OpenLoops: Supplementing Monte Carlos with Loop Matrix Elements

Philipp Maierhöfer

IPPP Durham

PSR 2015: Parton Showers, Event Generators and Resummation

Kraków, 26 May 2015

In collaboration with

J. Lindert and S. Pozzorini,
S. Kallweit (MUNICH), M. Schönherr (Sherpa)

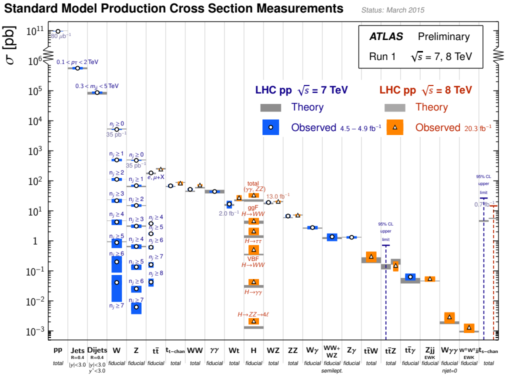
Outline

1. **NLO Automation**
   - Introduction
   - OpenLoops: algorithm
   - Usage
2. **Electroweak Corrections**
   - General considerations
   - $W + 1, 2, 3$ jets
3. **NNLO Applications: real-virtual corrections**
   - $pp \to W^+W^-$

NLO automation

Many accuracte measurements performed in LHC Run 1

- Consistent with Standard Model predictions.
- Run 2: Precision is likely to be the key to find BSM effects as small deviations from SM predictions.



- Many processes → automated tools required.
- Feasibility 2 → 4 NLO QCD well established.

Importance of NLO electroweak and NNLO QCD simulations becomes more and more apparent.

Shower Monte Carlos meet NLO

Before NLO matching: two different worlds

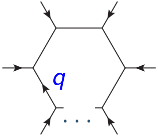
NLO calculations	Shower Monte Carlos
▪ mostly fixed order ▪ high precision cross sections for inclusive observables ▪ until a few years ago: man-years of work per process	▪ based on leading order matrix elements ▪ shapes and exclusive observables ▪ highly automated ▪ "close to experiments"

Ongoing: NLO QCD plus parton shower (and merging) is becoming the default accuracy for LHC simulations.

Upgrade existing tools to NLO, using loop matrix elements from a one-loop provider. Various more or less generic providers: `BlackHat`, `GoSam`, `HELAC-NLO`, `MadLoop`, `NJet`, `OpenLoops`, ...

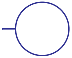
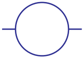
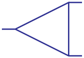
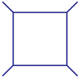
Loop amplitudes

Diagrammatic representation of a loop amplitude:
factorised colour, numerator $\mathcal{N}$, loop integral, momentum q.



$$= \mathcal{C} \cdot \int d^d q \, \frac{\mathcal{N}(q)}{D_0 \, D_1 \, \ldots \, D_{N-1}}$$

Must be reduced to scalar 1-, 2-, 3-, 4-point basis integrals:

$= \mathcal{C} \cdot \sum a_i$  $+ \sum b_i$  $+ \sum c_i$  $+ \sum d_i$ 

Reduction techniques/tools of our choice

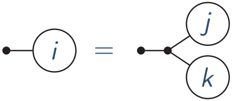
- OPP reduction: `CutTools` [Ossola, Papadopoulos, Pittau], scalar integrals from `OneLOop` [van Hameren]
- Tensor integral reduction: `Collier` [Denner, Dittmaier, Hofer]

Need to calculate the numerator function $\mathcal{N}(q)$.

From tree recursion to loop diagrams

Recursive construction of tree wave functions

Starting from external legs, connect wave functions w^α with vertices and propagators to recursively build "sub-trees". Wave functions of sub-trees are 4-tuples of complex numbers (for the spinor/Lorentz index).



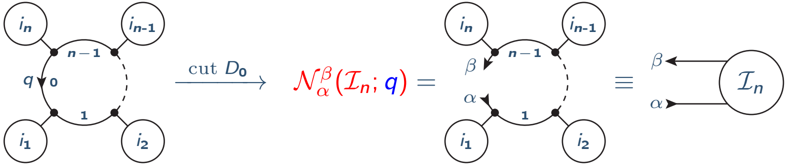
external lines are not depicted

$$w^\beta(i) = \frac{X^\beta_{\gamma\delta}}{p_i^2 - m_i^2}\, w^\gamma(j)\, w^\delta(k)$$

$X^\beta_{\gamma\delta}$ describes the interaction of i, j, k

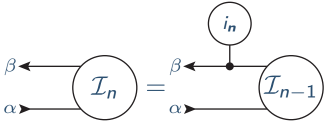
Loop diagrams

A one-loop diagram is an ordered set of sub-trees $\mathcal{I}_n = \{i_1, \ldots, i_n\}$, connected by loop propagators.



OpenLoops recursion

Connect sub-trees along the loop to build the numerator $\mathcal{N} \equiv \mathcal{N}_\alpha^\alpha$



$$\mathcal{N}_\alpha^\beta(\mathcal{I}_n; q) = X_{\gamma\delta}^\beta(q)\, \mathcal{N}_\alpha^\gamma(\mathcal{I}_{n-1}; q)\; w^\delta(i_n)$$

Naively, the numeric recursion only works by fixing the loop momentum to numeric values. Original idea behind OPP reduction: do this for multiple values of $q \rightarrow$ CPU expensive.

OpenLoops: regard $\mathcal{N}$ as a polynomial in q and formulate a recursion for the polynomial coefficients.

$$\mathcal{N}_\alpha^\beta(\mathcal{I}_n; q) = \sum_{r=0}^{n} \mathcal{N}_{\mu_1\ldots\mu_r;\alpha}^\beta(\mathcal{I}_n)\, q^{\mu_1} \ldots q^{\mu_r}, \qquad X_{\gamma\delta}^\beta = Y_{\gamma\delta}^\beta + q^\nu Z_{\nu;\gamma\delta}^\beta$$

$$\mathcal{N}_{\mu_1\ldots\mu_r;\alpha}^\beta(\mathcal{I}_n) = \left[Y_{\gamma\delta}^\beta\, \mathcal{N}_{\mu_1\ldots\mu_r;\alpha}^\gamma(\mathcal{I}_{n-1}) + Z_{\mu_1;\gamma\delta}^\beta\, \mathcal{N}_{\mu_2\ldots\mu_r;\alpha}^\gamma(\mathcal{I}_{n-1}) \right] w^\delta(i_n)$$

The "open-loops" coefficients $\mathcal{N}_{\mu_1\ldots\mu_r;\alpha}^\beta$ encode the **functional dependence** of the numerator on the loop momentum.

Remarks

- Inspired by a Dyson-Schwinger recursion for loop momentum polynomials [van Hameren '09], but diagrammatic.
- Numerical implementation requires only universal building blocks, derived from the Feynman rules of the theory.
- Naturally works with both, tensor integrals and OPP.
  OPP similarly efficient as tensor integrals thanks to
  - cheap multiple evaluations of the numerator for multiple values of the loop momentum.
  - interference with Born amplitude, colour and helicity summation before reduction to scalar integrals.
- By 1-2 orders of magnitudes faster than original OPP based algorithms. → Performance breakthrough.
- Rational contributions of type R_2 are restored by counterterm-like Feynman rules. [Draggiotis, Garzelli, Malamos, Papadopoulos, Pittau '09, '10; Shao, Zhang, Chao '11]

Performance

process	diags	size/MB	time/ms
$u\bar{u} \to t\bar{t}$	11	0.1	0.27(0.16)
$u\bar{u} \to W^+W^-$	12	0.1	0.14
$u\bar{d} \to W^+g$	11	0.1	0.24
$u\bar{u} \to Zg$	34		0.75
$gg \to t\bar{t}$	44	0.2	1.6(0.7)
$u\bar{u} \to t\bar{t}g$	114	0.4	4.8(2.4)
$u\bar{u} \to W^+W^-g$	198	0.4	3.4
$u\bar{d} \to W^+gg$	144	0.5	4.0
$u\bar{u} \to Zgg$	408		17
$gg \to t\bar{t}g$	585	1.2	40(14)
$u\bar{u} \to t\bar{t}gg$	1507	3.6	134(101)
$u\bar{u} \to W^+W^-gg$	2129	2.5	89
$u\bar{d} \to W^+ggg$	1935	4.2	120
$u\bar{u} \to Zggg$	5274		524
$gg \to t\bar{t}gg$	8739	16	1460(530)

i7-3770K (single thread), gfortran 4.8, dynamic (ifort static ~30% faster), tensor integral reduction with `Collier`.

Colour and helicity summed, heavy quark loops included.

W/Z production includes leptonic decays and non-resonant contributions.

$t\bar{t}$ production numbers in brackets are for massless decays.

CutTools provides similar performance for $2 \to 4$, but is slower for lower multiplicities.

Installation

OpenLoops is available for download from hepforge

`http://openloops.hepforge.org`

Requirements: `gfortran` 4.6 or later, `Python` `2.x` ($x \geq 4$)

The easiest way to install and keep your installation up to date is to pull a copy from the subversion repository (there's also a tarball available on the website)

```
svn checkout \
  http://openloops.hepforge.org/svn/OpenLoops/branches/public \
  OpenLoops cd OpenLoops
```

go to the website and look up the processes which are available for download and install the ones you need, e.g. for Z + up to three jets:

```
./openloops libinstall ppzj ppzjj ppzjjj
```

Or to install everything: `./openloops libinstall all.coll`

To update OpenLoops and installed processes: `./openloops update`

Use OpenLoops in your own programs

OpenLoops can be used via

- a Binoth Les Houches Accord (BLHA) interface
- its native interface in `Fortran` and `C`

Provides tree level matrix elements for Born and real corrections, including spin and colour correlations, 1-loop and (1-loop)^2.
For NLO simulations these must be supplemented by phase space integration and a subtraction procedure, possibly a parton shower, multi-jet merging, hadronisation, ...

$\mathcal{O}(100)$ processes available: almost the full range of $2 \to 4$ (plus decays) Standard Model processes with QCD corrections.

Physics models supported (processes will be added step-by-step)

- Full Standard Model with QCD and electroweak corrections
- Higgs Effective field theory
- Two Higgs Doublet Model Type I and II

Code example

```
program main
  use openloops
  implicit none
  integer :: id
  real(8) :: psp(0:3,5), m2l0, m2l1(0:2), acc, sqrt_s = 1000
  call set_parameter("order_ew", 1) ! coupling order
  call set_parameter("mass(23)", 91.2)
  id = register_process("1 -1 -> 23 2 -2", 11) ! dd~ -> Zuu~
  call start() ! initialise
  call set_parameter("alpha_s", 0.1)
  call set_parameter("mu", 100)
  call phase_space_point(id, sqrt_s, psp) ! phase-space pt. from Rambo
  call evaluate_loop(id, psp, m2l0, m2l1, acc) ! matrix element
  print *, "tree, loop ep^0,-1,-2, accuracy:", m2l0, m2l1, acc
  call finish()
end program main
```

Interfaces to Monte Carlo programs

- **Sherpa**
  Interface included in the official release 2.1.0 (or later).
  Full-featured event generator, `Sherpa+OpenLoops` is well tested and used in several published simulations.
- **Herwig++**: Official support in the next Herwig++ release.
- **POWHEG**: Coming soon.
- **MUNICH**: Parton-level Monte Carlo by S. Kallweit
  (to be published). Particularly used in NNLO calculations.

Applications

NLO: focus on applicability to experiments, "beyond fixed order"

- MEPS@NLO for $\ell\ell\nu\nu + 0, 1\,\mathrm{jets}$ [Cascioli, Höche, Krauss, PM, Pozzorini, Siegert]
- (loop)^2 merging for $HH + 0, 1\,\mathrm{jets}$ [PM, Papaefstathiou]
- NLO $W^+W^-b\bar{b}$ with $m_b > 0$ [Cascioli, Kallweit, PM, Pozzorini]
- MEPS@NLO $W^+W^-W^\pm + 0, 1\,\mathrm{jets}$ [Höche, Krauss, Pozzorini, Schönherr, Thompson, Zapp]
- S-MC@NLO for $t\bar{t}b\bar{b}$ with $m_b > 0$ [Cascioli, PM, Moretti, Pozzorini, Siegert]
- MEPS@NLO for $t\bar{t} + 0, 1, 2\,\mathrm{j}$ [Höche, Krauss, PM, Pozzorini, Schönherr, Siegert]

NNLO: for real-virtual corrections, partly for (1-loop)^2

- $Z\gamma$ [Grazzini, Kallweit, Rathlev, Torre]
- ZZ [Cascioli,Gehrmann,Grazzini,Kallweit,PM,v. Manteuffel,Pozzorini,Rathlev,Tancredi,Weihs]
- W^+W^- [Gehrmann,Grazzini,Kallweit,PM,v. Manteuffel,Pozzorini,Rathlev,Tancredi]
- $t\bar{t}$ [Abelof, Gehrmann-De Ridder, PM, Pozzorini]

NLO-EW: $W + 1, 2, 3\,\mathrm{jets}$ [Kallweit Lindert, PM, Pozzorini, Schönherr]

Electroweak Sudakov Effects

Naively, or more precisely at small momentum transfer $Q \lesssim M_W$, electroweak corrections are small compared to QCD corrections: $\alpha \sim \alpha_s^2$

But: Electroweak corrections can be strongly enhanced by log $\frac{Q^2}{M_W^2}$.

Large logarithms arise from weak boson exchange between on-shell legs.

Estimate for 1-loop leading log (LL) and next-to-leading log (NLL) at $Q = 1\,\text{TeV}$:

$$\text{LL} \simeq -\frac{\alpha}{\pi s_W^2} \log^2 \frac{Q^2}{M_W^2} \simeq -26\%$$

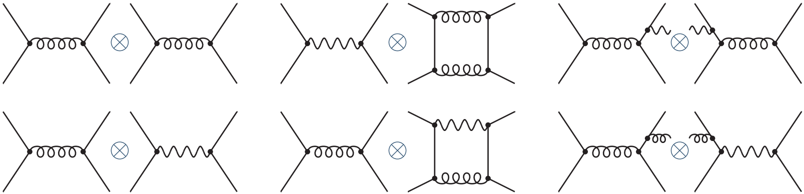
$$\text{NLL} \simeq +\frac{3\alpha}{\pi s_W^2} \log \frac{Q^2}{M_W^2} \simeq +16\%$$

- $\mathcal{O}(10\%)$ effect at 1 TeV.
- Could hide BSM effects in tails of distributions.
- Strongly process dependent.

Need electroweak NLO corrections

Power Counting

The notion of QCD and EW corrections is not well defined.



- EW correction to "QCD Born" or QCD correction to "EW Born"?
- For individual diagrams, the correction type can be defined via the origin of the UV poles (if any), but not for a full process.
- Real corrections involve photon and QCD parton emissions.
- For a consistent definition of the perturbative order and for infra-red finiteness:

Take into account comlpete $\mathcal{O}(\alpha^n \alpha_s^m)$ contributions.

NLO EW Automation

Things to consider

- Electroweak infra-red subtraction: photon emission. Heavy boson emission: only partial cancellation between real and virtual. Usually best treated as separate processes.
- Jet/Photon definition: collinear $q \to q\gamma$ splitting.
- Matching to QED shower. Merging?

Performance (wrt. QCD corrections)

process	diagrams		size [MB]		time [ms/point]	
	QCD	EW	QCD	EW	QCD	EW
$d\bar{d} \to t\bar{t}$	11	33	0.1	0.2	0.27	0.69
$gg \to t\bar{t}$	44	70	0.2	0.3	1.6	2.8
$d\bar{d} \to t\bar{t}g$	114	360	0.4	0.9	4.8	13
$gg \to t\bar{t}g$	585	660	1.4	1.6	40	56
$d\bar{d} \to t\bar{t}u\bar{u}$	236	1274	0.8	2.8	12	48
$gg \to t\bar{t}gg$	8739	7614	18	16	1458	1557

CPU: i7-3770K, compiler: gfortran 4.8

EW corrections are moderately slower than QCD corrections.

NLO EW $W^+ + 1, 2, 3$ jets

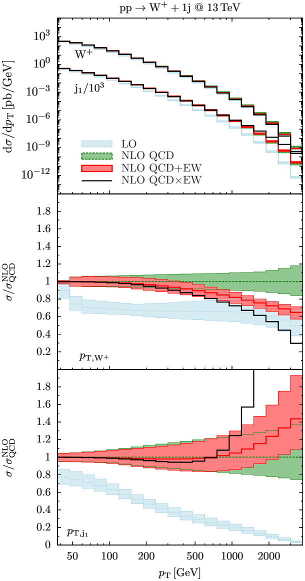
W+jets: Testing ground for theory and tools in multi-jet cases.

- NLO QCD available up to $W + 5j$ [Blackhat '09,'11,'13]
- $\mathcal{O}(< 10\%)$ QCD scale uncertainties.
- NLO EW $W + 1j$: $\mathcal{O}(-50\%)$ EW Sudakov effects at TeV scale [Kühn, Kulesza, Pozzorini, Schulze '04-'07; Denner, Dittmaier, Kasprzik, Mück '09-'11]

Here: NLO EW $W^+ + 1, 2, 3$j

- On-shell external W bosons: $\Gamma_W = 0$
- Regulate s channel propagators of heavy particles with a small width $\Gamma_{\mathrm{reg}} \to 0$.

W^++j inclusive



Inclusive

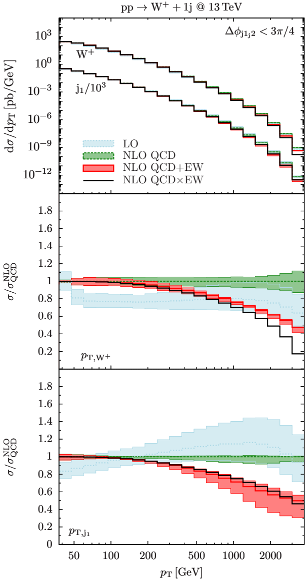
- -30% EW corr. wrt. NLO QCD at 2 TeV.
- Large QCD corrections, arising from (back-to-back) dijet configurations.

  → effectively LO.

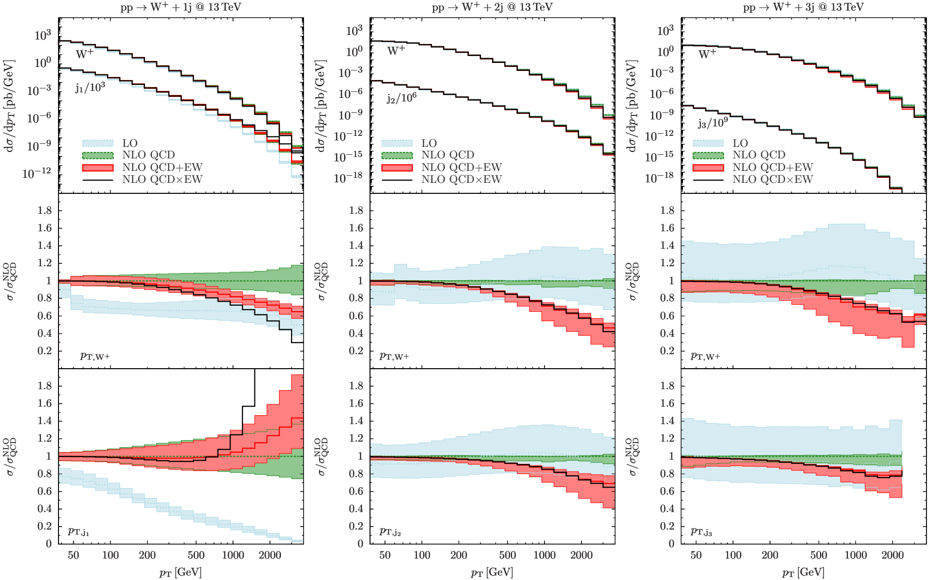
Exclusive

- -40% EW corr. wrt. NLO QCD at 2 TeV.
- Moderate QCD corrections.
- EW corrections exceed QCD uncertainties above $p_{T,W} > 300\ GeV$

W^++j, $\Delta\phi_{j1,j2} < 3\pi/4$



W+1j **W+2j** **W+3j**


pp → W⁺ + 1j @ 13 TeV
pp → W⁺ + 2j @ 13 TeV
pp → W⁺ + 3j @ 13 TeV
W⁺
j₁/10³
j₂/10⁶
j₃/10⁹
LO
NLO QCD
NLO QCD+EW
NLO QCD×EW
dσ/dp_T [pb/GeV]
σ/σ_QCD^NLO
p_T,W+
p_T,j₁
p_T,j₂
p_T,j₃
p_T [GeV]

NNLO $pp \to W^+W^-$ [Gehrmann, Grazzini, Kallweit, PM, v. Manteuffel, Pozzorini, Rathlev, Tancredi]

W^+W^- crosss section: 20% excess (2σ) wrt. NLO+gg theory.

- underestimated uncertainties?

Ingredientes to on-shell W^+W^- production

- 2-loop master integrals [Gehrmann, Tancredi, v. Manteuffel, Weihs; Caola, Henn, Melnikov, Smirnov] and amplitudes [Gehrmann, Tancredi, v. Manteuffel]
- 1-loop amplitudes from OpenLoops
- NNLO q_T subtraction [Catani, Grazzini; Kallweit, Rathlev]

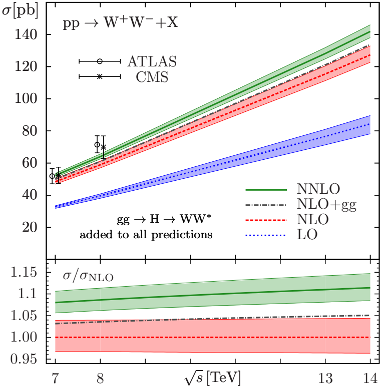
The real-virtual subtraction is particularly challenging, because one must integrate over phase space regions where the gluon is soft or colliner with a massless quark $\to$ 1-loop matrix elements become numerically unstable.

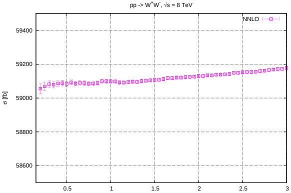
Matrix elements and subtraction terms are both divergent.
$\to$ High precision required to have precision left after subtraction.

Works with the "vanilla" OpenLoops stability trigger
with Collier and CutTools (could be tuned further).

NNLO $pp \to W^+W^-$: Results

- NNLO corrections between 9% (7 TeV) and 12% (14 TeV).
- Only 35% of the NNLO correction arises from $gg \to W^+W^-$.
- 3% uncertainty estimated from scale variations.
  No significant reduction wrt. NLO.
  More reliable, because at NNLO all channels are open.
- Eases tension with ATLAS/CMS 8 TeV data.





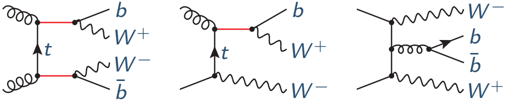
Dependence of the cross section on $q_T/m_{WW} \to 0$

- Stable at 1%.
- Increasing uncertainties from real-virtual and double real to similar amounts.

Subtraction of Top Quark Contributions

Huge top contamination due to

$pp \to Wt \to W^+W^-b$ (30-60% at NLO for 7-14 TeV) and
$pp \to t\bar{t} \to W^+W^-b\bar{b}$ (400%-800% at NNLO).



"Top-free" definition of W^+W^-

- Experimental: suppress top contribution with jet vetos and subtract remnant top background.
- 4-flavour scheme: massive b-quarks, excluding b-quark emission
- 5-flavour scheme: massless b-quark emissions needed to cancel collinear $g \to b\bar{b}$ singularity.

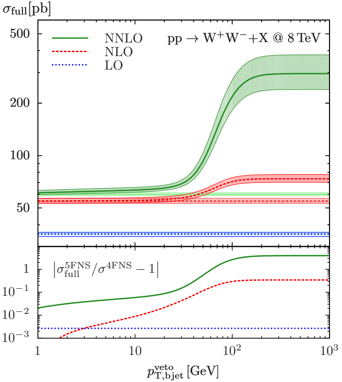
Subtract top contribution by exploiting $\xi_t = \Gamma_t/\Gamma_t^{phys}$ scaling.

Numerical fit $\sigma^{5F} = \xi_t^{-2}\sigma_{t\bar{t}}^{5F} + \xi_t^{-1}\sigma_{Wt}^{5F} + \sigma_{W^+W^-}^{5F}$

Assess theory ambiguities by comparing 4F and top subtracted 5F.

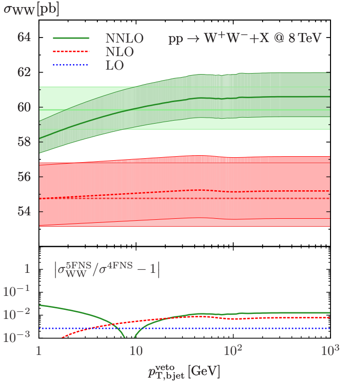
Top Subtraction and b-jet Veto

b-jet veto: cross section for $p_T < p_{T,\text{b-jet}}^{\text{veto}}$



Full 5F cross section vs. 4F

Top contamination still 10% at $p_{T,\text{b-jet}}^{\text{veto}} = 10\,\text{GeV}$.
No plateau, due to singularity.



Top-free 5F cross section vs. 4F

Stable top subtraction for $p_{T,\text{b-jet}}^{\text{veto}} = 10\,\text{GeV}$.
1% agreement between 4F and 5F.

Conclusions

OpenLoops makes using 1-loop matrix elements easy

- provides all matrix elements which are needed for NLO
- wide range of processes available, continuously extended
- very fast
- numerically stable, even for NNLO real-virtuals
- Standard Model, Higgs Effective Theory, soon 2HDM

Electroweak corrections

- NLO EW $2 \to 4$ at moderatey higher CPU cost wrt. QCD
- Large corrections due to EW Sudakov logs in the TeV region
- $W + 1, 2, 3j$: multi-jet significantly different from $W + 1j$

http://openloops.hepforge.org