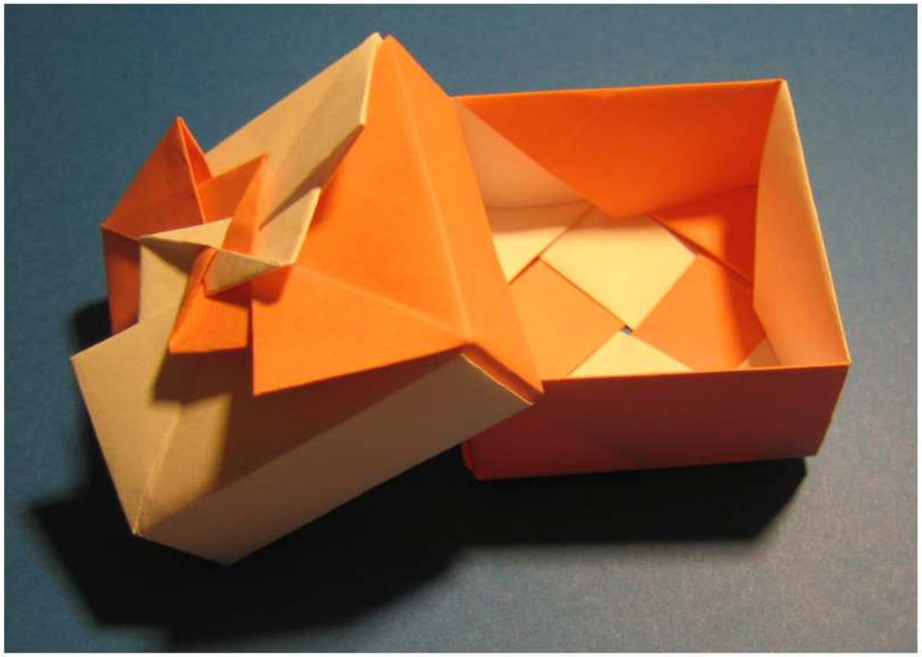
Status and Future plans for POWHEG

Carlo Oleari
Università di Milano-Bicocca, Milan

MC generators and future challenges, a joint ATLAS/CMS/LPCC workshop
19 November 2012

- List of processes in the POWHEG BOX
- Automation
- Scale and pdf's uncertainties by reweighting
- Multi-scale Improved NLO
- NLO corrections in decays
- Parallelization
- Conclusions



Processes implemented in the POWHEG BOX

Full list at: http://powhegbox.mib.infn.it

- Heavy quarks
  - heavy-quark pair production (Frixione, Nason, Ridolfi, 2007)
  - $t\bar{t}$ production with decay, with exact spin-correlations in decay products. NLO corrections only in the production part of the process (Campbell, Ellis, Nason, 2012)
  - single top (Alioli, Nason, Re, C.O., 2009) and tW (Re, 2010) + t-channel four-flavor scheme (Re, 2012)
  - $t\bar{t} + 1$ jet (Kardos, Papadopoulos, Trocsanyi, 2011 $^{(*)}$) and (Alioli, Moch, Uwer, 2011)
  - $t\bar{t}Z/W$ (Garzelli, Kardos, Papadopoulos, Trocsanyi, 2011-12 $^{(*)}$)
  - $Wb\bar{b}$ (Reina, C.O., 2011)

$^{(*)}$POWHEL, obtained merging POWHEG and HELAC, using the POWHEG BOX.

In cyan, processes that rely on the POWHEG BOX but have not yet been added to the public version of the BOX.

Processes implemented in the POWHEG BOX

- Vector bosons
  - Z/W (with decay) (Alioli, Nason, Re, C.O., 2008)
  - Z/W (with decay) + 1 jet (Alioli, Nason, Re, C.O., 2010)
  - W^+W^+ plus two jets (Melia, Nason, Rontsch, Zanderighi, 2011)
  - W^+W^+ plus two jets via VBF (Jäger, Zanderighi, 2011)
  - diboson production (with decay), (Melia, Nason, Rontsch, Zanderighi, 2011)
  - W production with some EW corrections, (Bernaciak, Wackeroth, 2012)
  - W production with EW corrections and QED shower, (Barzè, Montagna, Nason, Nicrosini, Piccinini, 2012)
  - Zjj in VBF (Jäger, Schneider, Zanderighi, 2012)
  - Zjj (Re, 2012)

$\Longrightarrow$ Please notice that EW corrections have been addressed recently in the POWHEG BOX, with multi-photon emissions treated on the same ground as gluon emissions.

Processes implemented in the POWHEG BOX

- Jet production
  - dijet (Alioli, Hamilton Nason, Re, C.O., 2010)

- Higgs boson
  - Higgs boson in gluon fusion (Alioli, Nason, Re, C.O., 2008)
  - Higgs boson in VBF (Nason, C.O., 2010)
  - $t\bar{t}H$ (Garzelli, Kardos, Papadopoulos, Trocsanyi, 2011 $^{(*)}$)
  - tH^- (Klasen, Kovarik, Nason, Weydert,2012)
  - Higgs boson in gluon fusion with quark mass and EW effects (Bagnaschi, Degrassi, Slavich, Vicini, 2011)
  - Hj and Hjj in gluon fusion (Campbell, Ellis, Frederix, Nason, Williams and C.O., 2012)

- SUSY
  - Slepton pair production (B. Jäger, A. von Manteuffel, S. Thier, 2012)

Towards automation

The POWHEG BOX automates many features of the generation of NLO + parton shower events, except for the following:

- ✓ the lists of the Born and real processes
- ✓ the Born phase space
- ✓ the Born squared amplitudes, the color-correlated and spin-correlated amplitudes, for all partonic subprocesses
  All these amplitudes are common ingredients of a NLO calculation
- ✓ the real squared amplitude for all the relevant real-emission subprocesses
- ✓ the finite part of the virtual corrections, computed in conventional dimensional regularization or in dimensional reduction
- ✓ the Born color structures in the limit of large number of colors.

Automation

The are two useful interfaces:

- ✓ an interface to MadGraph 4, built in collaboration with Rikkert Frederix, that automatically builds the codes to compute the Born, Born color- and spin-correlated amplitudes, the real amplitude and the Born color structure in the limit of large number of colors. This is done just once and for all, when a new process is implemented in the POWHEG BOX.

  This interface has been used to build the code for Hj and Hjj production, with the virtual part taken from the MCFM code (Campbell, Ellis, Frederix, Nason, Williams and C.O.)

- ✓ an interface to GoSam (Cullen, Greiner, Heinrich, Luisoni, Mastrolia, Ossola, Reiter, Tramontano), built in collaboration with Luisoni and Tramontano , that writes automatically the code for the computation of the finite part of the virtual contributions. Currently under development (Luisoni, Nason, Tramontano and C.O.)

There is also a commitment of the authors of MCFM [Williams, Campbell, Ellis] to keep in mind the POWHEG BOX needs when building the virtual routines.

Scales and pdf's uncertainties

- The calculation of renormalization- and factorization-scale uncertainties and pdf's dependence is very time consuming if one has to generate new Monte Carlo samples for each pdf set and scale choices.

- It would be desirable to have, associated with each event, extra information for a fast calculation of scale and pdf dependencies.

- The POWHEG BOX provides, since some time, a primitive pdf reweighting facility (used by ATLAS in the study for dijet production).

- Scale and pdf reweightings in the POWHEG BOX are now implemented (Hamilton, Nason, Re)

How it works

Since the POWHEG BOX relies upon third-party matrix elements, it is difficult to track the coefficients of scale logarithms.

The procedure followed by the POWHEG BOX is then the following:

- At the first run, it saves on the event file, for each generated event, the value of the random-number seeds used to compute the last phase-space point, the value of the integrand ($\tilde{B}$ function) plus some other information to recover the flavor structure of the event.
- Subsequent runs can be done using different scales and pdf sets with respect to the ones used in the first generation of the event.

  For each event, the POWHEG BOX sets the random-number seeds to the stored values (to generate the same phase-space point), and computes the integrand again, with the desired new scales and pdf set.

  Then it stores the same event in a new file, appending to it the new weight computed as

$$\text{new weight} = \text{old weight} \times \frac{\text{new integrand}}{\text{old integrand}}$$

The procedure can be repeated as many times as one likes, appending to each event as many new weights as one desires.

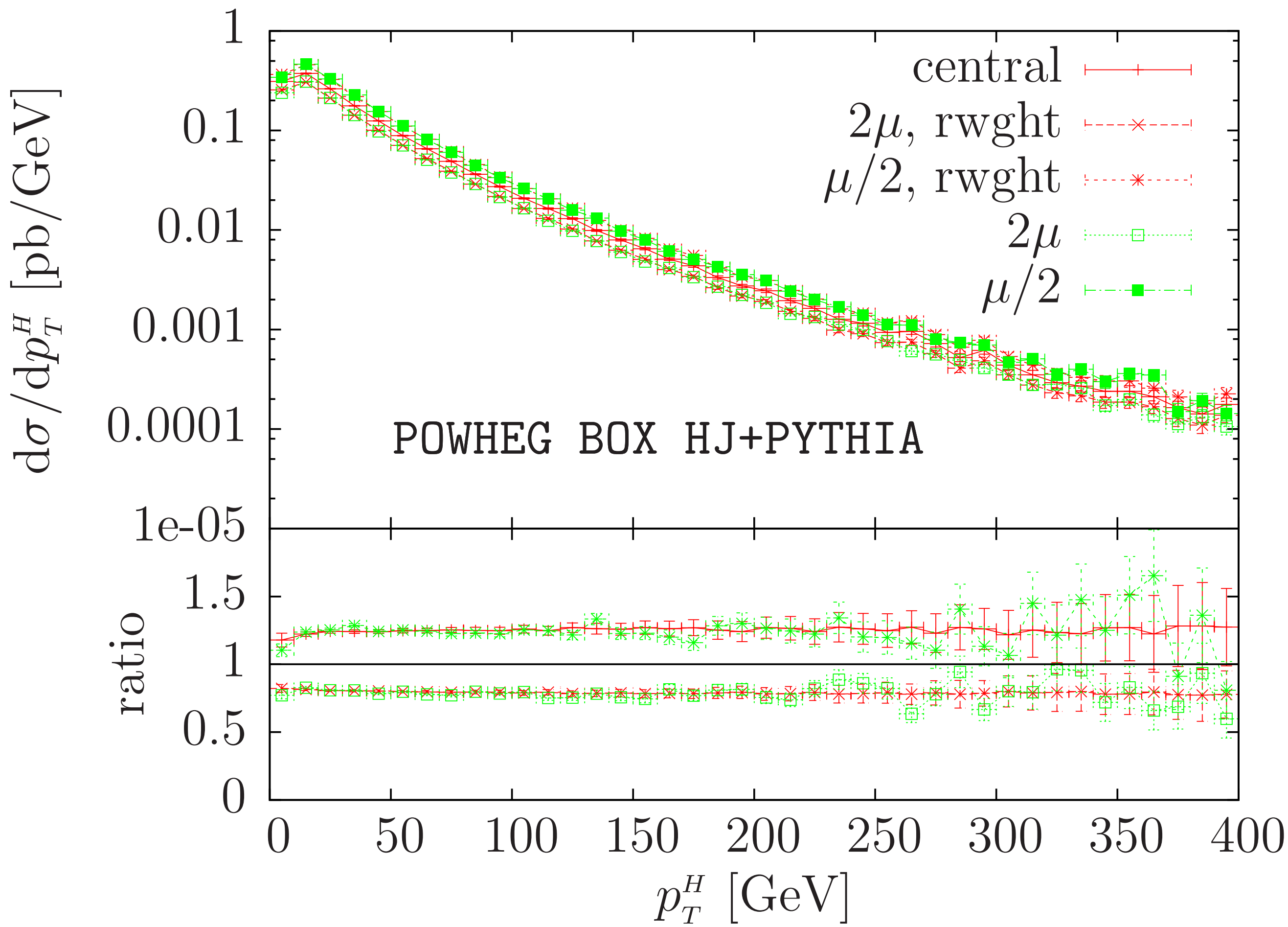
Outputs

```
<event>
      5  10001  1.11742E+01  5.31839E+00 -1.00000E+00  2.43019E-01
      21    -1     0     0   503   501  0.0000E+00  0.0000E+00  3.8159E+02  3.8159E+02  0.0000E+00  0E+00  9E+00
      21    -1     0     0   511   503  0.0000E+00  0.0000E+00 -2.6130E+02  2.6130E+02  0.0000E+00  0E+00  9E+00
      25     1     1     2     0     0 -1.0585E+01  5.8747E+01  3.0383E+02  3.3391E+02  1.2499E+02  0E+00  9E+00
      21     1     1     2   502   501  1.1414E+01 -5.3494E+01  5.0756E+01  7.4619E+01  1.5078E-06  0E+00  9E+00
      21     1     1     2   511   502 -8.2893E-01 -5.2533E+00 -2.3430E+02  2.3436E+02  2.6973E-06  0E+00  9E+00
 #rwgt           1          31   11.4024688439630           10103      16           0
 #new weight,renfact,facfact,pdf1,pdf2   11.17420000        1.00000        1.00000       21100        21100   lha
 #new weight,renfact,facfact,pdf1,pdf2   9.120545413        2.00000        2.00000       21100        21100   lha
 #new weight,renfact,facfact,pdf1,pdf2   13.32958972        0.50000        0.50000       21100        21100   lha
 #new weight,renfact,facfact,pdf1,pdf2   11.80161381        1.00000        2.00000       21100        21100   lha
 #new weight,renfact,facfact,pdf1,pdf2   8.733756763        2.00000        1.00000       21100        21100   lha
 #new weight,renfact,facfact,pdf1,pdf2   10.41747715        1.00000        0.50000       21100        21100   lha
 #new weight,renfact,facfact,pdf1,pdf2   14.55076449        0.50000        1.00000       21100        21100   lha
</event>
```

The `#rwgt` line contains info to recover flavor and type of the POWHEG BOX subprocess, the integrand and the random-number seeds. It is generated at the first run.

The `#new weight` . . . lines are generated in subsequent (very quick!) runs, where only a single call to the cross section is made for each event, in order to compute the `new weight`.

Reweighted versus direct computation



The reweighting code can be found at:

svn://powhegbox.mib.infn.it/trunk/POWHEG-BOX/ReweightingStuff

with instructions on how to apply it to specific processes. It will be merged with the main POWHEG BOX code as soon as we are confident that it does what is needed.

Feedback from experimentalists is welcome.

MINLO: Multi-scale Improved NLO

✓ The purpose of MINLO is to improve the NLO computation of inclusive quantities when regions of the phase space with widely different scales are approached.

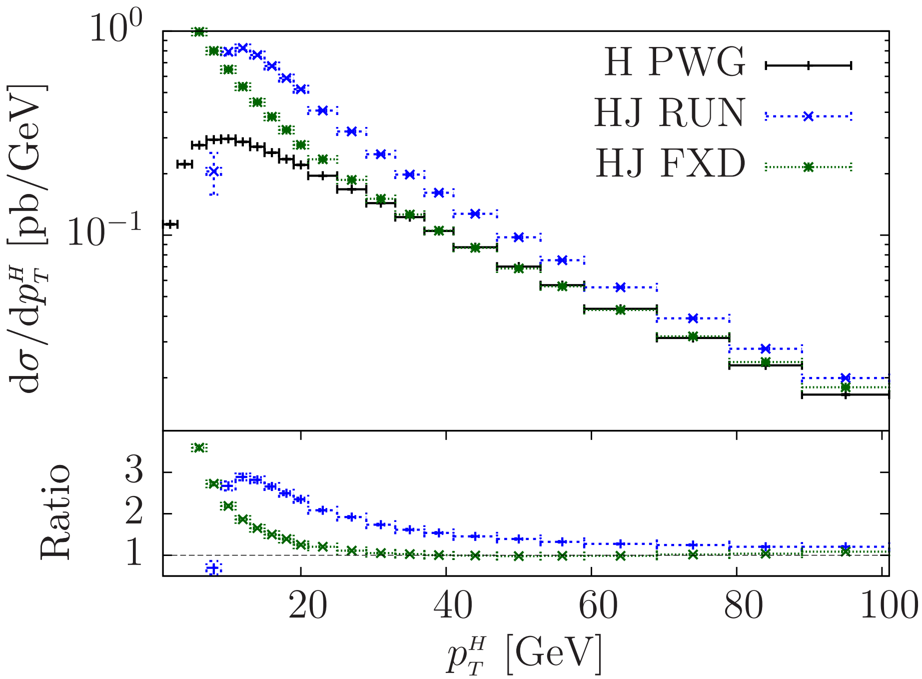
✓ The MINLO procedure (Hamilton, Nason, Zanderighi, 2012) has been inspired by the CKKW method. It achieves its goals by:

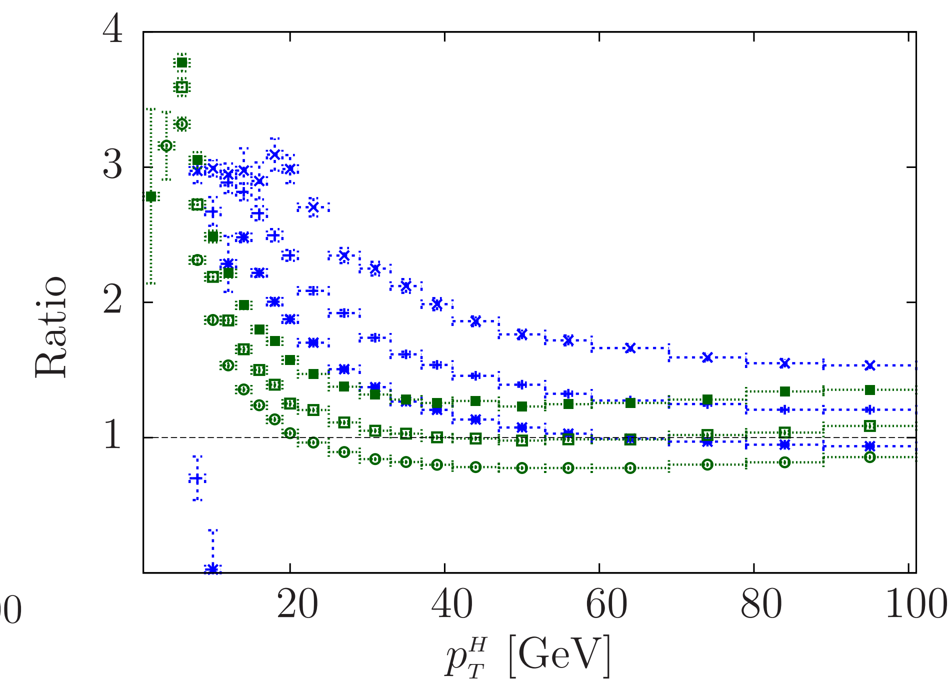
- finding the most probable branching history
- computing for each vertex a branching scale and using it as renormalization scale for the corresponding α_s
- adding appropriate Sudakov form factors to exponentiate large logarithms present in the NLO calculation
- subtracting the expansion of the Sudakov form factors at NLO from the fixed NLO result
- assigning to the factorization scale the value of the smallest branching scale in the underlying Born
- ...

✓ The MINLO procedure has been implemented and made public in the POWHEG BOX and can be found at:

`svn://powhegbox.mib.infn.it/trunk/POWHEG-BOX/MINLO`

HJ NLO vs H POWHEG





- H PWG: the POWHEG BOX gg_H generator interfaced with PYTHIA, $m_H = 120$ GeV
- HJ RUN: Hj NLO calculation with $\mu_R = \mu_F = p_T^H$
- HJ FXD: Hj NLO calculation with $\mu_R = \mu_F = m_H$
- Ratios over H PWG

- Error bands obtained varying μ_R and μ_F by a factor of 2 above and below their common central value, with the constraint $1/2 \leq \mu_R/\mu_F \leq 2$
- Bands do not overlap at $p_T^H \lesssim 30$ GeV
- NLO shapes differ from LL resummed result (POWHEG) at small p_T^H

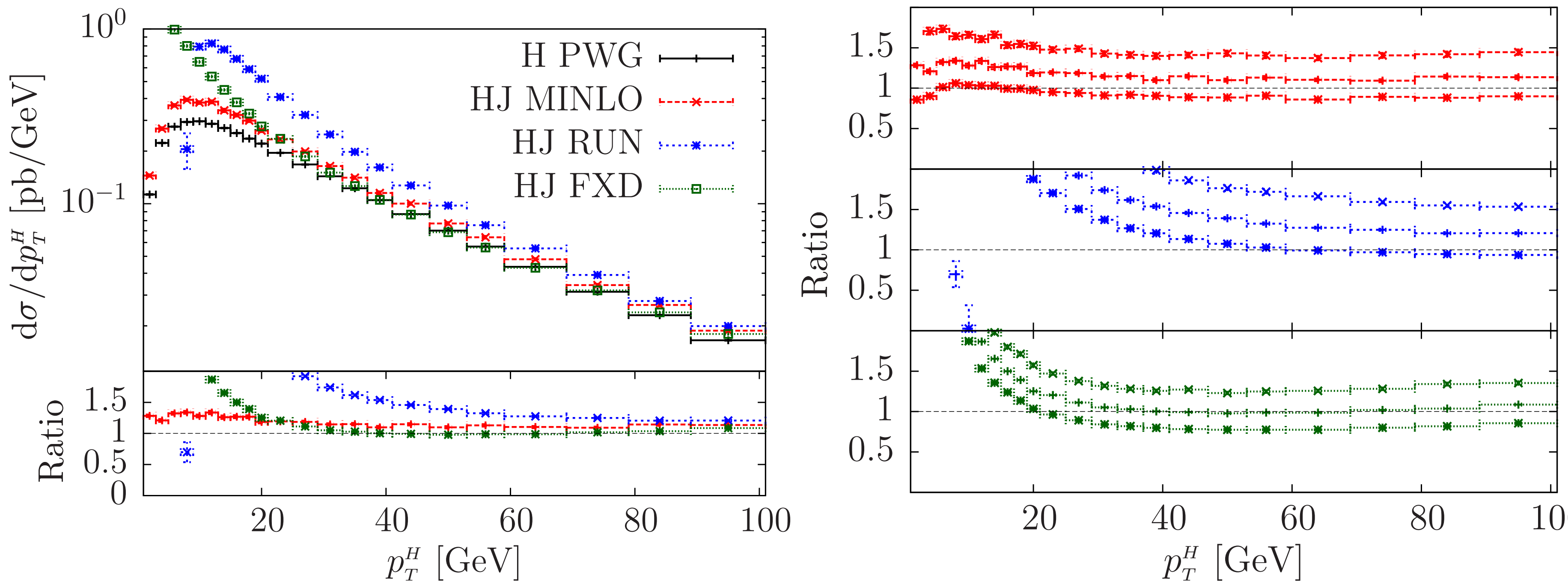
MINLO HJ

- In the standard POWHEG implementation of Hj production, the NLO computation of the Hj cross section suffers from large uncertainties due to scale choices, and, furthermore, does not have a good match with the H production POWHEG generator, at small transverse momenta.

- This problem is easily tracked back to the fact that the Hj NLO calculation does not attempt to resum large logarithms of the jet transverse momentum, not even at the LO level.

- We then apply the MINLO procedure when computing the $\bar{B}$ function for the generation of the underlying Born kinematics.

MINLO HJ



- H PWG: the (showered) `gg_H` POWHEG BOX result
- RUN and FXD need a generation cut (or Born suppression factor) at small p_T^H. The MINLO result is instead finite (up to a cut-off $\approx \Lambda_{\rm QCD}$)
- We can trust the MINLO result at small p_T^H only as a LO result (see the widening of the MINLO uncertainty band at small p_T^H). However, at least, we get a result that is sensible also at low p_T^H, rather than divergent.

MINLO-POWHEG

- ✓ The MINLO approach improves the POWHEG implementations involving associated jet production, in the singular phase-space region.
- ✓ It provides a better match with the corresponding lower-multiplicity process. For example, Hjj matches Hj when approaching the one-jet region, and Hj matches H when approaching the zero-jet region.
- ✓ It eases considerably the construction of matched samples with different jet multiplicities.

Further work for finding simple criteria for matching MINLO-POWHEG samples of different multiplicities is in progress.

Resonance decays in NLO+PS

- NLO+PS generators usually include NLO corrections only to the production part of the process. Thus, for example, in W production with subsequent decay to hadrons, no NLO corrections are included for the W decay products. Same in $t\bar{t}$ production, where the NLO corrections to the $t \to Wb$ decay are not included, and, in case the W decays hadronically, the corresponding NLO corrections are also not included.

- Some shower generators are capable to include approximate NLO effects in t and W decays (see, for example, Pythia 8).

- Several NLO results, including NLO corrections to top decays, are available in the literature (Melnikov and Schulze; Bernreuther and Si; Campbell and Ellis)

- The POWHEG BOX has been upgraded to deal with NLO corrections in decays.

Resonance decays in NLO+PS

- Several aspects of the POWHEG BOX have been modified in order to keep track not only of the list of particles for each subprocess but also of the information regarding to which resonance they belong to
- Radiation can now come both from the production part of the process and from the decay part
- Since the POWHEG BOX also implements an automatic subtraction scheme for NLO calculations, a few parts of this procedure have been updated for resonance treatment
- These extensions have all been completed. They are designed to be fully general

First results will appear for the $t\bar{t}$ and single top processes. Work is in progress (Campbell, Ellis, Nason).

Interface to shower

- With no NLO corrections to decays (but exact NLO spin correlations): can be interfaced to shower as before.

- With NLO corrections to decays: POWHEG generates the hardest radiation, whether it comes from production or decay. The shower should not generate anything harder, neither in production nor in decay.

In the standard Les Houches interface, we can veto radiation in production, by setting the variable `scalup`. The shower treats the resonance decays independently.

There is the need to go beyond the standard Les Houches interface.

For example, Pythia 8 has methods for vetoing the shower off a resonance, allowing for greater flexibility. It is desirable that future interface accords will take account of these needs too.

Parallelization

- Currently the POWHEG BOX can be run on a multicore computer or on several nodes in parallel: not only the generation of events can be done in parallel, but also the calculation of the grids.

- We have a few script files that can guide the user on what to do to set and start the runs

- more info can be found at:
  `svn://powhegbox.mib.infn.it/trunk/POWHEG-BOX/Parallel-grids`

Conclusions

- ✓ The list of POWHEG BOX processes increases steadily. See

  `http://powhegbox.mib.infn.it`

  for an up to date list.

- ✓ Progresses are made towards the full automation. Collaborations with developers of automatic codes are welcome.

- ✓ In order to quickly compute scale uncertainties and pdf dependencies, a facility for event reweighting has been set up.

- ✓ The MINLO procedure yields a considerable improvement of the generators near the singular regions, allowing for better matching among samples with different jet multiplicity.

- ✓ The POWHEG BOX has been extended, to allow for the inclusion of NLO corrections to decay processes.

- ✓ Efforts have been made towards parallelization of several steps of the POWHEG BOX generation.

- ✓ In the next future we will merge all these new features into a single POWHEG BOX code: POWHEG BOX 2.0