

Event generators for collider physics

Andy Buckley

University of Glasgow

CHACAL 2024 Summer School,
Witwatersrand University, 18 Jan 2023



University
of Glasgow

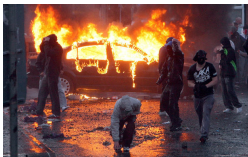


Introduction

Who am I?

Who am I?

- ▶ From Belfast, Northern Ireland:



(bad day)



(good day!)

Who am I?

- ▶ From Belfast, Northern Ireland:
- ▶ Physics degree & PhD at Cambridge Uni



Who am I?

- ▶ From Belfast, Northern Ireland:
- ▶ Physics degree & PhD at Cambridge Uni
- ▶ Postdoc→fellowships→faculty in
Durham→Edinburgh→CERN→Glasgow



Who am I?

- ▶ From Belfast, Northern Ireland:
- ▶ Physics degree & PhD at Cambridge Uni
- ▶ Postdoc→fellowships→faculty in
Durham→Edinburgh→CERN→Glasgow

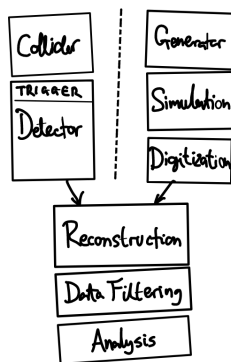


- ▶ ⇒ **simulating and researching particle physics since 2001...**
(Yikes. It doesn't feel that long...)

Introduction

“Evgen” is simulation of the fundamental particle-collision process

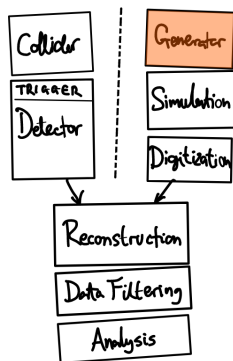
- ▶ Ubiquitous in HEP, from experiment design to interpretation
- ▶ I'll focus on fully differential “shower+hadronisation” gens (SHGs): those that make “realistic” events that can be fed to a detector simulation
- ▶ These are serious theory tools \Rightarrow the link between “hardcore theory” and experiment. Precision (and CPU cost) have rocketed in the last decade
- ▶ But often treated as *black boxes*. . . rarely a good idea. A little extra understanding can go a long way, so let's see what we can do!



Introduction

“Evgen” is simulation of the fundamental particle-collision process

- ▶ Ubiquitous in HEP, from experiment design to interpretation
- ▶ I'll focus on fully differential “shower+hadronisation” gens (SHGs): those that make “realistic” events that can be fed to a detector simulation
- ▶ These are serious theory tools \Rightarrow the link between “hardcore theory” and experiment. Precision (and CPU cost) have rocketed in the last decade
- ▶ But often treated as *black boxes*. . . rarely a good idea. A little extra understanding can go a long way, so let's see what we can do!



What is an event generator?

- ▶ “Event generator” is an overloaded phrase: many pheno people refer to partonic MC integrators as event generators.
- ▶ For experiment purposes a real EG produces *exclusive events*
 - Realistic final-particle multiplicities & composition, like real data
 - Fortunately HEP final-states *really* can be described in full detail
- ▶ **Correlations** are not easily fakeable, e.g. from sampling data distributions: *microscopic models* produce best and most richly structured phenomena
 - i.e. event generators are based on fundamental QFT
 - but *approximately*: can't explicitly calculate full-multiplicity processes
- ▶ **Since QCD is the strongest force, QCD effects usually dominate MC physics.** But this is changing, cf. precision era and large EW corrections for future colliders, for both lepton and hadron beams.

Shower & hadronisation event generators (SHGs)

What's an SHG good for? Depends who you ask!

- ▶ **Experimentalists:** design of colliders, detectors & analyses, background estimation, signal estimation, pile-up estimation, unfolding... ~**everything!**
- ▶ **Theory/pheno:** dressing parton level calculations to make them more realistic ("easily" include effects that aren't the focus of the study e.g. decays or UE); constraining BSM models by "recasting" experimental data
- ▶ **Generator authors:** understanding (how to work with) QCD – both perturbative and non-perturbative; enabling both the above

SHGs often take in partonic events via *LHE format* (though this is evolving), and output full particle-level events via *HepMC format(s)*



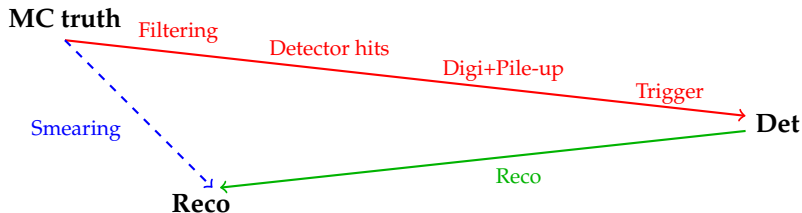
A selective list

- ▶ **Partonic subprocess generators (used as SHG input via LHE):**
 - **Multi-leg LO:** MadGraph5, Sherpa, AlpGen
 - **NLO (+ multileg):** POWHEG-BOX, Sherpa, OpenLoops, MadGraph5-aMC@NLO, MCFM
 - **Specialist processes:** HEJ, Prophecy4f, WHiZard, Protos
- ▶ **Main general-purpose event gens:**
 - **C++:** Sherpa, Herwig 7, Pythia 8
 - **FORTRAN:** FHerwig and FPythia
- ▶ **Afterburners:**
 - EvtGen, Photos, Tauola, Jimmy
- ▶ **Specialist all-in-one:**
 - **Min bias & air showers:** PHOJET, EPOS, QGSJET, SYBILL
 - **Heavy ion:** HIJING, HYDJET, Starlight, Angantyr
- ▶ **Utilities:**
 - LHAPDF, HepMC

Evgen in experiment data-processing

Typical experimental use of generators is to feed their output into a detector sim, e.g. based on Geant 4.

Then apply the same reconstruction + analysis as for data:

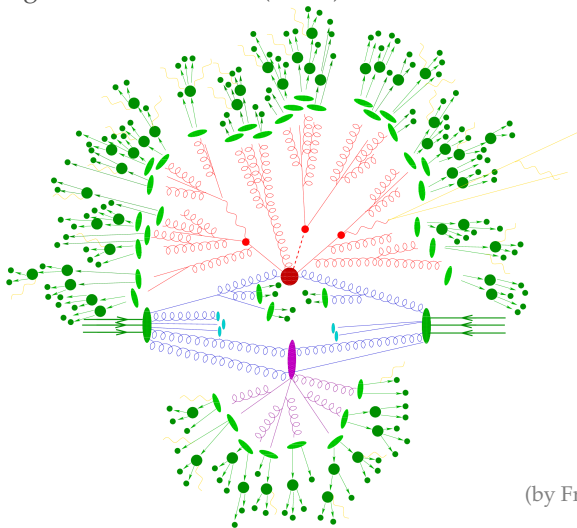


The generator bit of this chain was historically cheap: CPU/memory requirements *much* less than detector geometry + B -field stepping + material interaction + secondaries. **Not true these days!** ☞ ☞

Generator capabilities, complexity & CPU demands can now be very high: process complexity \Rightarrow inefficiency

Anatomy of an SHG

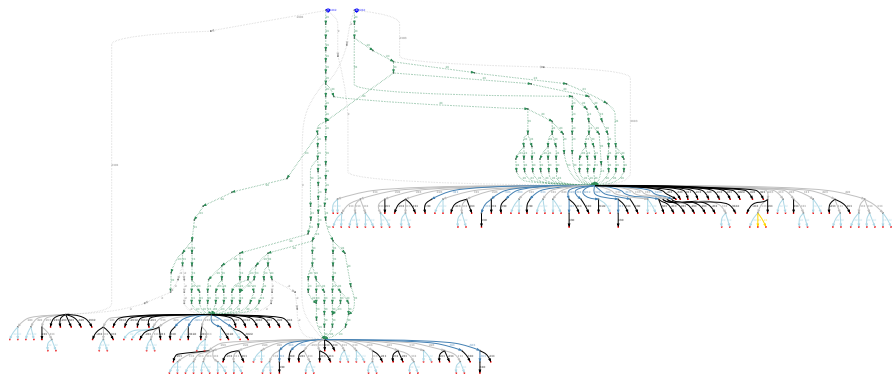
Computed outward, starting from the hard-scattering matrix element, and evolving back to the beams (PDFs) and forward to the final state:



(by Frank Krauss)

A “real” MC event

Here’s an MC event graph showing all the particles (lines) and interactions (vertices) in a Pythia *event record*.

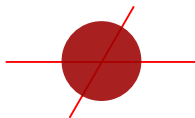


Guess the event type?

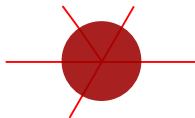
2212=proton, 1–6=quarks, 11–16=leptons, 21–25=gluon/photon/Z/W/H

Matrix elements

SHG step-by-step



SHG step-by-step



Matrix elements and phase-space

Cross-sections for a scattering subprocess $ab \rightarrow n$ computed in collinear factorization, with *factorization scale* μ_F :

$$\begin{aligned}\sigma &= \sum_{a,b} \int_0^1 \int_0^1 dx_a dx_b \int f_a^{h_1}(x_a, \mu_F) f_b^{h_2}(x_b, \mu_F) d\hat{\sigma}_{ab \rightarrow n}(\mu_F, \mu_R) \\ &= \sum_{a,b} \int_0^1 \int_0^1 dx_a dx_b \int d\Phi_n f_a^{h_1}(x_a, \mu_F) f_b^{h_2}(x_b, \mu_F) \\ &\quad \times \frac{1}{2\hat{s}} |\mathcal{M}_{ab \rightarrow n}|^2(\Phi_n; \mu_F, \mu_R),\end{aligned}$$

This is the core of all event generation: a combined *integral* of PDFs and **partonic matrix element** over phase space in $x_{a,b}$ and Φ .

The “MC” comes in because the integral is done by Monte Carlo **sampling** in $4(n-1) + 2$ phase-space dimensions: error reduces as $\sqrt{\text{samples}}$, rather than degrading with N_{dim} .

ME integration & generation

Event generation is preceded by an integration phase to evaluate total cross-section and map the $d\hat{\sigma}$ structure in ME phase-space

Naïve sampling doesn't get far: easy to spend all CPU on *atypical points* which are invalidated if and when a sample hits the typical set.

A better strategy:

- ▶ Jacobian-transform phase space to **remove divergent structures**;
- ▶ But there are *many* characteristic divergences in matrix elements
⇒ **multi-channel integration**: one per divergence, cf. MG5
- ▶ Use standard sampling techniques, or **adaptive sampling algorithms** — even ML — to “learn the space”
- ▶ It's still slow, and these methods are very far from perfect. The #1 MC production challenge is phase-space efficiency in NLO multijet processes

Integration can take *months*! Massively parallel computations becoming normal. **Typically save integration results as “gridpack” files for batched event generation.**

(Far) beyond LO

We are long past the point where SHGs only handle lowest-order partonic subprocesses (sometimes enhanced with LO *ME corrections* for the first parton-shower emission).

Extra partonic emissions at tree-level increase the final state multiplicity and change the event kinematics directly. Automated by LO *merging and matching schemes* like MLM, CKKW(-L), etc.

Beyond tree-level is more involved. An NLO cross-section has 3 parts:

$$d\sigma^{\text{NLO}} = d\tilde{\Phi}_n \left[\mathcal{B}(\tilde{\Phi}_n) + \alpha_s \mathcal{V}(\tilde{\Phi}_n) \right] + d\tilde{\Phi}_{n+1} \alpha_s \mathcal{R}(\tilde{\Phi}_{n+1})$$

But **infrared divergences** occur in both the *Real emission* and *Virtual correction* parts – i.e. in **different Φ dimensionalities**.

Bloch–Nordsieck / KLN theorems: for infra-red-safe observables, these divergences must cancel. cf. ME squaring

(Far) beyond LO

Subtraction: use **universal splitting kernel** \mathcal{S} which encodes real emission divergence structure so $\mathcal{R} - \mathcal{B} \otimes \mathcal{S}$ is finite \rightarrow computable:

$$\begin{aligned} \sigma^{\text{NLO}} = & \int_n d\tilde{\Phi}_n^{(4)} \mathcal{B} + \alpha_s \int_{n+1} d\tilde{\Phi}_{n+1}^{(4)} \left[\mathcal{R} - \mathcal{B} \otimes \mathcal{S} \right] \\ & + \alpha_s \int_n d\tilde{\Phi}_n^{(D)} \left[\tilde{\mathcal{V}} + \mathcal{B} \otimes \int_1 d\Phi_1^{(D)} \mathcal{S} \right], \end{aligned}$$

Many NLO ME calculators, but only a few automated ones. aMC@NLO and Sherpa fully automated; POWHEG-BOX is a framework to assist manual implementation.

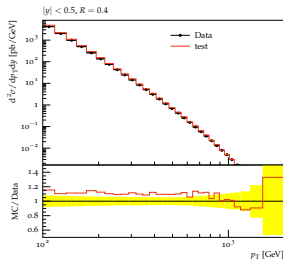
Virtual terms from dedicated calculators, e.g.

BlackHat/OpenLoops/NJETS/GoSAM via BLHA interface.

Technically solved: processes like $W + 5$ jets or **fully decayed $t\bar{t}$ and single-top** are possible at NLO... *if you can spare the integration time!*

Biased event generation and weights

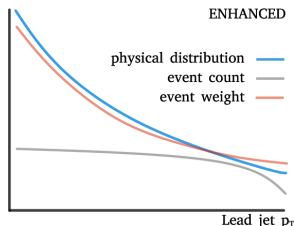
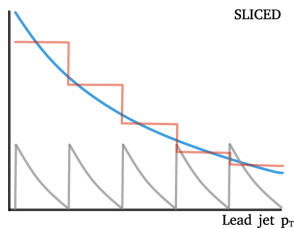
- ▶ For physics purposes, we want a flat distribution of event statistics across observables
- ▶ But many distributions fall fast: if we wait for an unbiased generator to produce a TeV-scale jet, we need to make as many events as the LHC does! *At sim-reco level...*
- ▶ Neat trick: bias the sampling to produce events not from a physical distribution but from an “enhanced” one, e.g. $\hat{p}_T^4 \cdot d\hat{\sigma}(\Phi)/d\omega$
- ▶ Experiments often also create piece-wise “sliced” samples with matched min and max cuts on subprocess \hat{p}_T



Distributions like this are hard
to make without biasing

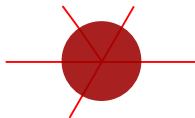
Biased event generation and weights

- ▶ For physics purposes, we want a flat distribution of event statistics across observables
- ▶ But many distributions fall fast: if we wait for an unbiased generator to produce a TeV-scale jet, we need to make as many events as the LHC does! *At sim-reco level...*
- ▶ Neat trick: bias the sampling to produce events not from a physical distribution but from an “enhanced” one, e.g. $\hat{p}_T^4 \cdot d\hat{\sigma}(\Phi)/d\omega$
- ▶ Experiments often also create piece-wise “sliced” samples with matched min and max cuts on subprocess \hat{p}_T

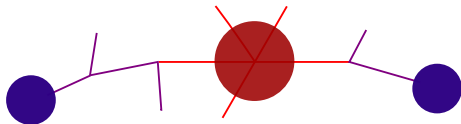


Parton showers

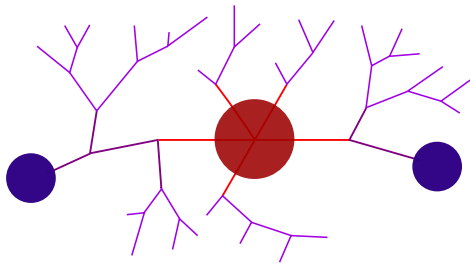
SHG step-by-step



SHG step-by-step



SHG step-by-step



Splitting functions and resummation

Limited parton–jet or parton–hadron duality: can compare *fixed-order partonic events* to data for **IR-safe** observables.

Soft and collinear phase-space, $1 \text{ GeV} < p_T \lesssim 20 \text{ GeV}$, gets large *resummation corrections* from multiple QCD emissions.

Analytic resummation is process-specific. But factorizes in the collinear limit: cross-section for process σ_0 with parton i to be accompanied by a collinear parton j with momentum fraction z :

$$d\sigma \approx \sigma_0 \sum_{\text{partons}, i} \frac{\alpha_s}{2\pi} \frac{d\theta^2}{\theta^2} dz P_{ji}(z, \phi) d\phi$$

where θ is the angle between i and j and

$$\begin{aligned} P_{qq}(z) &= C_F \frac{1+z^2}{1-z}, & P_{gq}(z) &= C_F \frac{1+(1-z)^2}{z}, \\ P_{gg}(z) &= C_A \frac{z^4+1+(1-z)^4}{z(1-z)}, & P_{qg}(z) &= T_R(z^2 + (1-z)^2) \end{aligned}$$

These are the spin-averaged QCD collinear *splitting functions*, or *DGLAP kernels*. Divergences as $\theta, z \rightarrow 0 \Rightarrow$ **soft & collinear emissions dominate**

Parton showers

Parton showers are Markov chain algorithms based on the QCD splitting functions \rightarrow process-independent, approximate resummation. Creates parton multiplicities \sim realistic hadron multiplicities.

From splitting functions can calculate *probability of no emission* between scales Q and q (setting an IR cutoff for resolvability & perturbativity):

$$\Delta_i(Q^2, q^2) = \exp \left\{ - \sum_j \int_{q^2}^{Q^2} \frac{dk^2}{k^2} \frac{\alpha_s}{2\pi} \int_{q^2/k^2}^{1-q^2/k^2} dz P_{ji}(z) \right\}$$

The famous *Sudakov form factor*. Can be inverted to generate a random parton emission with physical k distribution from a random number

Split into *initial-state* (ISR) and *final-state* (FSR) showers in SHGs:

- 1 ISR: generate high- p_T extra emissions on the incoming parton legs back to the proton (using *backward evolution*);
- 2 FSR: dress all final-state partons with a forward evolution down to the QCD perturbative cutoff Λ_{QCD}

Motivating the Sudakov

Integration is the natural continuous limit of a sum:

$$\Sigma \rightarrow \int$$

What's the equivalent continuum limit of a *product*?

$$\Pi \rightarrow ???$$

Motivating the Sudakov

Integration is the natural continuous limit of a sum:

$$\Sigma \rightarrow \int$$

What's the equivalent continuum limit of a *product*?

$$\prod = \exp \left\{ \Sigma \ln \right\} \rightarrow \exp \left\{ \int \ln \right\}$$

Every dt scale-step has a differential probability of no-emission; the Sudakov *exponentiation* multiplies them together.

More about showers

- ▶ In practice it's a bit more complex, with running α_s (and carefully chosen running scale), spin effects, quark masses, etc.
- ▶ Any evolution variable $k^2 \propto \theta^2$ is permitted in the collinear limit and will resum the divergence.
 - But some are better than others: colour-coherence effects suppress emissions outside the previous emission cone.
 - Quantum effect reproduced by θ -ordering and p_T -ordering, but not virtuality. All modern generators enforce a degree of colour coherence.
- ▶ Compared to the ME, Markov process \rightarrow shower misses correlations.



More about showers

- ▶ Initial state shower adds complication
 - Emissions modifying initial partons need to change the PDF x and flavour used \Rightarrow ISR Sudakovs include a PDF ratio term $\frac{x/z f_j(x/z, k^2)}{x f_i(x, k^2)}$ to \sim cancel up the chain
 - Forward-evolving from the hadron to find a consistent hard process configuration would be hopelessly inefficient \Rightarrow backward evolution.
- ▶ Actually, $1 \rightarrow 2$ showers have problems:
 - Can't have finite relative p_T and real, on-shell partons since $1 \rightarrow 2$ violates Lorentz symmetry \Rightarrow *reshuffling*
 - Much modern activity uses $2 \rightarrow 3$ dipole showers and higher variants – also for NLO subtraction compatibility: CSS, MatchBox, DIRE, Vincia, ...

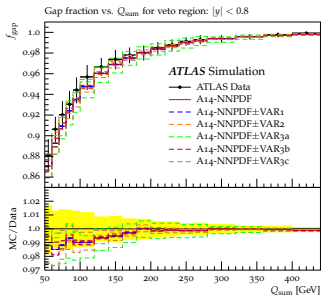
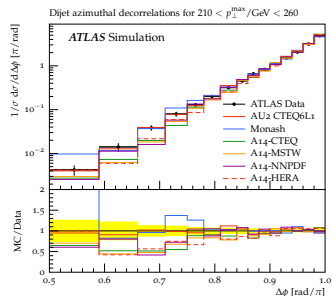
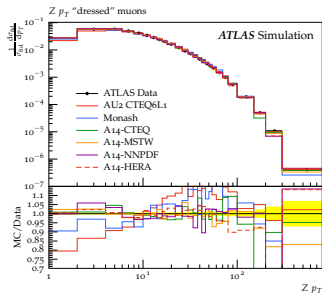


Forward evolution of spacelike shower, but more fun

Shower observables

ISR: extra jets, jet distributions,
 $Z p_T$, gaps

NB. distinctly not collinear!

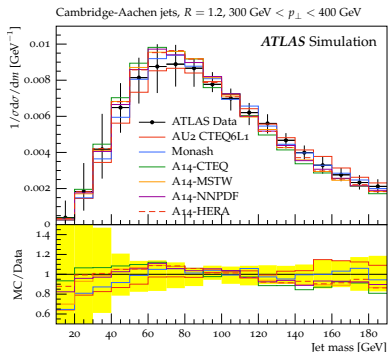
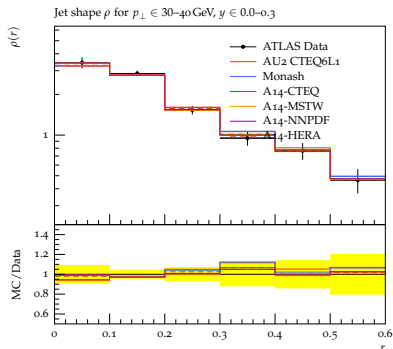
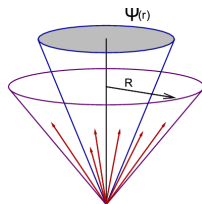


Shower observables

FSR: jet shapes, jet masses

i.e. adding structure to the parton \leftrightarrow jet duality

As jet scale increases, jets become increasingly collimated



ME–shower interfacing: “matching”

Really not enough space to do this topic justice: **huge** developments in last 20 years

Issues are almost always because of **double counting** when the shower is used: an n -leg ME with parton shower *contains* the $n + 1, \dots$ terms.

To improve on the Born+shower approximation, need to **remove overlap**.

For LO multi-leg: MLM and CKKW schemes both designed to **replace the shower’s collinear splitting functions with proper matrix elements in the relevant (hard) phase-space**

Phase-space slicing definitions took 10 years to iterate to better control. Introduces *merging scales*, which need to be chosen to minimise observable sensitivity: not “fire and forget” generation

Matching at NLO

Natural to go beyond tree-level matching: completely consistent NLO includes the Born, one-loop, and one-real emission amplitudes and all their interferences, *and* smoothly matches the real emissions between shower and ME.

Main benefit: NLO *scale stability*. Normalisation could always be taken from partonic highest-tech integrators, but (1-emission) shapes now also stable without disrupting formal accuracy of parton shower

⇒ **rough rule: shapes from real, normalisation from virtual**

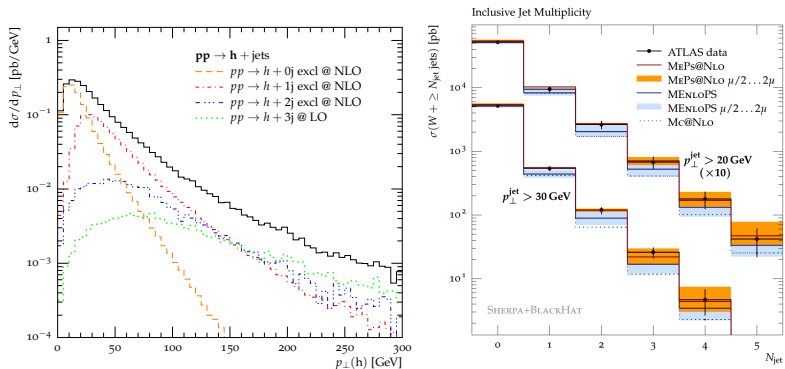
- ▶ **MC@NLO method:** extension of fixed-order subtraction to use shower-specific splitting functions: **process-specific, “some” negative weights.**
- ▶ **POWHEG¹ method:** “NLO matrix-element correction”. Closer to all-positive weights, and **shower-independent.** Convenience ⇒ large uptake.

¹Not, not, *not* ‘PowHeg’!!! ‘PoWHEG’, if you must.

State of the art: NLO matching + merging

Now very impressive situation: automatic generation & merging of many NLO and LO multi-leg + shower in MEPS@NLO (Sherpa) and FxFx (MG5-aMC@NLO)

Bookkeeping *tour de force!* And at huge CPU cost in *unweighting*

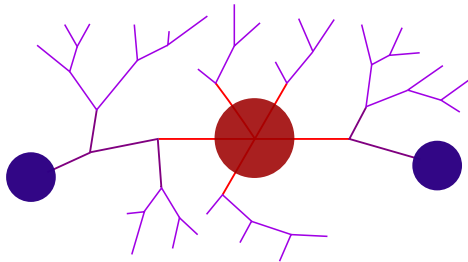


Increasingly NLO with EW corrections. NNLO-PS sometimes possible.

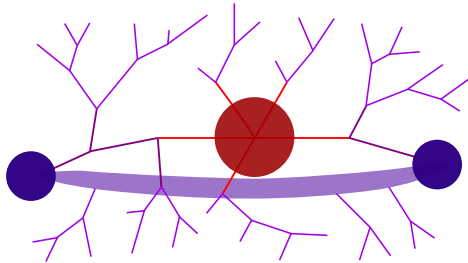
Which features are worth the CPU/opportunity cost?

Non-perturbative stuff
(that we wish wasn't there...)

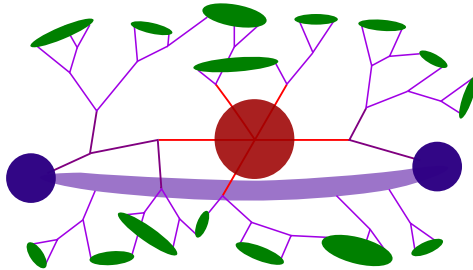
SHG step-by-step



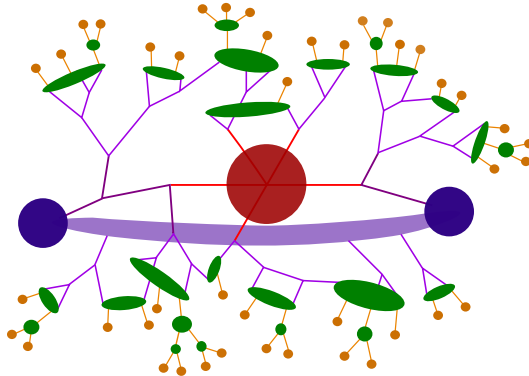
SHG step-by-step



SHG step-by-step

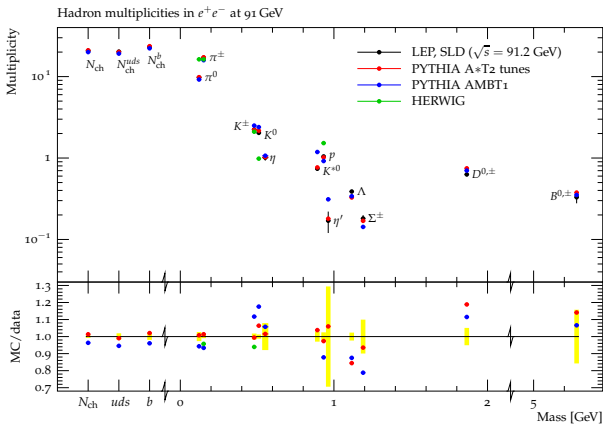


SHG step-by-step



Hadronisation

- ▶ At scales below shower cutoff $Q_0 \sim \mathcal{O}(\Lambda_{\text{QCD}})$, confinement means that physics is **non-perturbative**. **Source of most tuning params**
- ▶ Observe limited transverse momenta and Q^2 -independent energy fractions: most quantum number flow done by the shower fragmentation, so hadronisation can be \sim localised
- ▶ Two main modern hadronisation models: *Lund string and cluster*

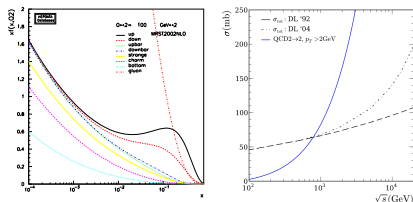


Hadronisation models

- ▶ **Lund string (Pythia):**
 - Inspired by *linear scaling of QCD potential* at large distances
 - Break colour strings to produce new quark pairs; gluons form **kinks** in strings
 - Lorentz invariance and LR-symmetry give Lorentz invariant *Lund symm frag function*
 - Kinematics well-described, but flavour – esp. baryons – not natural
- ▶ **Cluster hadronisation (Herwig, Sherpa):**
 - *Colour preconfinement*, seen in colour-connected neighbour parton mass spectrum
 - Non-pert $\rightarrow q\bar{q}$, then cluster colour singlets: requires finite gluon *constituent mass*
 - Clusters treated as *meson resonances*
- ▶ Both models (except Sherpa) also contain *colour reconnection* heuristics
- ▶ **Tuning!**

Underlying event

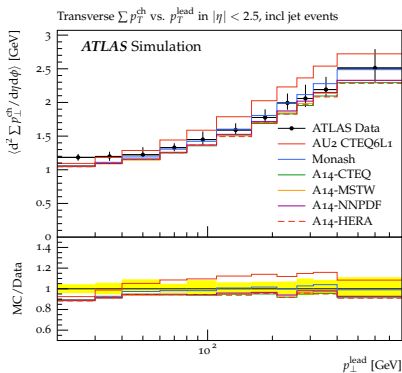
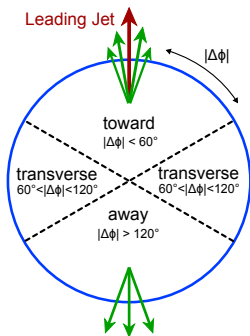
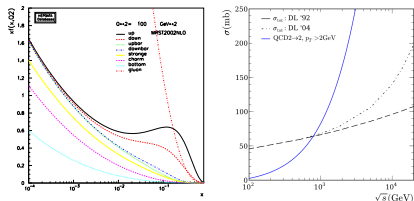
HERA data show inclusive jet cross-section rising strongly with energy due to low- x PDFs (esp. gluon)
 \Rightarrow unitarity violation



- ▶ Eikonal models interpret the bottom-up to top-down σ ratio as mean number of *multiple-parton interactions* (MPI) \Rightarrow **sample Poisson to make n pQCD subprocesses**
- ▶ Hadron impact parameter $\sim 1/Q \Rightarrow$ transverse overlap also important
- ▶ **Low hard-process scale $Q \Rightarrow$ low overlap & low n :** “minimum bias” cf. pile-up
- ▶ **High hard-process scale $Q \Rightarrow$ total overlap & high n :** *pedestal effect* \rightarrow “underlying event”
- ▶ **Extra details:** \hat{p}_T cutoff/screening, proton overlap form factor, colour reconnection. **Tuning!**

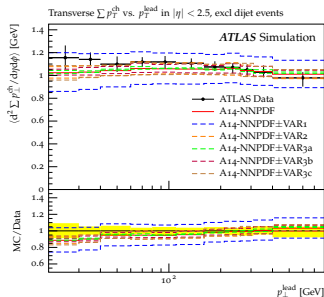
Underlying event

HERA data show inclusive jet cross-section rising strongly with energy due to low- x PDFs (esp. gluon)
 \Rightarrow unitarity violation



MC tuning

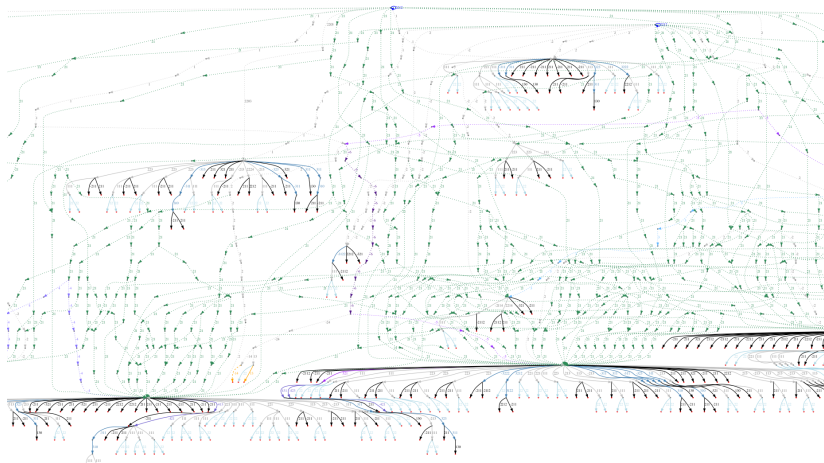
- ▶ Freedom to describe data with generator models, via the *ad hoc* and beyond-fixed-order components: MPI, hadronisation, \sim showers
- ▶ **Need to be careful!** A pragmatic trick at LO may backfire spectacularly when a better ME is added. **Knowing the limits of a generator configuration is important**
- ▶ A global view is crucial: one number/distribution can always be overtuned at cost of others
- ▶ Rivet & Professor/Apprentice tools used to build tunes & *eigentunes*



Using MC events... wisely

Example SHG event

(Part of) a single-top HepMC event graph:



Ouch! How to safely *use* this??

Caveats on event record interpretation

- ▶ The SHG generator (or LHE) event record is often called “truth” – a dangerous phrase.
- ▶ We’re doing quantum mechanics: there is no unambiguous truth!
⇒ event records are **half-physics, half-debug-info**... and zero indication of amplitude interference
- ▶ It gets worse: **kinematic frames aren’t defined (until the final-state) & momentum isn’t necessarily conserved at vertices!**
- ▶ **BEWARE!!**



Caveats on event record interpretation

- ▶ That said, like all good myths, there is a core of truth to the widespread physical interpretation of event records
- ▶ And sometimes precision EW or PDF theorists will request correction to partonic level rather than forward-folding of their calculations, e.g. “Born-level Z”.

NB. expts don't have to say “yes”!

- ▶ First **think about the physics** – e.g. is there a real distinction between hard photons and shower photons? Good discipline/introspection anyway!
- ▶ And first try to do what you need directly from the physical hadrons etc. See Rivet [↗](#) & ATLAS PUB note on safe truth observables [↗](#)

Apply brain!



Conclusions

Summary

- ▶ Event generators are super-, super-important for LHC physics
- ▶ And demands are only increasing: we demand processes and levels of data description (and predictivity) that would have been laughed at pre-LHC
- ▶ Both experiment and theory owe a great deal to the few phenomenologists who've provided us with these codes
- ▶ SHGs based on a core of perturbative QCD (& EW) of increasing sophistication. And increasing CPU cost...
- ▶ Wrapped with perturbative iterated parton showers \Rightarrow resum logs & generate a good approximation to "real" final-states. Plus pheno models and tuning for the stuff we don't understand *ab initio*
- ▶ **Follow-up material:**
 - MCnet review – [arXiv:1101.2599](#) ↗
 - Hoeche review – [arXiv:1411.4085](#) ↗
 - "Practical Collider Physics" – AB, White, White ↗
 - "QCD & Collider Physics" – Ellis, Stirling, Webber
 - "QCD" – Dissertori, Knowles, Schmelling
 - MCnet summer schools: 2018 ↗ 2019 ↗ 2022 ↗

Some hands-on exercises

Being able to run event generators yourself (rather than just using experiment samples) is a big advantage.

- ▶ Get Pythia 8 and generate QCD dijet events with ME \hat{p}_T cutoff of 50 GeV. Analyse with Rivet's MC_JETS analysis. What does it look like if you change the cutoff? How about the shower α_s ?
- ▶ Get the POWHEG-BOX heavy-flavour (h_{vq}) process and generate $t\bar{t}$ events. Shower with Pythia and compare to top-pair production in LO Pythia using also the MC_TTBAR analysis: differences in normalisation? shapes?
- ▶ Get MG5 and generate a $p p \rightarrow z \rightarrow \mu^+ \mu^-$ process with Pythia showering. Use `add process` to merge this with +1, 2 jets MEs. Analyse with Rivet or MadAnalysis5. What changes?
- ▶ Generate a BSM process, e.g. SUSY gluino pair production in Pythia's built-in processes (you will need an SLHA file, e.g. from an experiment publication), or a FeynRules UFO model like [SMEFTsim](#) ↗ in MG5

The MCnet [hepstore](#) ↗ Docker images may be useful