-jet merging

8 Program description

In CASCADE3 all variables are declared as `Double Precision`. With CASCADE3 the source of PYTHIA 6.28 is included to avoid difficulties in linking.

The input parameters are steered via steering files. The new format of steering is discussed in Section 8.1, the format of previous versions, which is appropriate for the internal off-shell processes, is discussed in Section 8.2.

8.1 Input parameters - new format

```
&CASCADE_input
NrEvents = -1                   ! Nr of events to process
Process_Id = -1                 ! Read LHE file
!MaxFactor = 10                 ! Max scaling factor for accept/reject
Hadronisation = 0               ! Hadronisation on (=1)
SpaceShower = 1                 ! Space-like Parton Shower
SpaceShowerOrderAlphas=2        ! Order alphas in Space Shower
TimeShower = 1                  ! Time-like Parton Shower
ScaleTimeShower = 4             ! Scale choice for Time-like Shower
!                                 1: 2(m^2_1t+m^2_2t)
!                                 2: shat
!                                 3: 2(m^2_1+m^2_2)
!                                 4: 2*scalup (from lhe file)
!ScaleFactorFinalShower = 1.    ! scale factor for Final State Parton Shower
PartonEvolution = 2             ! type of parton evolution in Space-like Shower
!                                 1: CCFM
!                                 2: full all flavor TMD evolution
! EnergyShareRemnant = 4        ! energy sharing in proton remnant
!                                 1: (a+1)(1-z)**a, <z>=1/(a+2)=1/3
!                                  2: (a+1)(1-z)**a, <z>=1/(a+2)=mq/(mq+mQ
!                                  3: N/(z(1-1/z-c/(1-z))**2), c=(mq/mQ)**2
!                                  4: PYZDIS: KFL1=1
! Remnant = 0                     ! =0 no remnant treatment
PartonDensity = 102200             ! use TMDlib: PB-TMDNLO-set2
! PartonDensity = 102100           ! use TMDlib: PB-TMDNLO-set1
! TMDDensityPath= './share'       ! Path to TMD density for internal files
```

```
Uncertainty_TMD = 0                        ! calculate and store uncertainty TMD pdfs
lheInput='MCatNLO-example.lhe' ! LHE input file
lheHasOnShellPartons = 1       ! = 0 LHE file has off-shell parton configuration
lheReweightTMD = 0             ! Reweight with new TMD  given in PartonDensity
lheScale = 2                   ! Scale defintion for TMD
!                                 0: use scalup
!                                 1: use shat
!                                 2: use 1/2 Sum pt^2 of final parton/particles
!                                 3: use shat for Born and 1/2 Sum pt^2 of final parton(particle)
!                                 4: use shat for Born and max pt of most forward/backward
!                                    parton(particle)
lheNBornpart = 2               ! Nr of hard partons (particles) (Born process)
ScaleFactorMatchingScale = 2.  ! Scale factor for matching scale when including TMDs
! lheWeightId = 0              ! use weight Id = ...  as weight for LHE file
&End


&PYTHIA6_input
P6_Itune = 370                 ! Retune of Perugia 2011 w CTEQ6L1 (Oct 2012)
P6_MSTJ(45) = 4                ! Nr of flavors in final state shower: g->qqbar
P6_PMAS(4,1)= 1.6              ! charm mass
P6_PMAS(5,1)= 4.75             ! bottom mass
P6_MSTJ(48) =   1              ! (D=0), 0=no max. angle, 1=max angle def. in PARJ(85)
! P6_MSTU(111) =  1            ! = 0 : alpha_s is fixed, =1 first order; =2 2nd order;
! P6_PARU(112) = 0.2           ! lambda QCD
P6_MSTU(112)=   4              ! nr of flavours wrt lambda_QCD
P6_MSTU(113)=                  ! min nr of flavours for alphas
P6_MSTU(114)=   5              ! max nr of flavours for alphas
&End
```

8.2 Input parameters - old format

```
* OLD STEERING FOR CASCADE
*
* number of events to be generated
*
NEVENT 100
*
* +++++++++++++++++ Kinematic parameters +++++++++++++++
*
'PBE1'    1    0    -7000.   ! Beam energy
'KBE1'    1    0     2212    ! -11: positron, 22: photon  2212: proton
'IRE1'    1    0       1     ! 0: beam 1 has no structure
*                            ! 1: beam 1 has structure
'PBE2'     1   0     7000.   ! Beam energy
'KBE2'     1   0     2212    ! 11: electron, 22: photon 2212: proton
'IRE2'     1   0        1    ! 0:  beam 3 has no structure
*                            ! 1:  beam 2 has structure
'NFLA'     1   0        4    ! (D=5) nr of flavours used in str.fct
* +++++++++++++++ Hard subprocess selection ++++++++++++++++++
'IPRO'     1   0        2    ! (D=1)
```

```
*                             !  2: J/psi g
*                             !  3: chi_c
’I23S’      1    0       0    ! (D=0) select 2S or 3S state
’IPOL’      1    0       0    ! (D=0)  VM->ll (polariation study)
’IHFL’      1    0       4    ! (D=4) produced flavour for IPRO=11
*                             ! 4: charm
*                             ! 5: bottom
’PTCU’      1    0       1.   ! (D=0) p_t **2 cut for process
* ++++++++++++ Parton shower and fragmentation ++++++++++++
’NFRA’     1     0       1    ! (D=1) Fragmentation on=1 off=0
’IFPS’     1     0       3    ! (D=3) Parton shower
*                             ! 0: off
*                             ! 1: initial state PS
*                             ! 2: final state PS
*                             ! 3: initial and final state PS
’IFIN’     1     0       1    ! (D=1) scale switch for FPS
*                             ! 1: 2(m^2_1t+m^2_2t)
*                             ! 2: shat
*                             ! 3: 2(m^2_1+m^2_2)
’SCAF’     1     0       1.   ! (D=1) scale factor for FPS
’ITIM’     1     0       0    ! 0: timelike partons may not shower
*                             ! 1: timelike partons may shower
’ICCF’     1     0       1    ! (D=1) Evolution equation
*                             ! 1: CCFM
*                             ! 0: DGLAP
* +++++++++++++ Structure functions and scales +++++++++++++
’IRAM’     1       0       0       ! (D=0) Running of alpha_em(Q2)
*                  ! 0: fixed
*                  ! 1: running
’IRAS’     1       0       1       ! (D=1) Running of alpha_s(MU2)
*                  ! 0: fixed alpha_s=0.3
*                  ! 1: running
     ’IQ2S’     1       0        3       ! (D=1) Scale MU2 of alpha_s
*                                        !  1: MU2= 4*m**2 (only for heavy quarks)
*                                        !  2: MU2 = shat(only for heavy quarks)
*                                        !  3: MU2= 4*m**2 + pt**2
*                                        !  4: MU2 = Q2
*                                        !  5: MU2 = Q2 + pt**2
*                                        !  6: MU2 = k_t**2
     ’SCAL’     1       0        1.0     !  scale factor for renormalisation scale
     ’SCAF’     1       0        1.0     !  scale factor for factorisation scale*
*     ’IGLU’     1       0        1201   ! (D=1010)Unintegrated gluon density
*                                        ! > 10000 use TMDlib (i.e. 101201 for JH-2013-set1)
*                                        !  1201: CCFM set JH-2013-set1 (1201 - 1213)
*                                        !  1301: CCFM set JH-2013-set2 (1301 - 1313)
*                                        !  1001: CCFM J2003 set 1
*                                        !  1002: CCFM J2003 set 2
*                                        !  1003: CCFM J2003 set 3
*                                        !  1010: CCFM set A0
*                                        !  1011: CCFM set A0+
*                                        !  1012: CCFM set A0-
```

```
*                                       !  1013: CCFM set A1
*                                       !  1020: CCFM set B0
*                                       !  1021: CCFM set B0+
*                                       !  1022: CCFM set B0-
*                                       !  1023: CCFM set B1
*                                       !  1: CCFM old set JS2001
*                                       !  2: derivative of collinear gluon (GRV)
*                              !  3: Bluemlein
*                 !  4: KMS
*                 !  5: GBW (saturation model)
*                 !  6: KMR
*                 !  7: Ryskin,Shabelski
* ++++++++++++ BASES/SPRING Integration procedure ++++++++++++
     'NCAL'     1       0      50000     ! (D=20000) Nr of calls per iteration for bases
     'ACC1'     1       0       1.0      ! (D=1) relative prec.(%) for grid optimisation
     'ACC2'     1       0       0.5      ! (0.5) relative prec.(%) for integration
* ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*     'INTE'     1       0        0      ! Interaction type (D=0)
*                                       ! = 0 electromagnetic interaction
*     'KT1 '     1       0      0.44     ! (D=0.0) intrinsic kt for beam 1
*     'KT2 '     1       0      0.44     ! (D=0.0) intrinsic kt for beam 2
*     'KTRE'     1       0      0.35     ! (D=0.35) primordial kt when non-trivial
*                                       ! target remnant is split into two particles
* Les Houches Accord Interface
     'ILHA'     1       0        0      ! (D=10) Les Houches Accord
*                                       ! = 0  use internal CASCADE
*                                       ! = 1  write event file
*                                       ! = 10 call PYTHIA for final state PS and remnant frag
* path for updf files
*     'UPDF'   '/Users/jung/jung/cvs/cascade2/cascade-2.0.1/share'
```

8.3 Random Numbers

CASCADE3 uses the `RANLUX` random number generator, with luxory level `LUX = 4`. The random number seed can be set via the environment variable `CASEED`, the default value is `CASEED=12345`

8.4 Event Output

When `HEPMC` is included, generated events are written out in HEPMC [44] format for further processing. The environment variable `HEPMCOUT` is used to specify the file name, by default this variable is set to `HEPMCOUT=/dev/null`.

The HEPMC events can be further processedfor example with Rivet [45].

9 Program Installation

CASCADE3 now follows the standard AUTOMAKE convention. To install the program, do the following

```
1) Get the source

tar xvfz cascade-XXXX.tar.gz
cd cascade-XXXX

2) Generate the Makefiles (do not use shared libraries)
./configure --disable-shared --prefix=install-path --with-lhapdf="lhapdflib_path"
--with-tmdlib="TMDlib-path" --with-gsl="gsl_lib" --with-hepmc="hepmc_path"

with (as example):
lhapdflib_path=/afs/desy.de/group/alliance/mcg/public/MCGenerators/lhapdf/5.8.1/i586_rhel40

3) Compile the binary
make

4) Install the executable and PDF files
make install

4) The executable is in bin
run it with:
export CASED=1242425
export HEPMCOUT=outfile.hepmc

cascade < steer_pp-LHEin
```

References

[1] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, *et al.*, JHEP **1407**, 079 (2014). 1405.0301.

[2] S. Frixione and B. R. Webber (2006). hep-ph/0612272.

[3] S. Frixione, P. Nason, and B. R. Webber, JHEP **08**, 007 (2003). hep-ph/0305252.

[4] S. Frixione and B. R. Webber (2002). hep-ph/0207182.

[5] S. Frixione and B. R. Webber, JHEP **0206**, 029 (2002). hep-ph/0204244.

[6] S. Alioli, K. Hamilton, P. Nason, C. Oleari, and E. Re, JHEP **04**, 081 (2011). 1012.3380.

[7] S. Frixione, P. Nason, and C. Oleari, JHEP **0711**, 070 (2007). 0709.2092.

[8] M. Bahr, S. Gieseke, M. Gigg, D. Grellscheid, K. Hamilton, *et al.*, Eur.Phys.J. **C58**, 639 (2008). 0803.0883.

[9] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, Comput. Phys. Commun. **191**, 159 (2015). 1410.3012.

[10] J. Alwall, S. Hoche, F. Krauss, N. Lavesson, L. Lonnblad, *et al.*, Eur.Phys.J. **C53**, 473 (2008). 0706.2569.

[11] R. Frederix and S. Frixione, JHEP **12**, 061 (2012). 1209.6215.

[12] K. Hamilton, P. Nason, and G. Zanderighi, JHEP **10**, 155 (2012). 1206.3572.

[13] M. Ciafaloni, Nucl. Phys. **B296**, 49 (1988).

[14] S. Catani, F. Fiorani, and G. Marchesini, Phys. Lett. **B234**, 339 (1990).

[15] S. Catani, F. Fiorani, and G. Marchesini, Nucl. Phys. **B336**, 18 (1990).

[16] G. Marchesini, Nucl. Phys. **B445**, 49 (1995). hep-ph/9412327.

[17] F. Hautmann, H. Jung, A. Lelek, V. Radescu, and R. Zlebcik, JHEP **01**, 070 (2018). 1708.03279.

[18] F. Hautmann, H. Jung, A. Lelek, V. Radescu, and R. Zlebcik, Phys. Lett. **B772**, 446 (2017). 1704.01757.

[19] J. Alwall *et al.*, Comput. Phys. Commun. **176**, 300 (2007). hep-ph/0609017.

[20] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht, M. Schönherr, and G. Watt, Eur. Phys. J. **C75**, 132 (2015). 1412.7420.

[21] M. Bengtsson, T. Sjostrand, and M. van Zijl, Z. Phys. **C32**, 67 (1986).

[22] S. Catani, M. Ciafaloni, and F. Hautmann, Nucl. Phys. **B366**, 135 (1991).

[23] V. Saleev and N. Zotov, Mod. Phys. Lett. A **9**, 151 (1994). [Erratum: Mod.Phys.Lett.A 9, 1517–1518 (1994)].

[24] A. Lipatov and N. Zotov, Eur. Phys. J. C **27**, 87 (2003). hep-ph/0210310.

[25] S. Baranov and N. Zotov, J. Phys. G **29**, 1395 (2003). hep-ph/0302022.

[26] S. P. Baranov, Phys. Rev. **D66**, 114003 (2002).

[27] S. Baranov, Phys.Rev. **D84**, 054012 (2011).

[28] F. Hautmann, Phys. Lett. **B535**, 159 (2002). hep-ph/0203140.

[29] S. P. Baranov, A. V. Lipatov, and N. P. Zotov, Phys. Rev. **D78**, 014025 (2008). 0805.4821.

[30] M. Deak and F. Schwennsen, JHEP **09**, 035 (2008). 0805.3763.

[31] S. Marzani and R. D. Ball, Nucl. Phys. **B814**, 246 (2009). 0812.3602.

[32] M. Deak, F. Hautmann, H. Jung, and K. Kutak, JHEP **09**, 121 (2009). 0908.0538.

[33] H. Jung, S. Baranov, M. Deak, A. Grebenyuk, F. Hautmann, *et al.*, Eur.Phys.J. **C70**, 1237 (2010). 1008.0152.

[34] A. van Hameren, Comput. Phys. Commun. **224**, 371 (2018). 1611.00680.

[35] A. Lipatov, M. Malyshev, and S. Baranov, Eur. Phys. J. C **80**, 330 (2020). 1912.04204.

[36] E. A. Kuraev, L. N. Lipatov, and V. S. Fadin, Sov. Phys. JETP **44**, 443 (1976).

[37] E. A. Kuraev, L. N. Lipatov, and V. S. Fadin, Sov. Phys. JETP **45**, 199 (1977).

[38] I. I. Balitsky and L. N. Lipatov, Sov. J. Nucl. Phys. **28**, 822 (1978).

[39] H. Jung, Comput. Phys. Commun. **143**, 100 (2002). hep-ph/0109102.

[40] H. Jung and G. P. Salam, Eur. Phys. J. **C19**, 351 (2001). hep-ph/0012143.

[41] S. Platzer and M. Sjodahl, Eur.Phys.J.Plus **127**, 26 (2012). 1108.6180.

[42] F. Hautmann, H. Jung, M. Krämer, P. Mulders, E. Nocera, *et al.*, Eur. Phys. J. **C 74**, 3220 (2014). 1408.3015.

[43] A. Bermudez Martinez, P. Connor, F. Hautmann, H. Jung, A. Lelek, V. Radescu, and R. Zlebcik, Phys. Rev. **D99**, 074008 (2019). 1804.11152.

[44] M. Dobbs and J. B. Hansen, Comput. Phys. Commun. **134**, 41 (2001).

[45] A. Buckley, J. Butterworth, L. Lonnblad, D. Grellscheid, H. Hoeth, J. Monk, H. Schulz, and F. Siegert, Comput. Phys. Commun. **184**, 2803 (2013). 1003.0694.