Stefano Frixione

Precision calculations: the NLO case

Amsterdam Particle Physics Symposium

Amsterdam, 30/11/2011

Typical uses of precision calculations:

Typical uses of precision calculations:

- Comparison to data

Typical uses of precision calculations:

- Comparison to data
- Extraction of parameters (masses, couplings,...)

Typical uses of precision calculations:

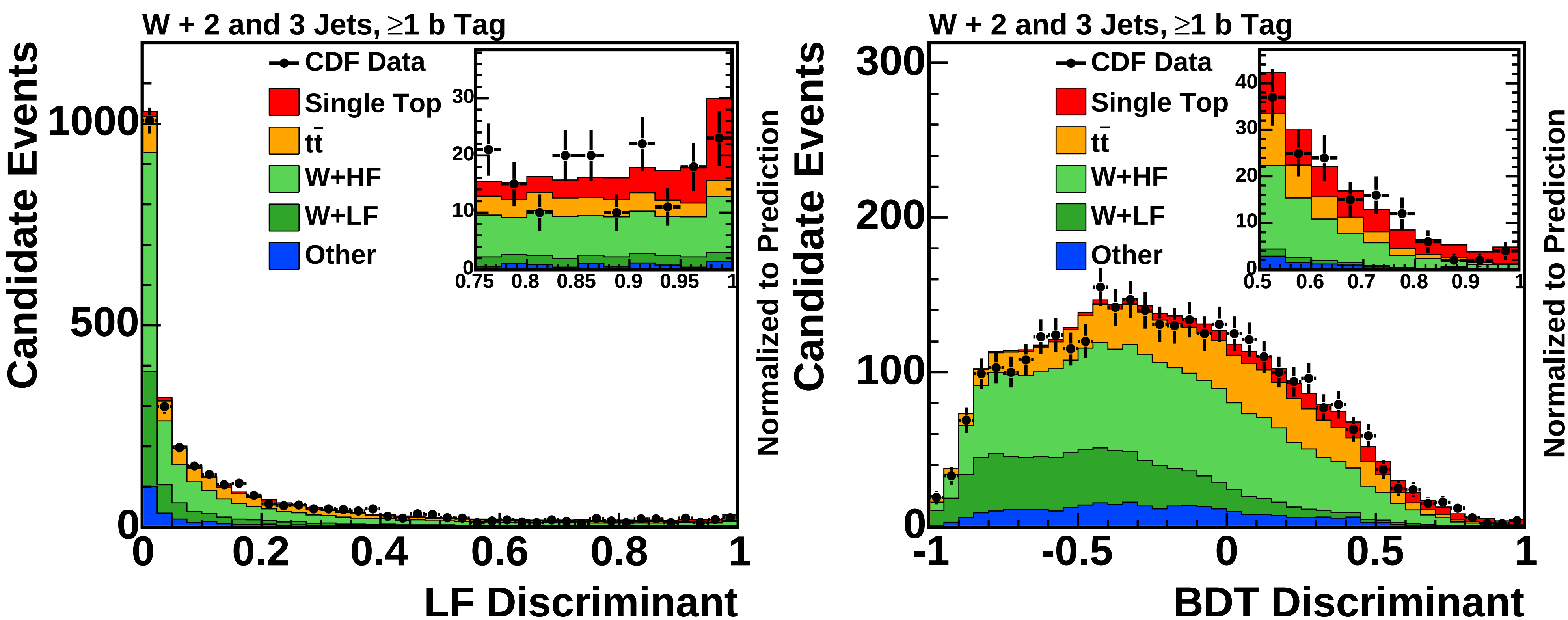
- Comparison to data
- Extraction of parameters (masses, couplings,...)
- Boost a theorist's self-esteem (e.g., my calculation is more difficult than yours)

Typical uses of precision calculations:

- Comparison to data
- Extraction of parameters (masses, couplings,...)
- Boost a theorist's self-esteem (e.g., my calculation is more difficult than yours)

But: things have changed with the coming of the LHC, and the more advanced Tevatron analyses

Single top at CDF (1004.1181)



Cut and count hopeless: uncertainty much larger than signal

Single-top discovery at the Tevatron is a paradigm for (some) LHC analyses

- If predictions for signal are wrong, there is no safety net
- For backgrounds: one must not overstretch predictions when tuning, since this may "hide" a signal. In other words: shapes must be trustable
- It is important to use fully-exclusive theoretical results, that can go through detector simulation

So what is the lesson to be learned?

Accurate, fully-differential, realistic (i.e., do not use standalone Pythia/Herwig for processes other than $2 \to 2$), and hadron-level predictions may play *a very important role*

Accurate, fully-differential, realistic (i.e., do not use standalone Pythia/Herwig for processes other than $2 \to 2$), and hadron-level predictions may play *a very important role*

Does this mean NLO in perturbation theory?

Accurate, fully-differential, realistic (i.e., do not use standalone Pythia/Herwig for processes other than $2 \to 2$), and hadron-level predictions may play *a very important role*

Does this mean NLO in perturbation theory?

YES

Accurate, fully-differential, realistic (i.e., do not use standalone Pythia/Herwig for processes other than $2 \to 2$), and hadron-level predictions may play *a very important role*

Does this mean NLO in perturbation theory?

YES

But beware: the usual motivations given by many theorists (e.g.: better description of jet structure; extra contributions from initial-state partons; "NLO" effects on distributions) are actually motivations for *tree-level* calculations (*beyond LO*)

As far as I'm concerned, the really crucial thing is:

- Precision: NLO is the first order at which the assessment of theoretical uncertainties is reliable (for both postdictions and predictions)

As far as I'm concerned, the really crucial thing is:

- Precision: NLO is the first order at which the assessment of theoretical uncertainties is reliable (for both postdictions and predictions)

This is even more true for N^{k}LO, $k \geq 2$, so what does the trick is:

- We can perform an NLO calculation (almost) as straightforwardly as an LO one
- Same for its matching with parton showers

Note: these two items are very recent advances, which enlarged the scope of NLO results far beyond what was previously thought possible

Timeline

1979 Passarino-Veltman tensor reduction

1980 Ellis-Ross-Terrano: subtraction method (observable dependent)

1980-circa 1995 Pain, pain, and more pain

1995-1997 Discovery of process-independent subtraction procedures

1997-circa 2008 Less pain, but pain still

$\sim$ 2000 Tree-level calculations fully automated

2002-2005 Discovery of NLO-MC matching techniques

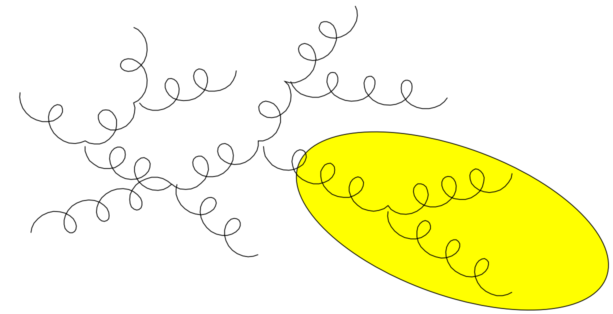
2005-2008 (Re)-discovery of process-independent one-loop techniques

2009-present Automation – pain is over

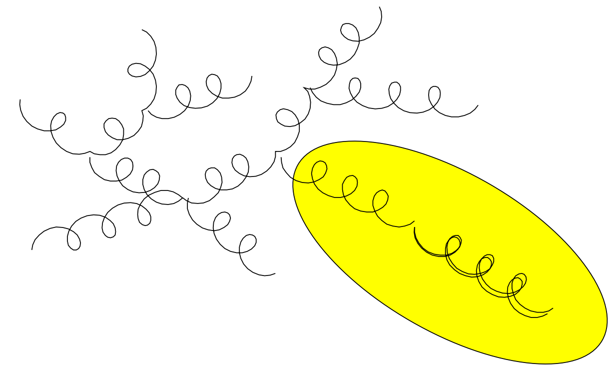
Fixed order

Anatomy of NLO computations

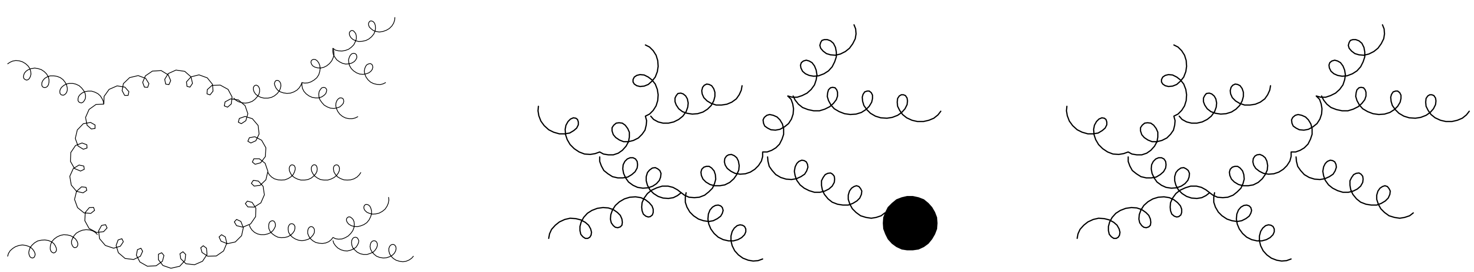
$$\frac{d\sigma}{dO} = \int d\phi_{n+1} \Big(\mathcal{M}^{(r)}(\phi_{n+1})\delta(O - O_{n+1}(\phi_{n+1}))$$



$$-\mathcal{M}^{(c.t.)}(\mathcal{P}\phi_{n+1})\delta(O - O_n(\mathcal{P}\phi_{n+1}))\Big)$$



$$+ \int d\phi_n \Big(\mathcal{M}^{(v)}(\phi_n) + \mathcal{M}^{(rem)}(\phi_n) + \mathcal{M}^{(b)}(\phi_n) \Big) \delta(O - O_n(\phi_n))$$



Things to do:

- Compute the real, Born, and one-loop matrix elements
- Subtract the singularities of the real matrix elements, thus cancelling the one-loop ones
- Parametrize the phase space, and integrate the (finite) results of the previous step

Major bottlenecks were the subtraction and the one-loop computations

Subtraction

Consider a $2 \to n$ pure-gluon process. There are

- $(n^2 + 3n)/2$ collinear singularities (two-body correlations)
- n soft singularities (three-body correlations)

Their systematic subtractions for any n in a process-independent manner is a solved problem

- FKS (Frixione, Kunszt, Signer, hep-ph/9512328 + ...)
- Dipole (Catani, Seymour, hep-ph/9605323 + ...)
- Antenna (Kosower, hep-ph/9720213)

An alternative technique is slicing (Owens, Harris; Laenen, Keller; ...), not suited to large n

FKS

- Use collinear singularities to organize subtractions $\Longrightarrow$ two-body kernels
- Define subtractions on-shell
- Use arbitrary functions (whose sum is equal to one) to damp all singularities except one collinear and one soft. These newly-constructed quantities are treated independently from each other

Dipole

- Use soft singularities to organize subtractions $\Longrightarrow$ three-body kernels
- Define subtractions off-shell. The recoil is distributed using a mapping defined by the three partons of each kernel. Hence, each subtraction term has a different kinematics
- All singularities are subtracted simultaneously

Implications:

- Number of subtraction terms scales as n^2 in FKS, and as n^3 in dipoles. By exploiting symmetries, FKS reduce this to a constant
  Variants of dipoles (Chung, Krämer, Robens, 2010) achieve n^2
- Numerics in dipoles become intractable for large n without the use of α-dependent subtractions (Nagy). This is not the case in FKS
- Importance sampling must be done dynamically in dipoles, not so in FKS
- FKS has a "collinear" structure. It is therefore the method of choice for NLO-parton shower matching formalisms (MC@NLO and POWHEG)
- Dipole is manifestly Lorentz invariant, FKS is not

Automation

- MadFKS (Frederix, Frixione, Maltoni, Stelzer 0908.4272)
- HELAC (Czakon, Papadopoulos, Worek 0905.0883)
- MadDipole (Frederix, Gehrmann, Greiner 1004.2905, 0808.2128)
- SHERPA (massless only) (Gleisberg, Krauss 0709.2881)
- Other less systematic attempts (Seymour, Tevlin 0803.2231; Hasegawa, Moch and Uwer 0911.4371)

Level and scope of automation differ, and at the moment it is difficult to assess the capabilities of these codes. I suppose dust will settle soon

One-loop computations

Several methods are now established:

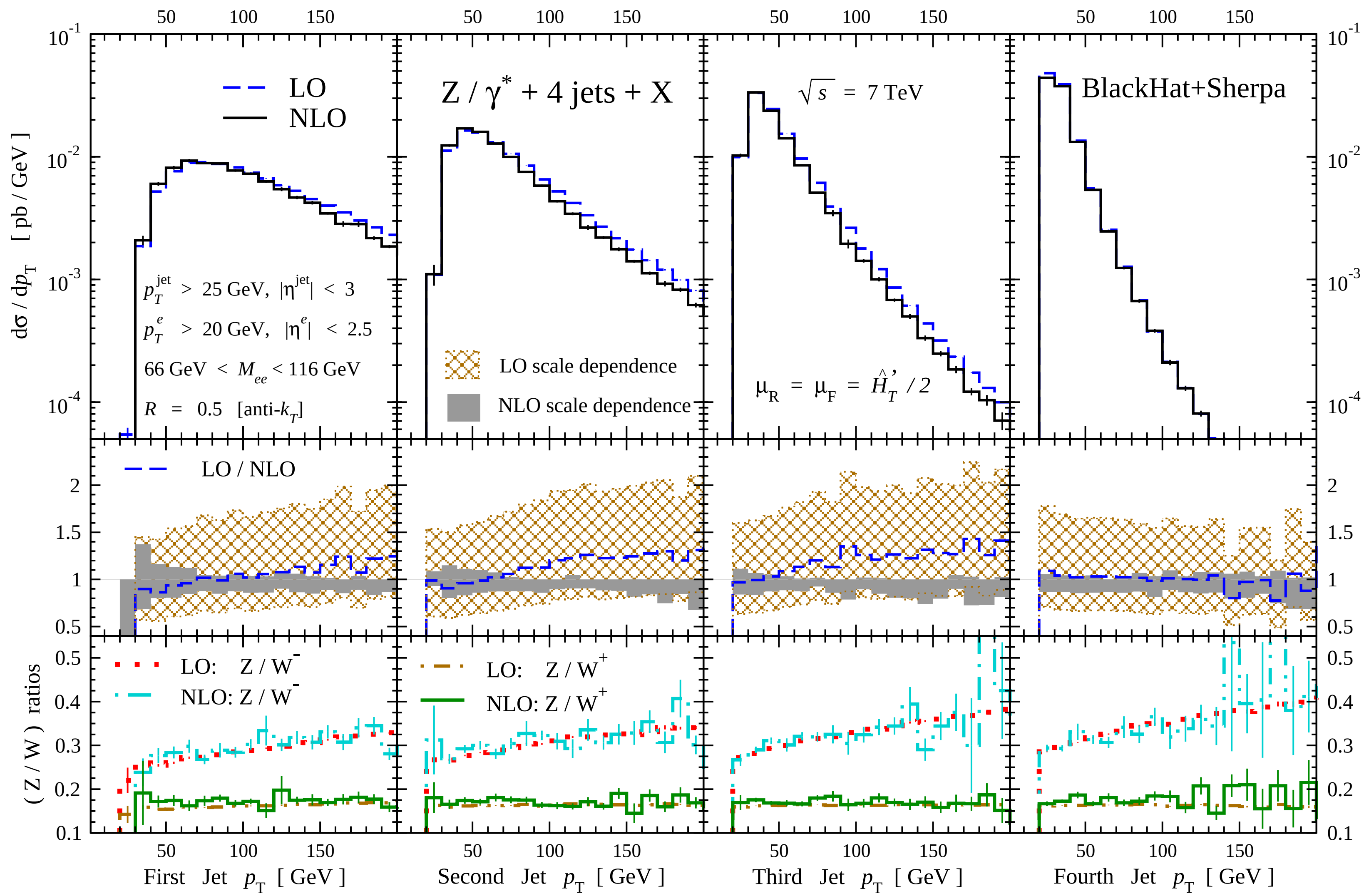
- Generalized Unitarity (Bern, Dixon, Dunbar, Kosower hep-ph/9403226 + ...; Ellis, Giele, Kunszt 0708.2398, +Melnikov 0806.3467)
- Integrand Reduction (Ossola, Papadopoulos, Pittau hep-ph/0609007; del Aguila, Pittau hep-ph/0404120; Mastrolia, Ossola, Reiter, Tramontano 1006.0710)
- Tensor Reduction (Passarino, Veltman 1979; Denner, Dittmaier hep-ph/0509141; Binoth, Guillet, Heinrich, Pilon, Reiter 0810.0992)

...and have been put to use and automated $\longrightarrow$

- GU $\longleftarrow$ BlackHat (Berger, Bern, Dixon, Febres Cordero, Forde, Gleisberg, Ita, Kosower, Maitre 1009.2338 + ...)
- GU $\longleftarrow$ Rocket (Ellis, Giele, Kunszt, Melnikov, Zanderighi 0810.2762 + ...)
- IR $\longleftarrow$ MadLoop (Hirschi, Frederix, Frixione, Garzelli, Maltoni, Pittau, 1103.0621), HELAC-NLO (Bevilacqua, Czakon, Garzelli, van Hameren, Kardos, Papadopoulos, Pittau, Worek, 1110.1499 + ...), GoSam (Cullen, Greiner, Heinrich, Luisoni, Mastrolia, Ossola, Reiter, Tramontano, 1111.2034)

Tensor Reduction, one of the two strategies used in GoSam, is not really suited to full automation

So far, GU applied mostly to large-multiplicity, massless final states (e.g. Z+4 jets by BlackHat), IR to lower-multiplicity, massive final states (e.g. HELAC $t\bar{t}b\bar{b}$)



$Z + 4$ jets, BlackHat+SHERPA, 1108.2229

Simply unthinkable just a few years ago

Matching to showers (NLOwPS)

This problem has attracted a lot of attention in the theory community, being quite challenging (and very relevant to phenomenology)

Main issue: MC's and NLO's "generate" identical classes of Feynman diagrams, that must not be counted twice

Why it is tricky: NLO computations are inclusive by nature; MC's are fully exclusive. Opposite requirements!

Proposals for NLOwPS's

- First working hadronic code (Z): Φ-veto (Dobbs, 2001)
- First correct general solution: MC@NLO (Frixione, Webber, 2002)
- Automated computations of ME's: grcNLO (GRACE group, 2003)
- Absence of negative weights (Nason, 2004; Frixione, Nason, Oleari, 2007) – POWHEG
- Showers with high log accuracy in ϕ_6^3 (Collins, Zu, 2002–2004)
- Proposals for $e^+e^- \to jets$ (Soper, Krämer, Nagy, 2003–2006)
- Within Soft Collinear Effective Theory (Bauer, Schwartz, 2006)
- Shower and matching with QCD antennae (Giele, Kosower, Skands 2007) – VINCIA
- With analytic showers (Bauer, Tackmann, Thaler, 2008) – GenEvA
- Together with MEC in e^+e^- (Lavesson, Lönnblad, 2008)

Some of these ideas have passed the crucial test of implementation. However, only two codes (MC@NLO and POWHEG) can be used to fully simulate a variety of hadronic processes

MC@NLO

- Compute what the MC does at the first non trivial order, and subtract it from the matrix elements. The resulting short-distance cross sections can be unweighted, and the hard events thus obtained are used as initial conditions for parton showers

- One set of analytical computations per MC
- Negative weights
- Strictly identical to MC in soft/collinear regions
- Strictly identical to NLO in hard emission regions;
  all $\mathcal{O}(\alpha_S^{2+b})$ terms not logarithmically enhanced are zero
- Inclusive cross sections identical to total cross section @NLO

POWHEG

- Replace the first MC emission with one generated with a p_T-ordered Sudakov, constructed by exponentiating the *full real matrix element*. Requires a truncated shower to restore the correct pattern of soft emissions for angular-ordered showers

- Short-distance computations independent of MCs
- No negative weights
- Differs from MC in soft/collinear regions if MC is not p_T-ordered. For angular-ordered showers, agreement with MC is restored by truncated showers (up to subleading terms)
- Differs from NLO in hard emission regions by $\mathcal{O}(\alpha_S^{2+b})$ terms; no piece of information on NNLO is used

MC@NLO vs POWHEG

The two approaches differ by terms of order higher than (N)LL+NLO (ie beyond nominal accuracy). These *may* not be small numerically (although they generally are)

MC@NLO vs POWHEG

The two approaches differ by terms of order higher than (N)LL+NLO (ie beyond nominal accuracy). These *may* not be small numerically (although they generally are)

- In MC@NLO, all $\mathcal{O}(\alpha_S^{2+b})$ terms and beyond not logarithmically enhanced (ie, non-MC) are set to zero. In POWHEG, one gets terms of this order, but these are spurious, since they are not obtained from an actual NNLO QCD calculation

  Neither code contains any information on non-logarithmic terms of $\mathcal{O}(\alpha_S^{2+b})$ ("NNLO") and beyond

MC@NLO vs POWHEG

The two approaches differ by terms of order higher than (N)LL+NLO (ie beyond nominal accuracy). These *may* not be small numerically (although they generally are)

- In MC@NLO, all $\mathcal{O}(\alpha_S^{2+b})$ terms and beyond not logarithmically enhanced (ie, non-MC) are set to zero. In POWHEG, one gets terms of this order, but these are spurious, since they are not obtained from an actual NNLO QCD calculation

  Neither code contains any information on non-logarithmic terms of $\mathcal{O}(\alpha_S^{2+b})$ ("NNLO") and beyond

- In MC@NLO the MC generates all non-hard emissions. This is not the case in POWHEG. Technically, this implies an ordering in p_T; thus, double-log accuracy is spoiled if an MC is used that is not ordered in p_T (such as HERWIG). It can be restored by adding a "soft" shower

  Soft showers are only available in HW++, but not in HW6

  Small effects on inclusive variables

Automation

MC@NLO

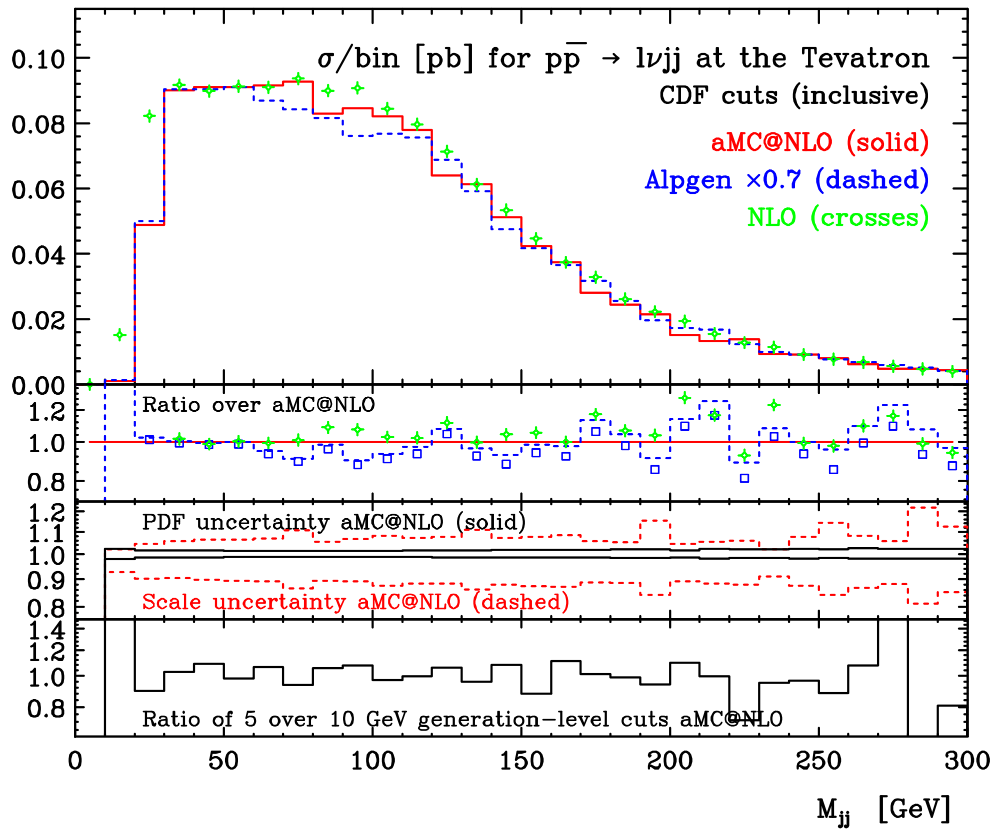
Fully automated in aMC@NLO, built upon MadGraph (trees), MadFKS (NLO subtraction), MadLoop (one-loop) (Frederix, Frixione, Torrielli; +Hirschi, Maltoni, Pittau). A unique framework for the whole business

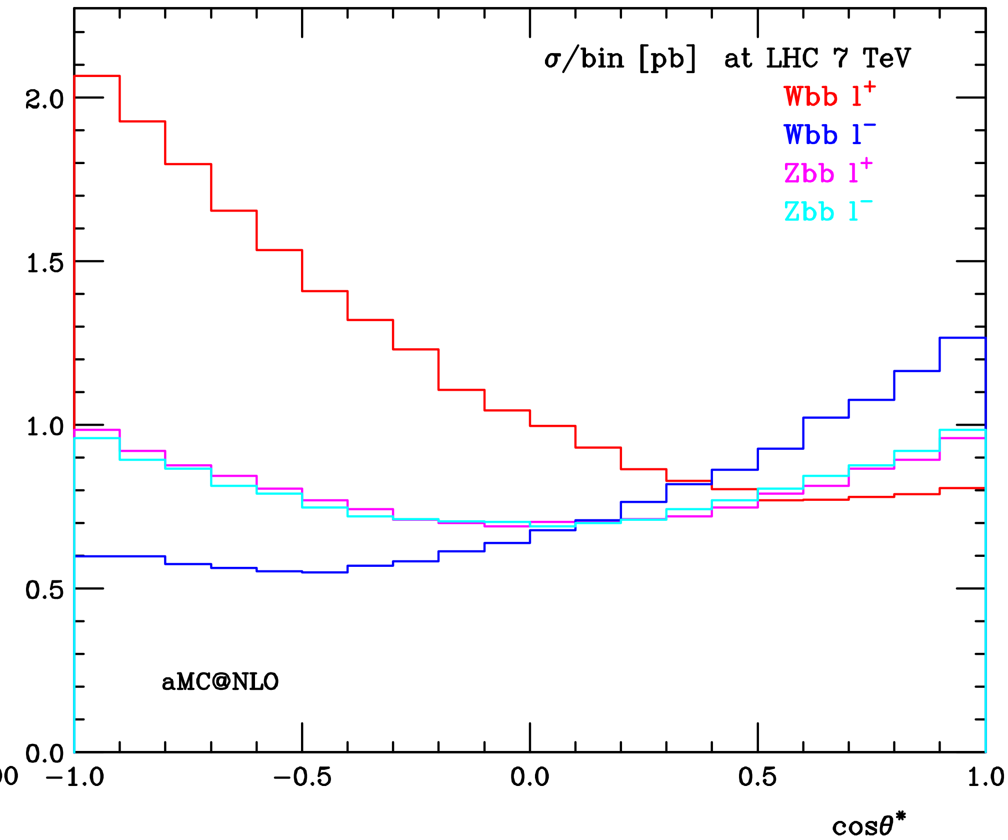
POWHEG

Automations in SHERPA (e.g. Hoeche, Krauss, Schonherr, Siegert, 1111.1220), where virtuals have to be provided externally, HELAC (e.g. Garzelli, Kardos, Papadopoulos, Trocsanyi, 1111.1444), based upon the POWHEG-Box framework (Alioli, Nason, Oleari, Re, 1002.2581)

I'm unable to comment on the present extent of automation/flexibility of the POWHEG implementations

Sample aMC@NLO applications





$$pp \longrightarrow (W \rightarrow)\ell\nu_\ell jj$$

$$pp \longrightarrow (Z/\gamma^* \rightarrow)\ell^+\ell^- b\bar{b} \Big/ (W \rightarrow)\ell\nu_\ell b\bar{b}$$

The only things we had to do: prepare input cards, and write the process-specific analysis routines

Conclusions and outlook

- The truly significant progress made recently has put NLO computations on the same footing as LO's – one just needs more CPU power
- Matching to shower is also automated – which gives one the possibility of performing experimental analyses completely with NLOwPS's

  (e.g. ATLAS is running aMC@NLO)

Conclusions and outlook

- The truly significant progress made recently has put NLO computations on the same footing as LO's – one just needs more CPU power
- Matching to shower is also automated – which gives one the possibility of performing experimental analyses completely with NLOwPS's
  (e.g. ATLAS is running aMC@NLO)

Future prospects

- Extend what was done in QCD to EW and BSM
- Construct faster programs

Conclusions and outlook

- The truly significant progress made recently has put NLO computations on the same footing as LO's – one just needs more CPU power
- Matching to shower is also automated – which gives one the possibility of performing experimental analyses completely with NLOwPS's

  (e.g. ATLAS is running aMC@NLO)

Future prospects

- Extend what was done in QCD to EW and BSM
- Construct faster programs

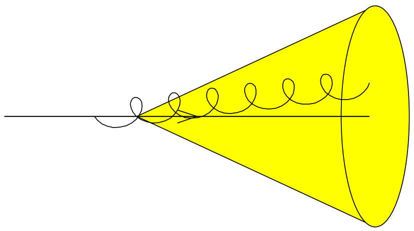
This is a very successful story, which however has had a limited impact on understanding NNLO computations. Why?

Extras

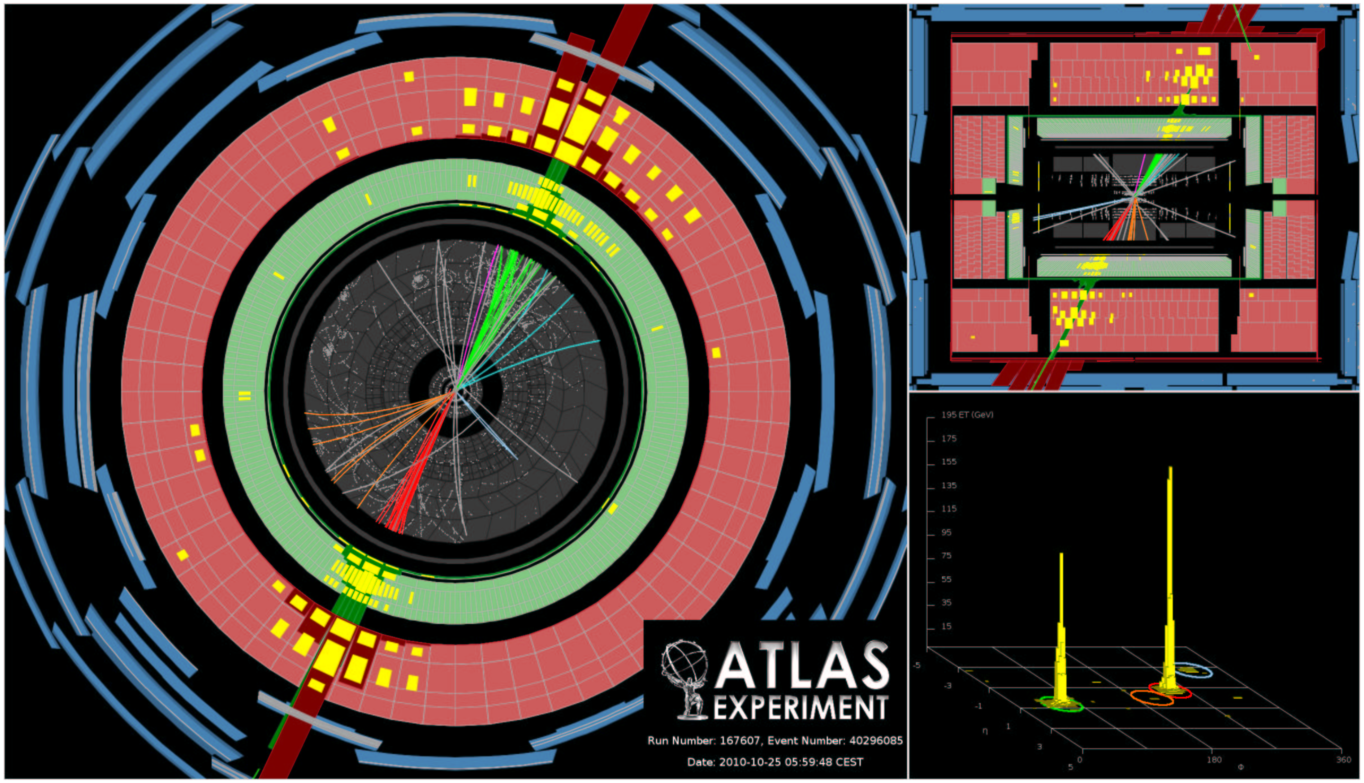
A: Better description of jet structure

(...in the sense that doing worse is difficult, perhaps?)

A jet in an NLO computation:



Actual jets:

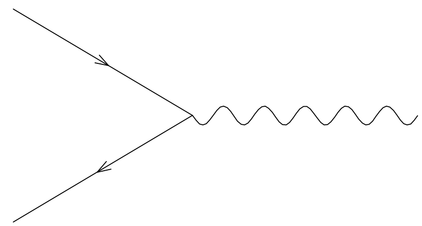


- One or two partons vs $\mathcal{O}(50)$ hadrons

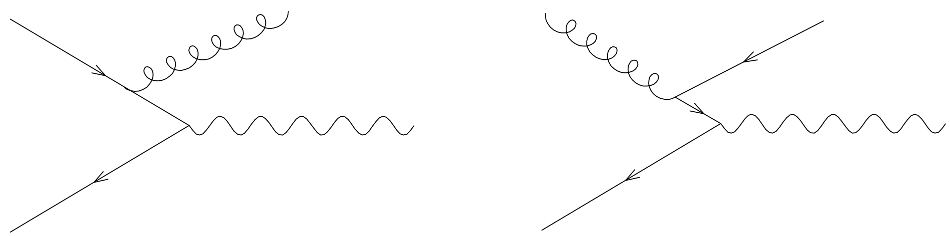
B: More combinations of initial-state partons

True, but misses the point. Consider Z production:

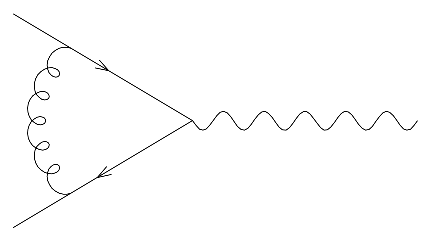
Born:



Real corrections:



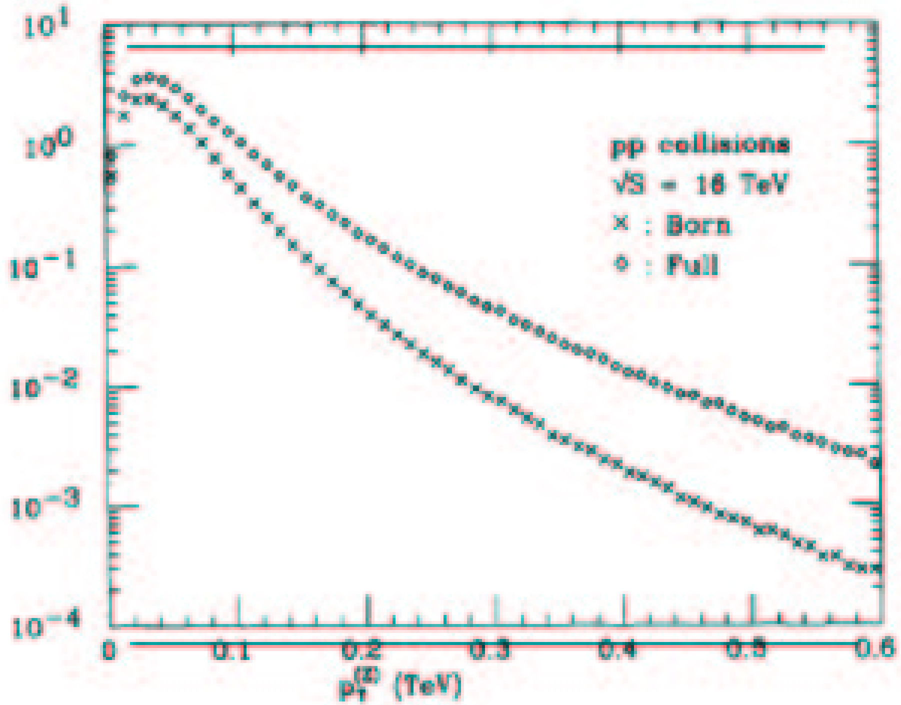
Virtual corrections:



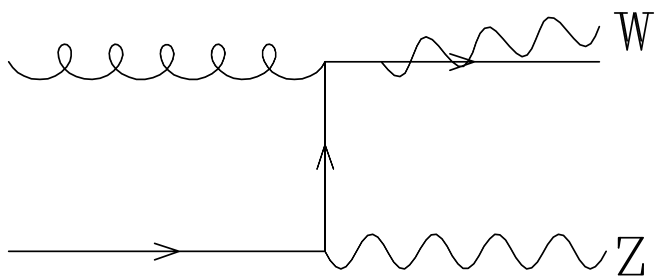
- This is a feature of tree-level corrections

C: NLO effects on distributions

Again, misses the point. Consider $p_T^{(Z)}$ in WZ production:



A K factor of about 6 at $p_T^{(Z)} = 600$ GeV. But:



is what dominates (double Sudakov log)

- Again, a feature of tree-level corrections

Construction of MC@NLO

The generating functional is:

$$\mathcal{F}_{\mathrm{MC@NLO}} = \mathcal{F}^{(2\to n+1)}\, d\sigma^{(\mathbb{H})}_{\mathrm{MC@NLO}} + \mathcal{F}^{(2\to n)}\, d\sigma^{(\mathbb{S})}_{\mathrm{MC@NLO}}$$

with the two *finite* short-distance cross sections

$$d\sigma^{(\mathbb{H})}_{\mathrm{MC@NLO}} = d\phi_{n+1}\left(\mathcal{M}^{(r)}(\phi_{n+1}) - \mathcal{M}^{(\mathrm{MC})}(\phi_{n+1})\right)$$

$$d\sigma^{(\mathbb{S})}_{\mathrm{MC@NLO}} = \int_{+1} d\phi_{n+1}\left(\mathcal{M}^{(b+v+rem)}(\phi_n) - \mathcal{M}^{(c.t.)}(\phi_{n+1}) + \mathcal{M}^{(\mathrm{MC})}(\phi_{n+1})\right)$$

- Black terms: pure NLO, same as before
- Red terms: MC subtraction terms, with a factorized form

$$\mathcal{M}^{(\mathrm{MC})} = \mathcal{K}^{(\mathrm{MC})}\mathcal{M}^{(b)}$$

Automation of MC@NLO $\equiv$ MadFKS+MadLoop+automation of $\mathcal{K}^{(\mathrm{MC})}$
$\equiv$ aMC@NLO (Frederix, Frixione, Torrielli)

Construction of POWHEG

Start with an exact phase-space factorization $d\phi_{n+1} = d\phi_n d\phi_r$, and construct

$$\overline{\mathcal{M}}^{(b)}(\phi_n) = \mathcal{M}^{(b+v+rem)}(\phi_n) + \int d\phi_r \left[\mathcal{M}^{(r)}(\phi_{n+1}) - \mathcal{M}^{(c.t.)}(\phi_{n+1})\right]$$

For a *given* p_T, define the *vetoed* *process-dependent* Sudakov

$$\Delta_R(t_I, t_0; p_T) = \exp\left[-\int_{t_0}^{t_I} d\phi'_r \frac{\mathcal{M}^{(r)}}{\mathcal{M}^{(b)}} \Theta(k_T(\phi'_r) - p_T)\right]$$

Obtain hard configurations (to be given to shower as initial conditions) from the short-distance cross section

$$d\sigma_{\text{POWHEG}} = d\phi_n \overline{\mathcal{M}}^{(b)}(\phi_n) \left[\Delta_R(t_I, t_0; 0) + \Delta_R(t_I, t_0; k_T(\phi_r)) \frac{\mathcal{M}^{(r)}(\phi_{n+1})}{\mathcal{M}^{(b)}(\phi_n)} d\phi_r\right]$$

which includes Sudakov suppression at $p_T \to 0$

- $k_T(\phi_r)$ will play the role of hardest emission
- The full real matrix element is exponentiated

Attaching (angular-ordered) showers

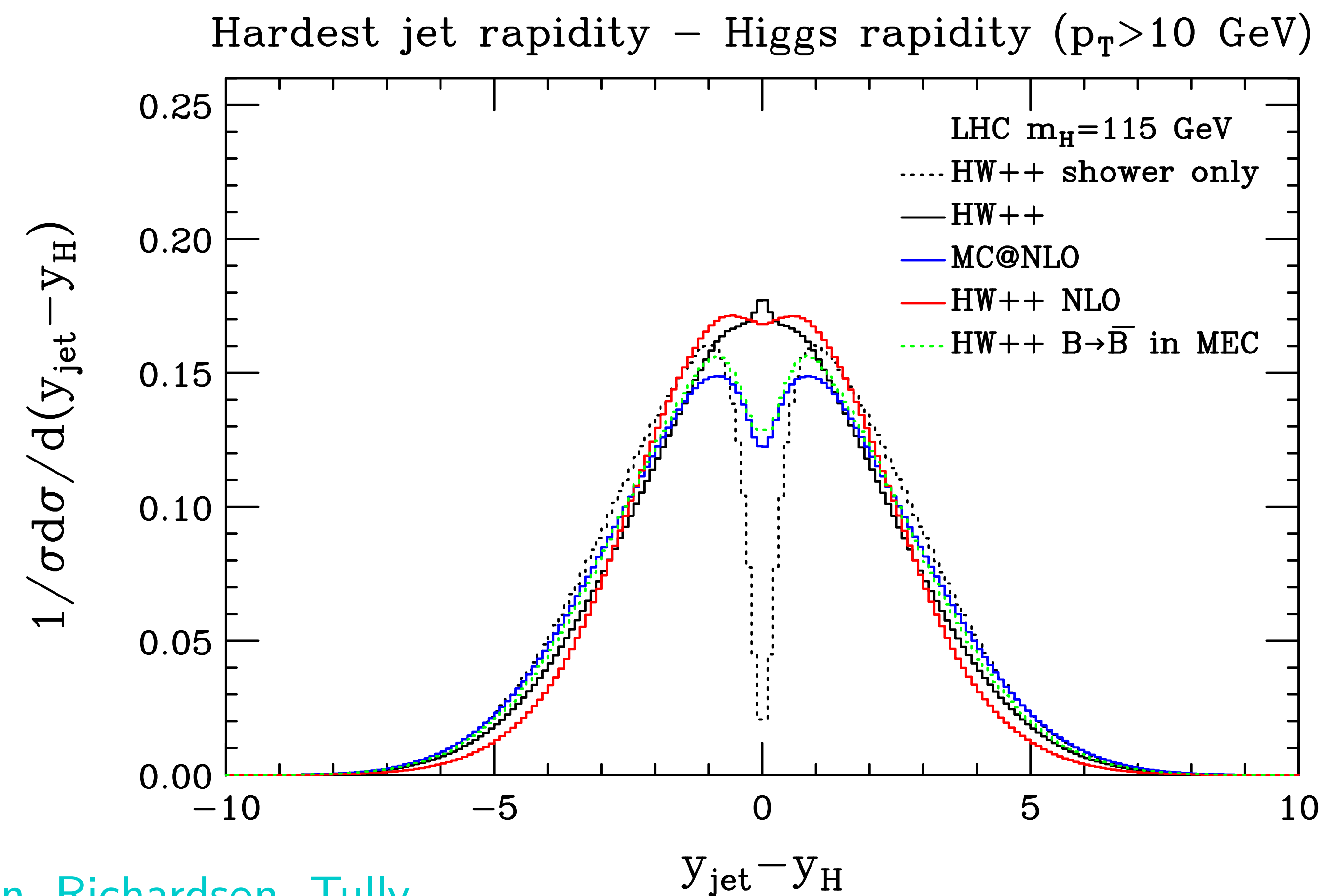
- One wants the matrix-element-generated p_T to be the hardest
  $\Longrightarrow$ veto emissions harder than p_T during shower
- But this screws up colour coherence

Colour coherence can be restored at the price of a more involved structure

$$\mathcal{F}_{\mathsf{POWHEG}}[t_I; p_T] \quad = \quad \Delta(t_I, t_0) + \int_{t_0}^{t_I} \frac{dt}{t} \int dz \Delta_R(t_I, t; p_T) \frac{\alpha_S}{2\pi} P(z) \\ \times \mathcal{F}_{\mathsf{v}}((1-z)^2 t; p_T)\, \mathcal{F}_{\mathsf{v}}(z^2 t; p_T)\, \mathcal{F}_{\mathsf{VT}}(t_I, t; p_T)$$

- $\mathcal{F}_{\mathsf{v}}(t; p_T)$ are *vetoed* showers. Evolve down to t_0, with all emissions constrained to have a transverse momentum smaller than p_T
- $\mathcal{F}_{\mathsf{VT}}(t_I, t; p_T)$ are *vetoed-truncated* showers. Evolve from t_I down to t (i.e., *not* t_0) along the hardest line. On top of that, they are vetoed

MC@NLO vs POWHEG: discrepancies



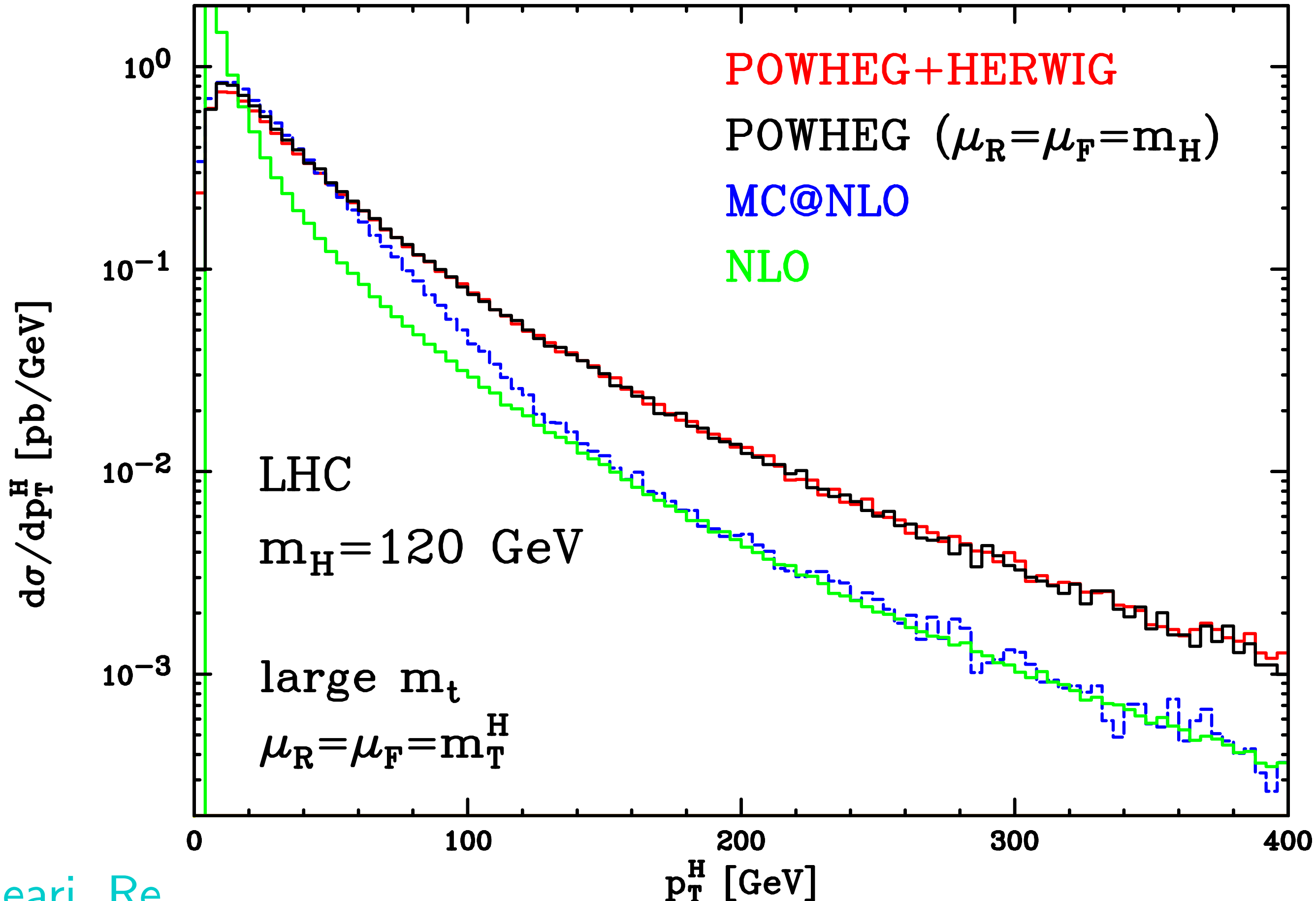
Hamilton, Richardson, Tully

HW/HW++ have dips at $\Delta y = 0$. Likely an artifact of dead zones

MC@NLO fills that dip, via hard radiation

POWHEG fills it much more, owing to extra $\mathcal{O}(\alpha_S^4)$ terms

MC@NLO vs POWHEG: discrepancies



Alioli, Nason, Oleari, Re

POWHEG a factor ~ 3 larger than MC@NLO$\equiv$ NLO in the tail

POWHEG result can be decreased by removing part of the real contribution from the exponent. Predictive power?

Note: MC@NLO and POWHEG use the *same matrix elements*