Present and future of the OpenLoops matrix element generator

Philipp Maierhöfer

IPPP Durham

Radcor-Loopfest
Los Angeles, 17 June 2015

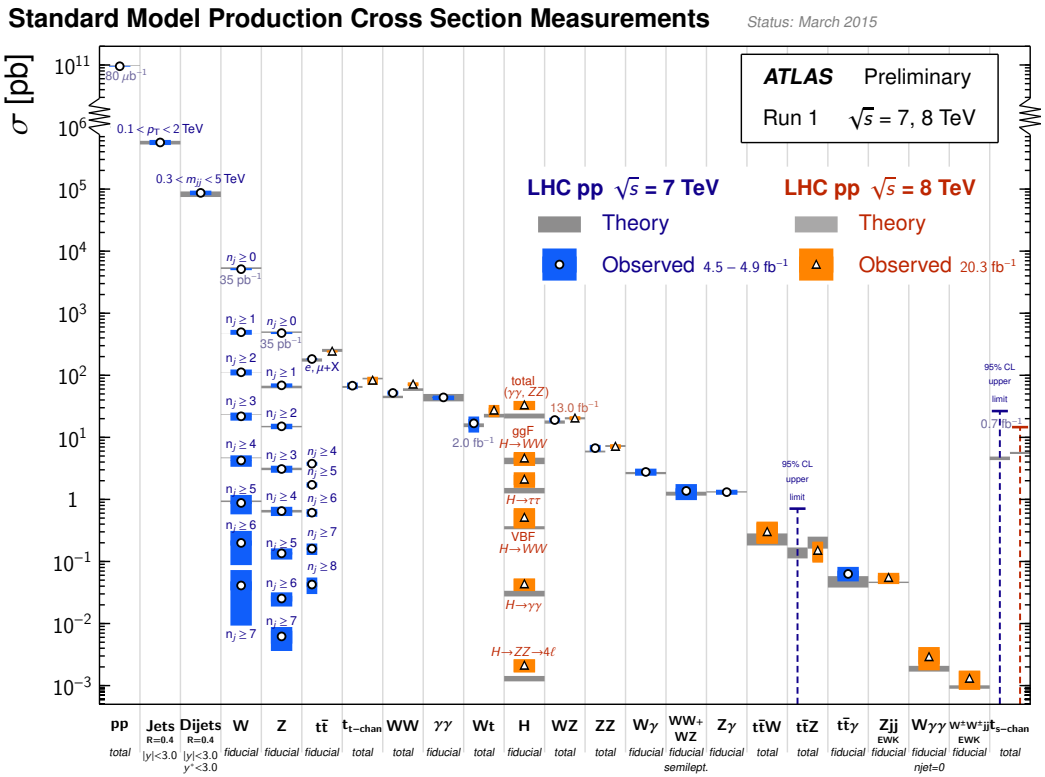
In collaboration with
J. Lindert and S. Pozzorini

Outline

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
  $\rightarrow$ require automation



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

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

Large electroweak corrections

Electroweak Sudakov Effects

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

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

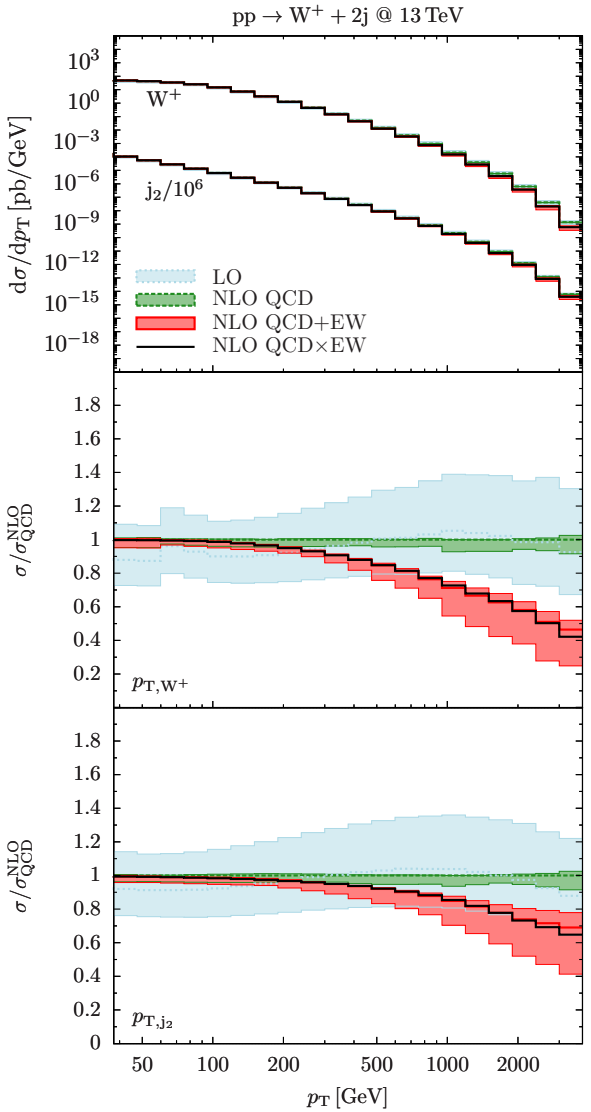
$$\text{LL} \simeq -\frac{\alpha}{\pi s_W^2} \log^2 \frac{Q^2}{M_W^2} \simeq -26\%$$

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

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

See talk by Jonas Lindert ⟵

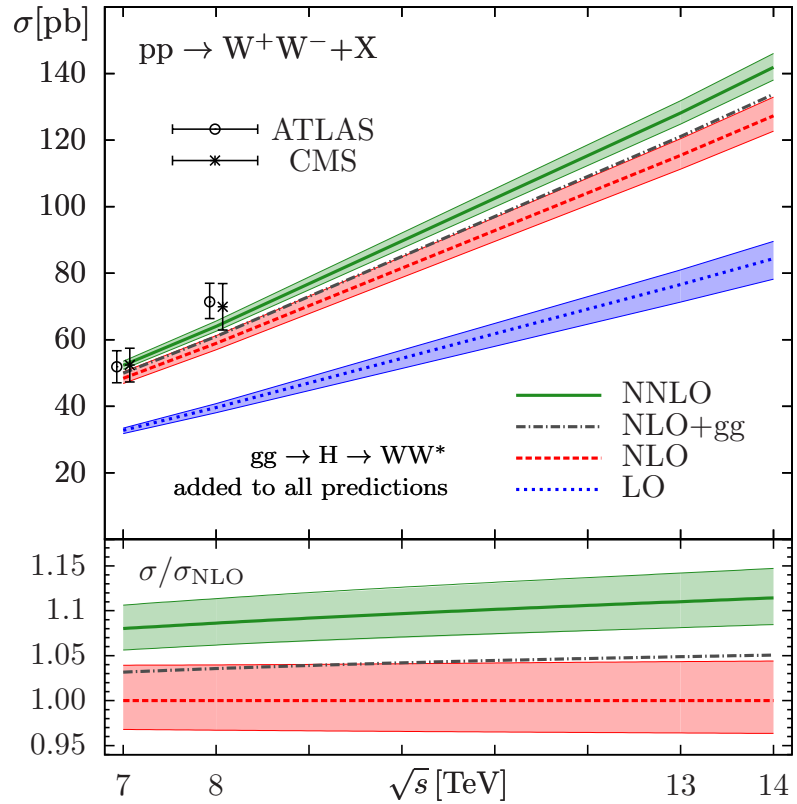
W+2j EW NLO [Kallweit, Lindert, PM, Pozzorini, Schönherr]



Large NNLO QCD corrections

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

- NNLO corrections between 9% (7 TeV) and 12% (14 TeV).
- 3% uncertainty estimated from scale variations. No significant reduction wrt. NLO.
- LO, NLO and NNLO uncertainty bands do not overlap. NNLO error considered more reliable, because all channels are open.
- Eases tension with ATLAS/CMS 8 TeV data.



[Gehrmann, Grazzini, Kallweit, PM, v. Manteuffel, Pozzorini, Rathlev, Tancredi]

see talk by Dirk Rathlev

Loop provider applications

- Back-end for for loop matrix elements in Monte Carlo frameworks (for experiments).
- Easy interfacing for people who work on Monte Carlos/ showers/ resummation or need a real-virtual and loop^2 provider for NNLO.

If you need 1-loop matrix elements, don't reinvent the wheel.
Most likely a loop generator will do the job for you.

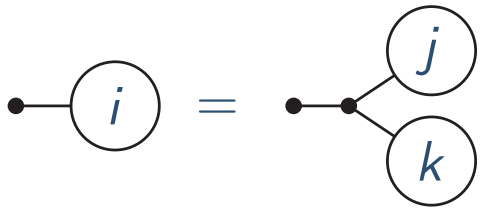
One-loop providers: `BlackHat`, `GoSam`, `HELAC-NLO`, `MadLoop`, `NJet`, `OpenLoops`, `Recola`, ...

- Not all programs are general purpose generators (are the processes you need available?).
- Not all programs are publically available.
- Is it fast enough (high multiplicity)?
- Is it numerically stable enough (esp. for NNLO)?

From tree recursion to loop diagrams

Recursive construction of tree Feynman diagrams

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).



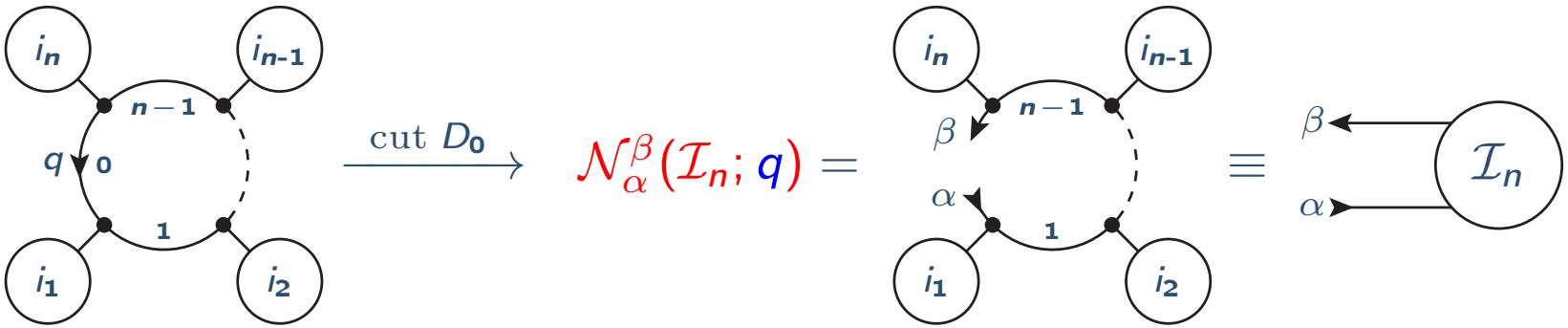
$$w^\beta(i) = \frac{X^\beta_{\gamma\delta}}{p_i^2 - m_i^2}\, w^\gamma(j)\, w^\delta(k)$$

external lines are not depicted

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

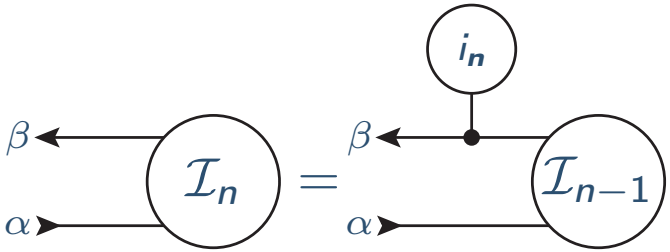
Loop diagrams

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 $\mathcal{N} \equiv \mathcal{N}^{\alpha}_{\alpha}$



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

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 $q \to$ very CPU expensive.

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

$$\mathcal{N}^{\beta}_{\alpha}(\mathcal{I}_n; q) = \sum_{r=0}^{n} \mathcal{N}^{\beta}_{\mu_1 \ldots \mu_r;\alpha}(\mathcal{I}_n)\, q^{\mu_1} \ldots q^{\mu_r}, \quad X^{\beta}_{\gamma\delta} = Y^{\beta}_{\gamma\delta} + q^{\nu} Z^{\beta}_{\nu;\gamma\delta}$$

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

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

Remarks

- Uses FeynArts to generate Feynman diagrams. [Hahn]
- Numerical implementation requires only universal building blocks, derived from the Feynman rules of the theory.
- Naturally works with both, tensor integrals and OPP reduction.
  - tensor integral reduction: `Collier` [Denner, Dittmaier, Hofer]
    see talk by Lars Hofer
  - OPP reduction [Ossola, Papadopoulos, Pittau]: `CutTools` [idem], with scalar integrals from `OneLOop` [van Hameren]
- By 1-2 orders of magnitudes faster than original OPP based algorithms, thanks to cheap multiple evaluations of the numerator function. $\rightarrow$ Performance breakthrough.
- 4-dimensional: 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	time/ms
$u\bar{u} \to t\bar{t}$	11	0.27(0.16)
$u\bar{u} \to W^+W^-$	12	0.14
$u\bar{d} \to W^+g$	11	0.24
$u\bar{u} \to Zg$	34	0.75
$gg \to t\bar{t}$	44	1.6(0.7)
$u\bar{u} \to t\bar{t}g$	114	4.8(2.4)
$u\bar{u} \to W^+W^-g$	198	3.4
$u\bar{d} \to W^+gg$	144	4.0
$u\bar{u} \to Zgg$	408	17
$gg \to t\bar{t}g$	585	40(14)
$u\bar{u} \to t\bar{t}gg$	1507	134(101)
$u\bar{u} \to W^+W^-gg$	2129	89
$u\bar{d} \to W^+ggg$	1935	120
$u\bar{u} \to Zggg$	5274	524
$gg \to t\bar{t}gg$	8739	1460(530)

i7-3770K (single thread), gfortran 4.8

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.

Colour and helicity summation

Naively, one first calculates amplitudes, which are then squared.

$$\sum_{\text{hel}}\sum_{\text{col}} \text{tree} \otimes \left[\sum_{\text{topo}} \int dq \frac{\mathcal{N}^{\text{topo}}_{\text{hel}}(q)}{D_0 \dots D_{n-1}}\right]$$

$\rightarrow$ one loop integral per colour structure $\times$ helicity config. $\times$ topology

In order to minimise OPP reduction calls, colour and helicity summation is performed after interference with the Born amplitude.

$$\sum_{\text{topo}} \int dq \left[\sum_{\text{hel}}\sum_{\text{col}} \text{tree} \otimes \frac{\mathcal{N}^{\text{topo}}_{\text{hel}}(q)}{D_0 \dots D_{n-1}}\right]$$

$\rightarrow$ one loop integral per topology

Calculating the numerator function $\mathcal{N}$ for a single helicity configuration is much faster than the reduction and the scalar integrals.

$\rightarrow$ **full helicity summation typically costs only a factor 2 CPU time.**

Squared loop amplitudes

For squared loop amplitudes,
"summation after interference" doesn't work.

→ OPP reduction becomes very inefficient.

But tensor integrals can be reused for every colour and helicity structure.

With tensor integral reduction, evaluating squared loop amplitudes is as fast as calculating NLO corrections.

While NLO corrections work very well with `CutTools`,
`Collier` is needed for loop induced amplitudes.

Also works for squared loop amplitudes for NNLO calculations.
Poles up to ϵ^{-4} can be calculated for cancellation checks.

A similar argument holds for colour correlated loop matrix elements.
→ Relevant for NNLO subtraction with coloured final states.
see talk be Michał Czakon

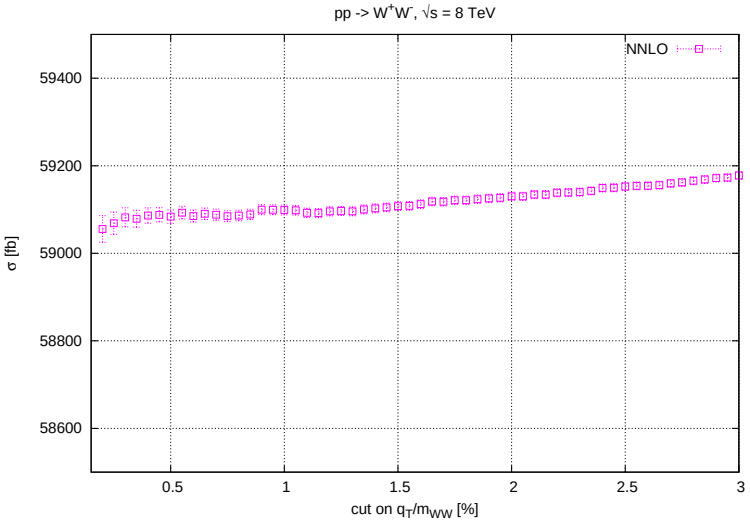
Real-virtual corrections: W^+W^-

The real-virtual contribution to an NNLO calculation is particularly challenging, because one must integrate over phase space regions with a soft and/or collinear particle
$\rightarrow$ 1-loop matrix elements become numerically unstable.

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

NNLO W^+W^- with q_T subtraction

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



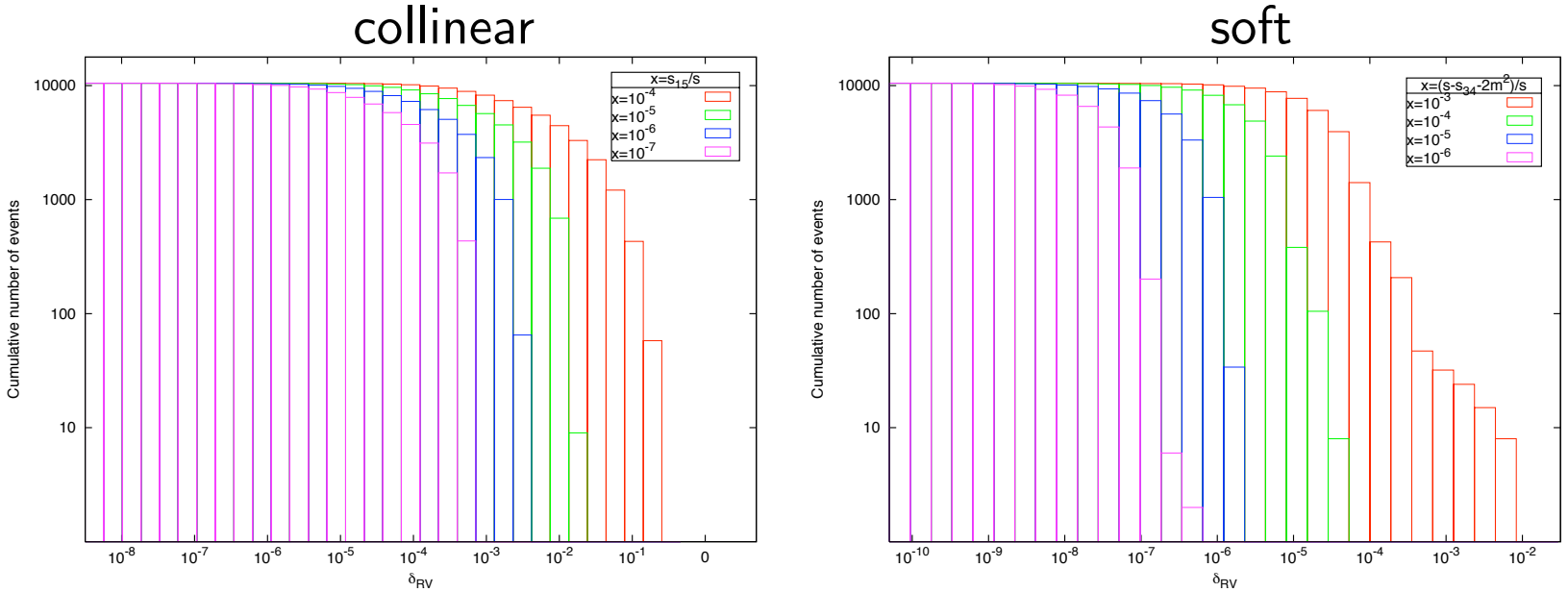
- Stable at 1%.
- Increasing uncertainties from real-virtual and double real to similar amounts.
- Works out of the box with the OpenLoops stability trigger with Collier and CutTools (could be tuned further).

Real-virtual corrections: $t\bar{t}$

$q\bar{q} \to t\bar{t}$ at leading colour (all channels: see talk by Gabriel Abelof)

Use a stability trigger system which is taylored to the problem.

- Estimate the numerical precision of the matrix elements and
- compare it to the size of the cancellation with the subtraction term.
- Repeat in quad precision if there is little precision left.



Cancellation for 10000 phase space points with `CutTools`

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, ...

$>$ 100 processes available: almost the full range of $2 \to 4$
(plus leptonic decays of vector bosons, off-shell and non-resonant)
Standard Model processes with QCD corrections.

- Continuously extended.
- Need something else? Tell us!

Use OpenLoops in your own programs

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

- Full Standard Model with QCD and electroweak corrections
- Higgs effective field theory (Higgs plus up to three jets available)
- Two Higgs Doublet Model Type I and II
- Approximations for special applications: arbitrary diagram selection with FeynArts and signal/ background/ interference selection.
- Performance and stability good enough for $2 \to 5$ processes.

The process generator will be included in a future release.

Support for `FeynRules` [Alloul, Christensen, Duhr, Degrande, Fuks] models is in development.

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
  id = register_process("1 -1 -> 23 2 -2", 11) ! dd~ -> Zuu~
  call start() ! initialise
  call set_parameter("mass(23)", 91.2)
  call set_parameter("alpha_s", 0.1)
  call set_parameter("mu", 100)
  call phase_space_point(id, sqrt_s, psp) ! rand. phase-space pt.
  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 upcoming Herwig++ release.
  see talk by Christian Reuschle
- **WHIZARD** see talk by Jürgen Reuter
- **POWHEG**: Coming soon.
- **MUNICH**: Parton-level Monte Carlo by Stefan Kallweit
  (to be published). Particularly used in NNLO calculations.

Conclusions

OpenLoops makes using 1-loop matrix elements easy

- Provides all matrix elements which are needed for NLO, loop induced, and loop squared for NNLO.
- Wide range of processes available, continuously extended. $2 \to 4$ standard, $2 \to 5$ feasible
- Very fast, numerically stable, suitable for NNLO real-virtuals
- Interfaces to several Monte Carlo Tools; in particular: ready to use and tested in real life with Sherpa

Going beyond Standard Model NLO QCD

- Electroweak corrections to the full Standard Model implemented. first $2 \to 4$ NLO EW application: $W + 1, 2, 3$ jets
- Higgs Effective Theory: $H + 3$ jets available
- First BSM model implemented: Two Higgs Doublet Model
- More to come: UFO model import in development

http://openloops.hepforge.org