

The POWHEG BOX

P. Nason
INFN, Sez. of Milano Bicocca

CERN, Aug. 10th 2009

POWHEG

Positive Weight Hardest Emission Generator

Method to generate the hardest emission first, with NLO accuracy, and independently of the SMC (P.N. 2004).

- SMC independent; no need of SMC expert; same calculation can be interfaced to several SMC programs with no extra effort
- SMC inaccuracies in the soft region only affect next-to-hardest emissions; no matching problems
- As the name says, it generates events with positive weight

Status of POWHEG

Up to now, the following processes have been implemented in POWHEG:

- $hh \rightarrow ZZ$ (Ridolfi, P.N., 2006)
- $e^+e^- \rightarrow \text{hadrons}$, (Latunde-Dada, Gieseke, Webber, 2006),
 $e^+e^- \rightarrow t\bar{t}$, including top decays at NLO (Latunde-Dada, 2008),
- $hh \rightarrow Q\bar{Q}$ (Frixione, Ridolfi, P.N., 2007)
- $hh \rightarrow Z/W$ (Alioli, Oleari, Re, P.N., 2008;)
(Hamilton, Richardson, Tully, 2008;)
- $hh \rightarrow H$ (gluon fusion) (Alioli, Oleari, Re, P.N., 2008; Herwig++)
- $hh \rightarrow H$, $hh \rightarrow HZ/W$ (Hamilton, Richardson, Tully, 2009;)
- $hh \rightarrow t + X$ (single top) **NEW** (Alioli, Oleari, Re, P.N., 2009)
- $hh \rightarrow Z + \text{jet}$, **Very preliminary** (Alioli, Oleari, Re, P.N., 2009)
- The **POWHEG BOX**, **Very preliminary**, (Alioli, Oleari, Re, P.N., 2009)

Towards automation: the POWHEG BOX

The MIB (Milano-Bicocca) group (Alioli, Oleari, Re, P.N.) has developed an automatic implementation of POWHEG for generic NLO processes. As a byproduct, it is also a general implementation of the subtraction method for NLO processes.

The framework was built around an implementation of $hh \rightarrow Z + 1\text{jet}$. It has been tested against existing implementation of $hh \rightarrow W/Z$, and single top production.

Plans:

- Write a paper describing all details of the algorithms.
The aim is to make the code understandable by others.
- Release the code

Authors of NLO calculations will be in a position to build their own POWHEG implementation using the POWHEG BOX.

It is likely that, as more processes are added, new problems will arise; thus, the POWHEG BOX will not be a BLACK BOX. It should be considered as a starting point for new implementations, where all previously encountered problems have been solved.

Thus, we will happily collaborate with people building new implementations.

The POWHEG BOX

The code framework is such that, given the Born cross section, the finite part of the virtual corrections, and the real graph cross section, one builds immediately a POWHEG generator. More precisely, the **user** must supply:

- The **Born phase space**
- The **lists of Born and Real** processes (i.e. $u \bar{s} \rightarrow W^+ c \bar{c}$, etc.)
- The **Born squared amplitudes** $\mathcal{B} = |\mathcal{M}|^2$, \mathcal{B}_{ij} , $\mathcal{B}_{j,\mu_j,\mu'_j}$, for all relevant partonic processes; \mathcal{B}_{ij} is the colour ordered Born amplitude squared, $\mathcal{B}_{j,\mu\nu}$ is the spin correlated amplitude, where j runs over all external gluons in the amplitude. All these amplitudes are common ingredient of an NLO calculation.
- The **Real squared amplitude**, for all relevant partonic processes.
- The finite part of the **virtual amplitude** contribution, for all relevant partonic processes.

MADGRAPH interface

Born (total, colour ordered and spin correlated) and real cross section can be obtained from MADGRAPH and MadDipole (Frederix, Gehrman, Greiner, 2008). We plan to build an interface to MADGRAPH and MadDipole, so that: **only the Born phase space and the virtual corrections** will be needed to build an implementation.

Strategy

Use the FKS framework according to the general formulation of POWHEG given in (Frixione, Oleari, P.N. 2007), hiding all FKS implementation details.

In other words, we use FKS, but the user needs not to understand it.

(Attempts to use the Catani-Seymour method did not work ...)

Ingredients provided by the BOX:

- The phase space for ISR and FSR, according to FNO2006.
- The combinatorics, the calculation of all R_α , the soft and coll. limits
- The calculation of \bar{B} . This is the inclusive cross section at fixed underlying Born variables. Soft and collinear remnants are automatically built by the BOX. (spinoff: NLO implementation using the FKS method)
- Generation of radiation
- Writing the event to the Les Houches interface

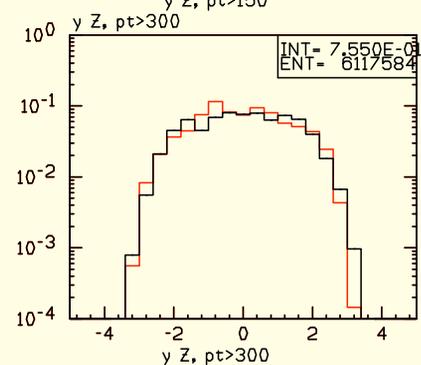
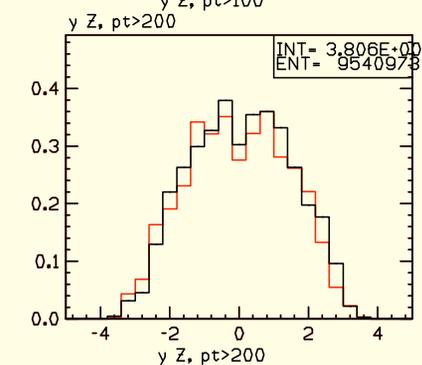
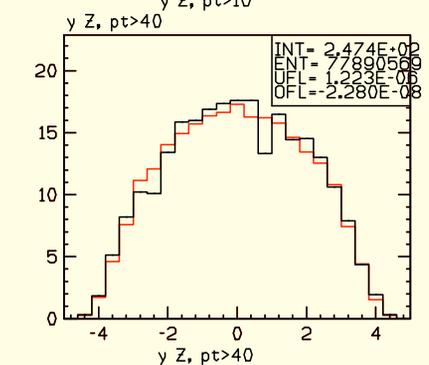
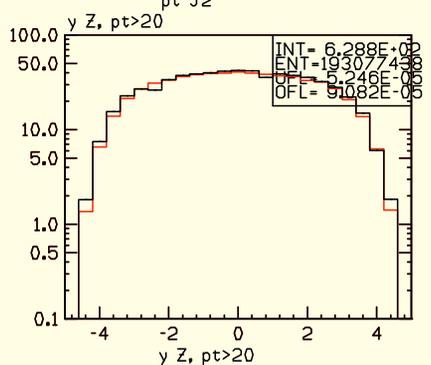
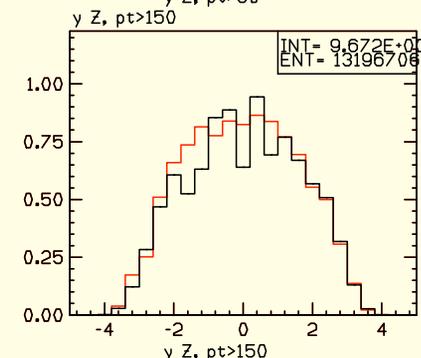
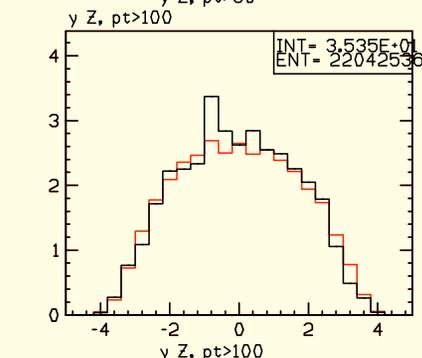
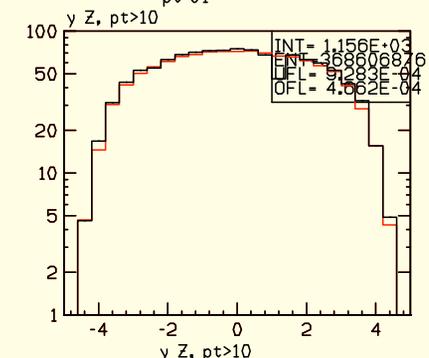
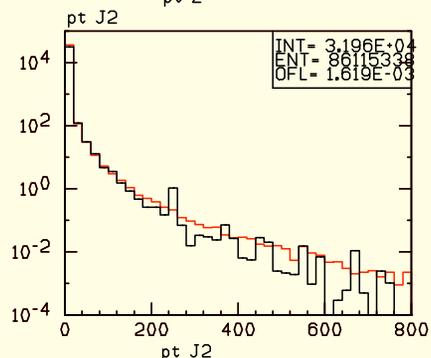
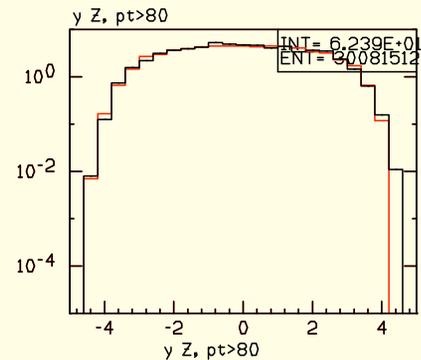
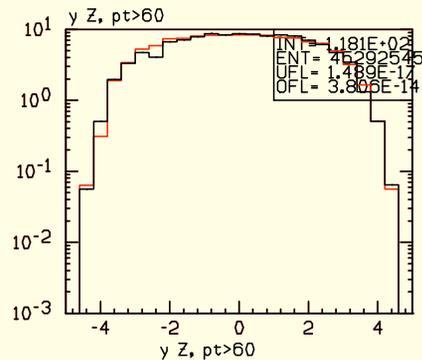
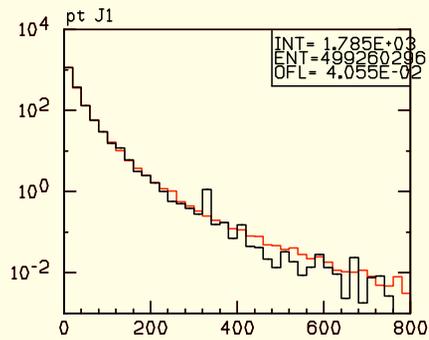
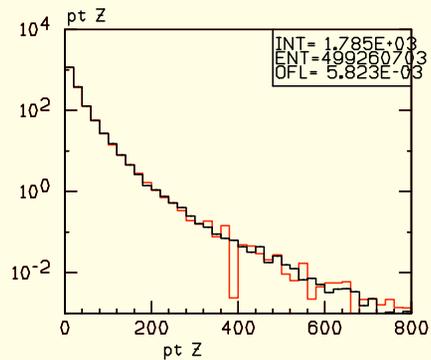
It works! Lots of more testing needed now ...

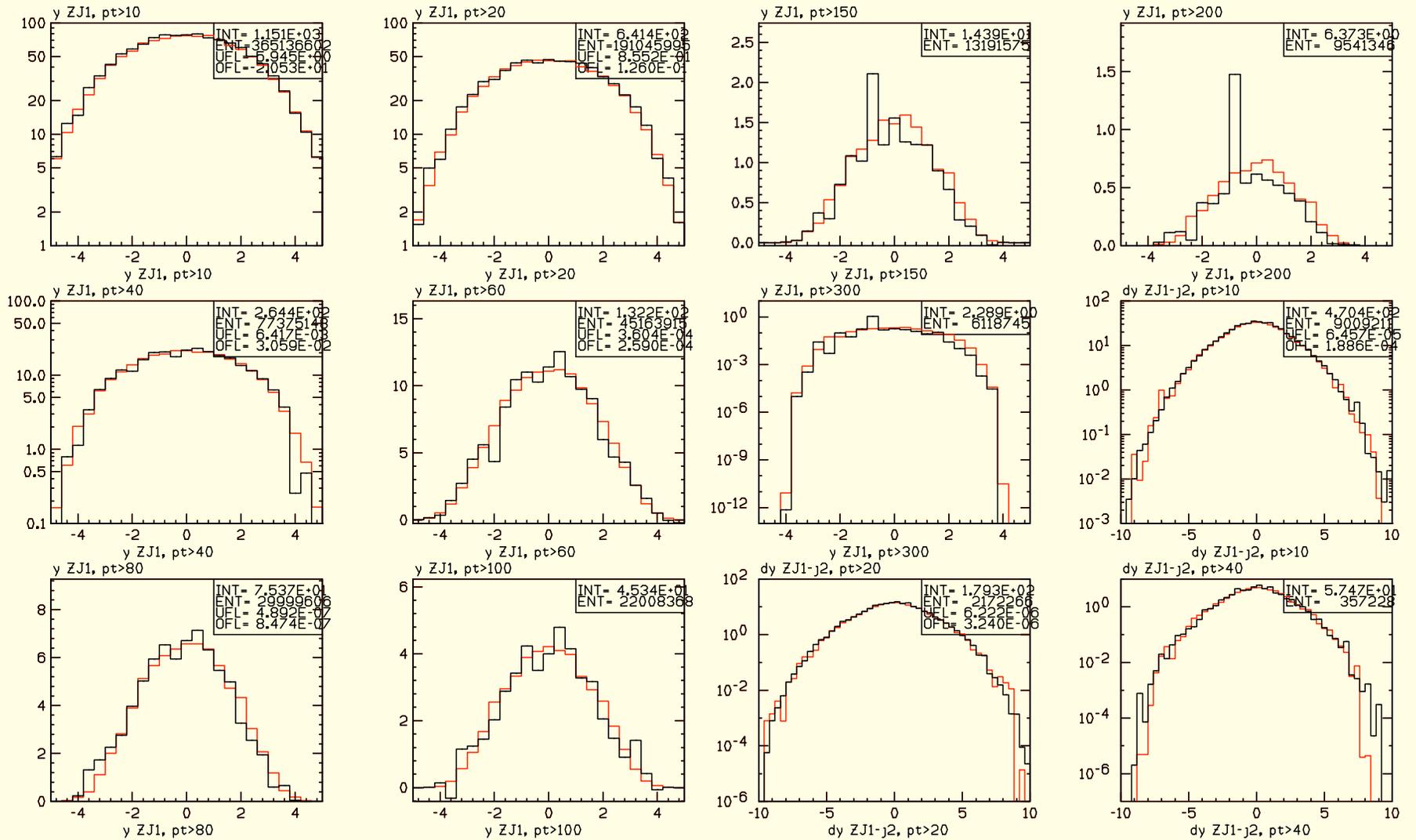
Case study: $Z + \text{jet}$ production

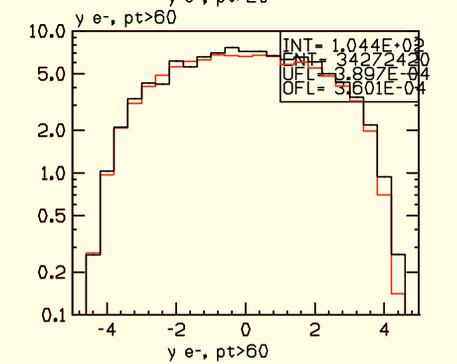
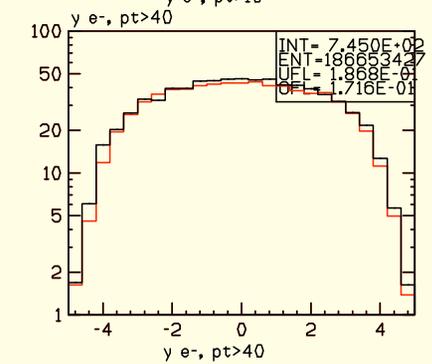
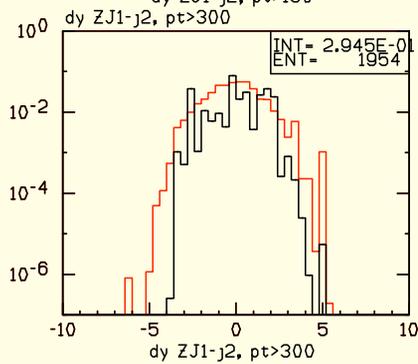
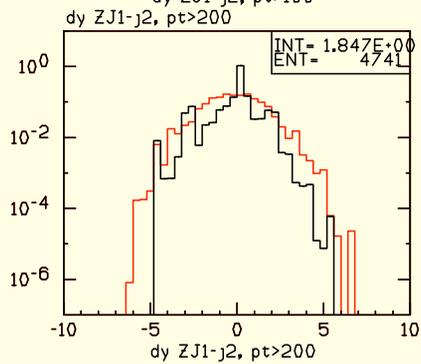
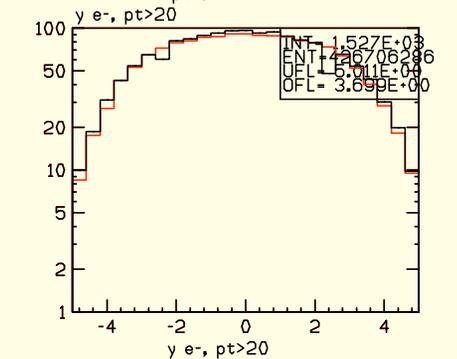
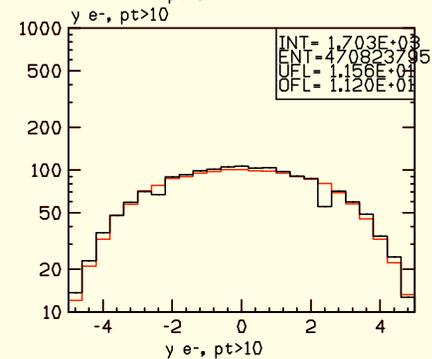
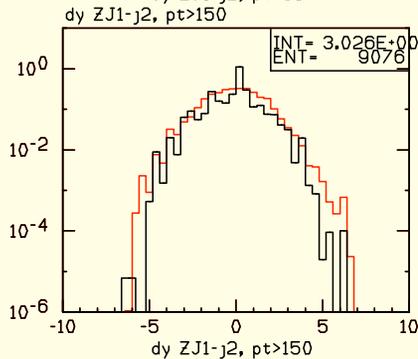
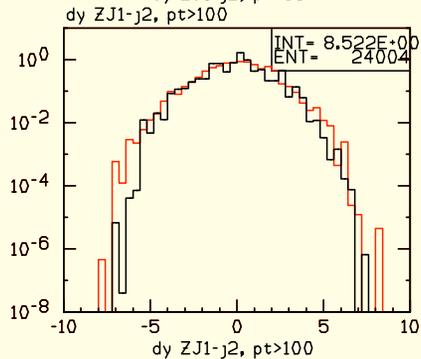
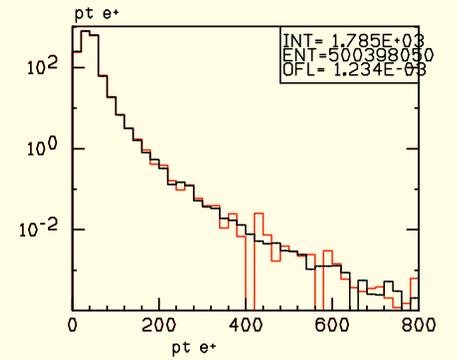
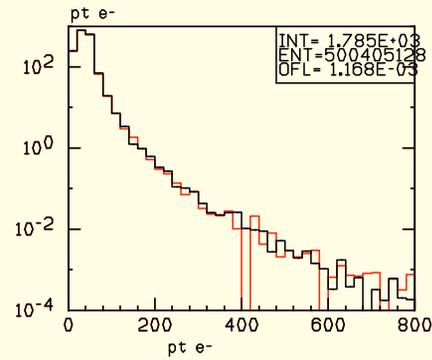
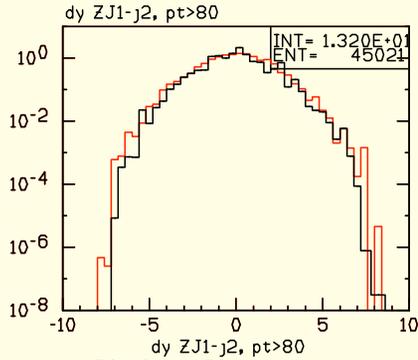
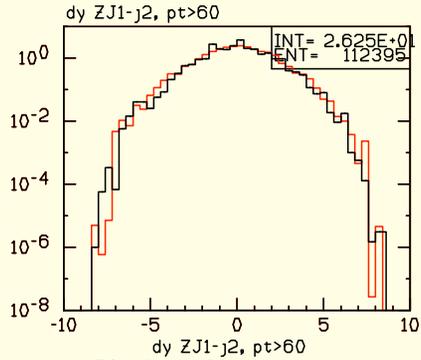
Got virtual matrix elements from MCFM;

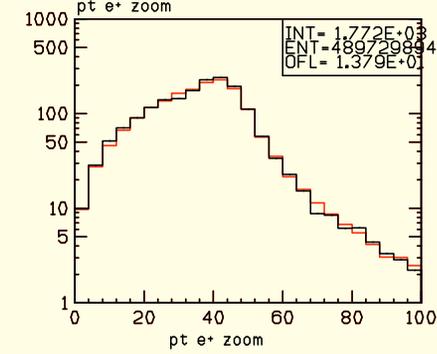
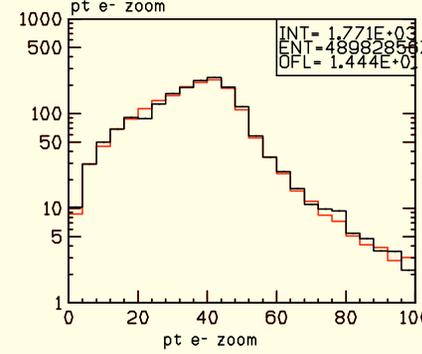
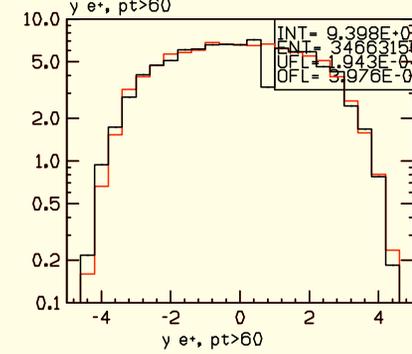
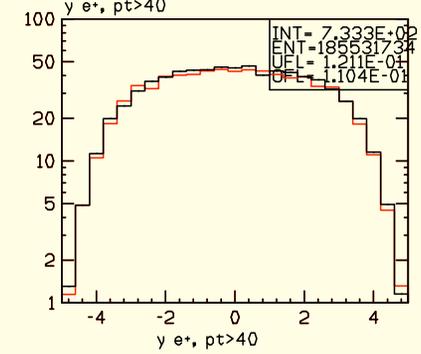
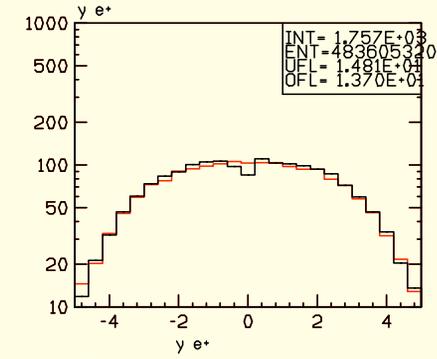
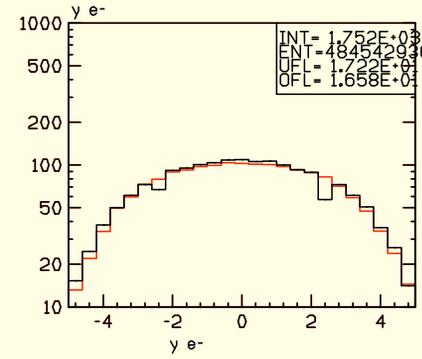
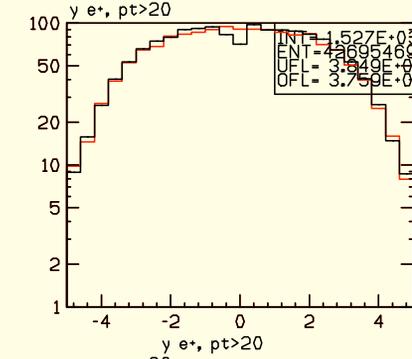
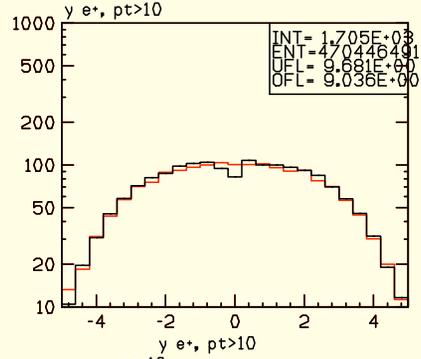
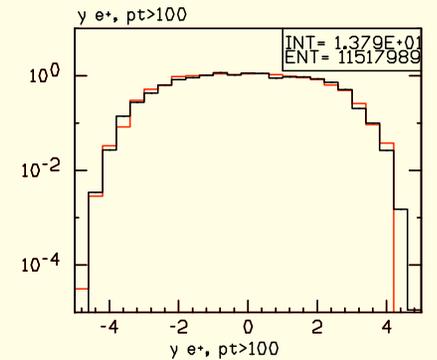
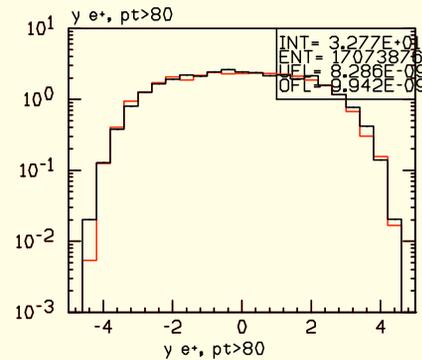
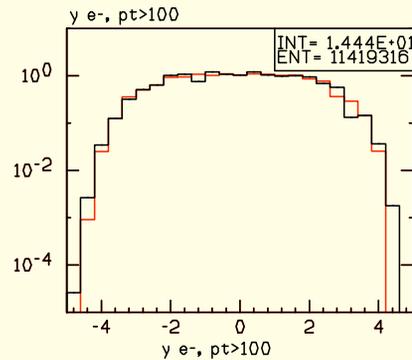
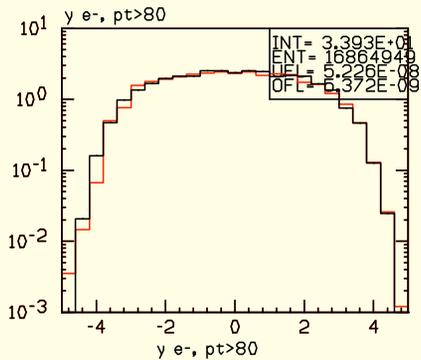
Compare NLO predictions obtained with MCFM and the POWHEG BOX

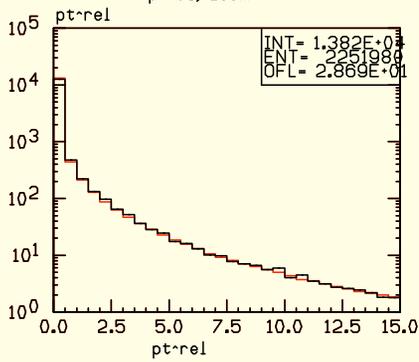
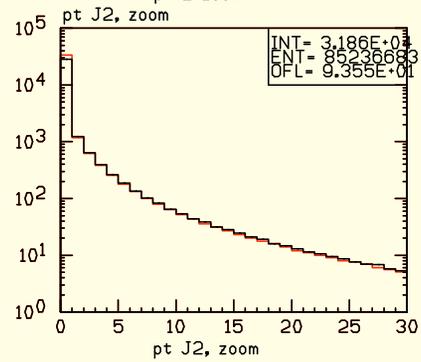
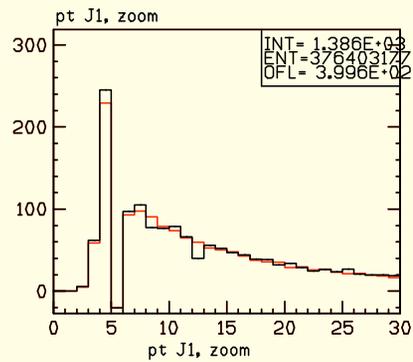
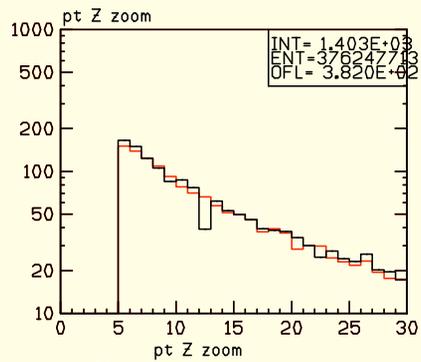
Virtual corrections are the same, but subtraction terms, soft and collinear remnants are all different; non trivial test;





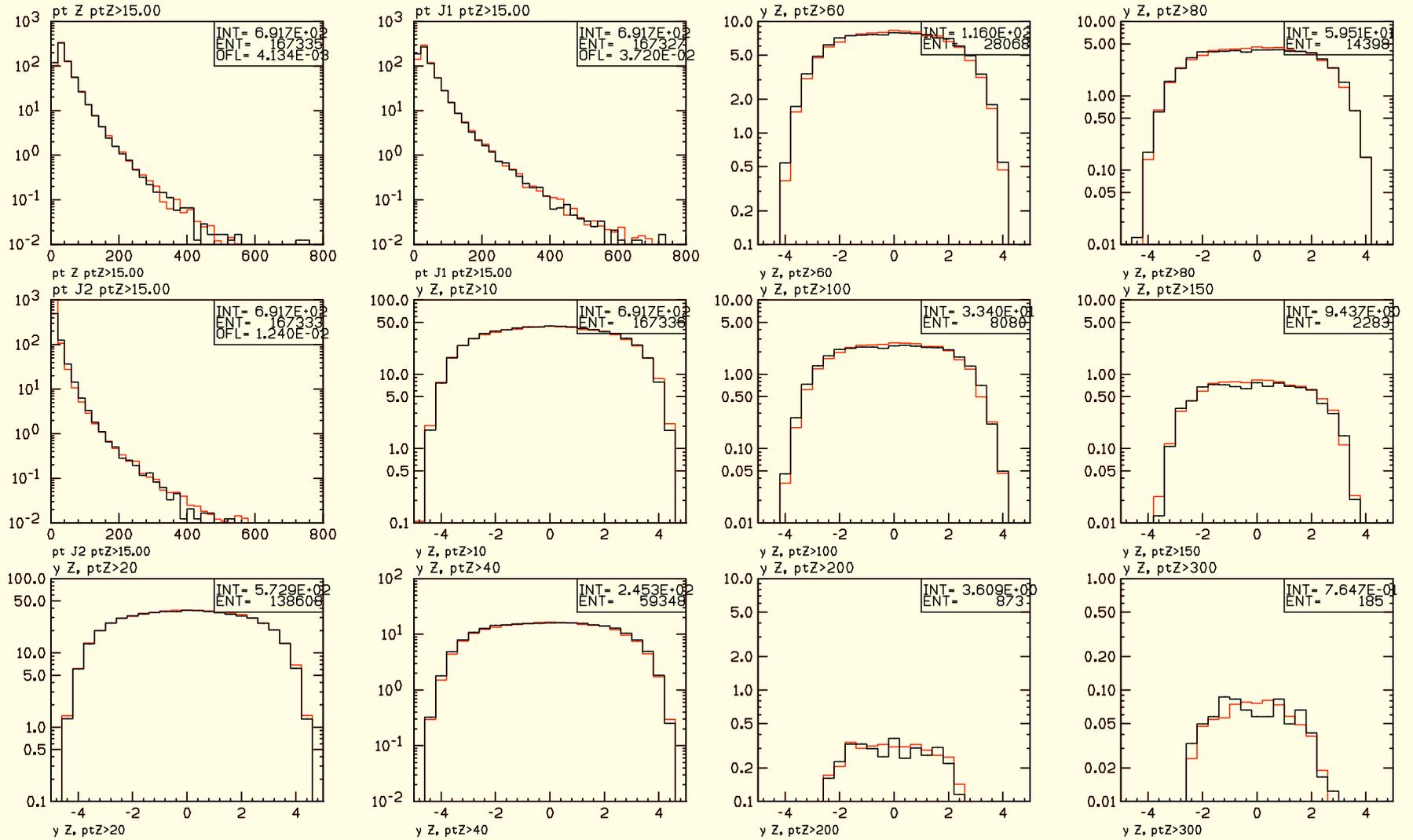


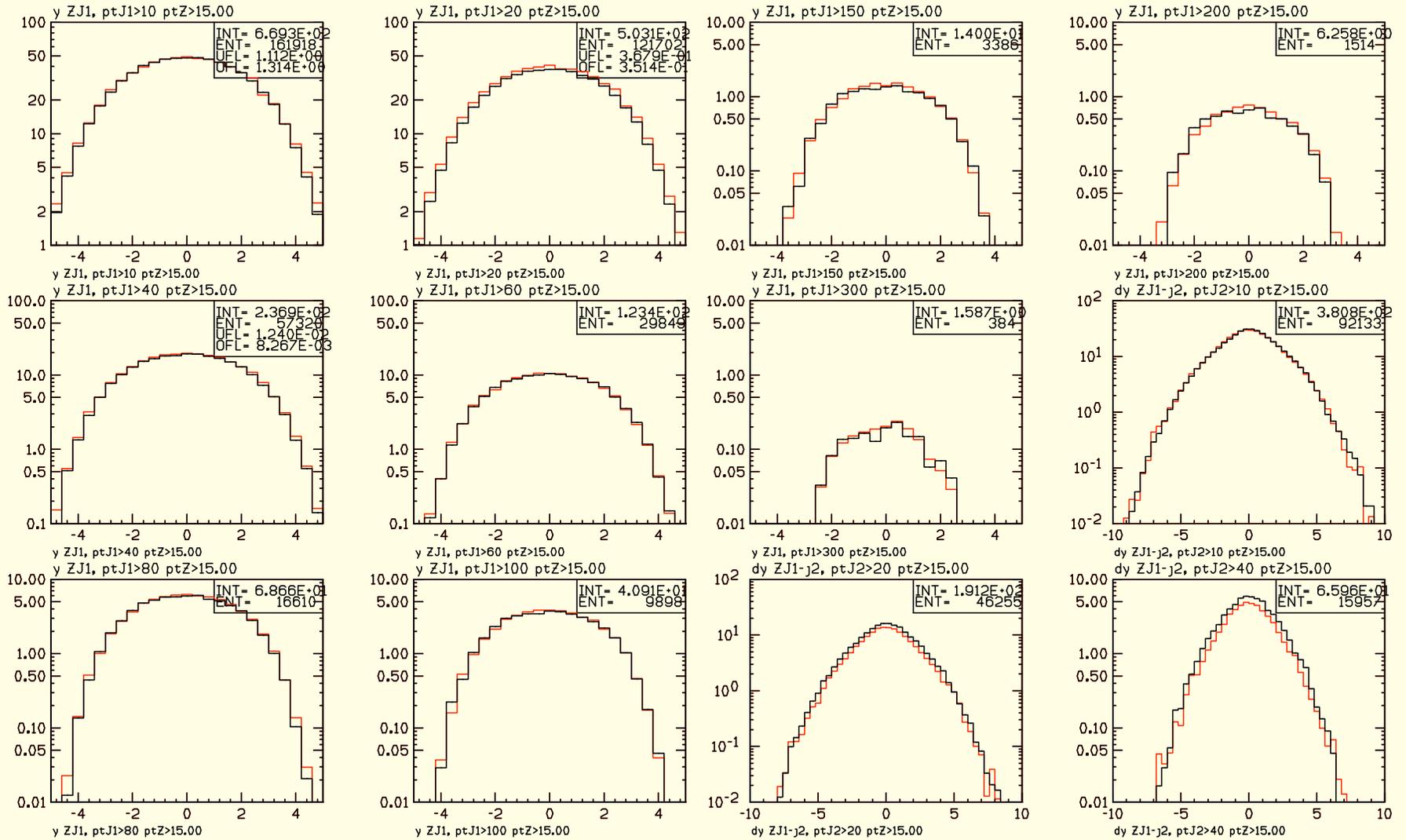


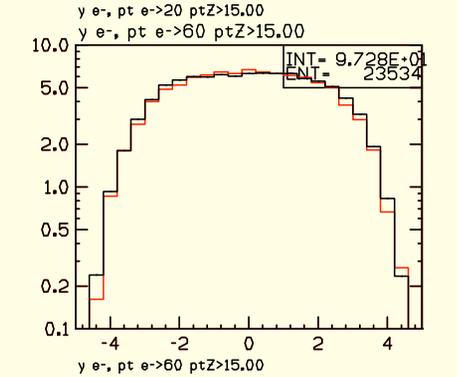
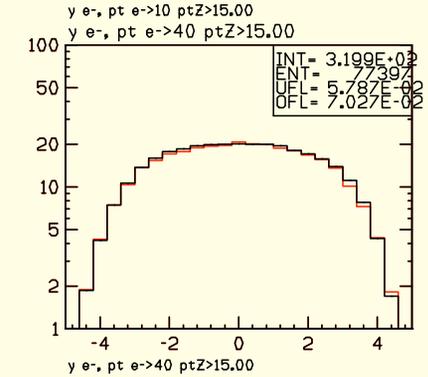
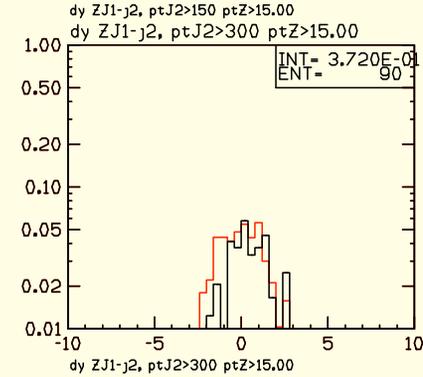
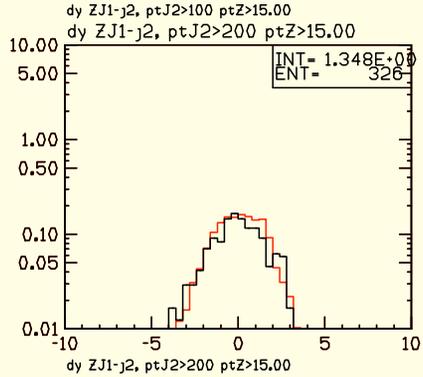
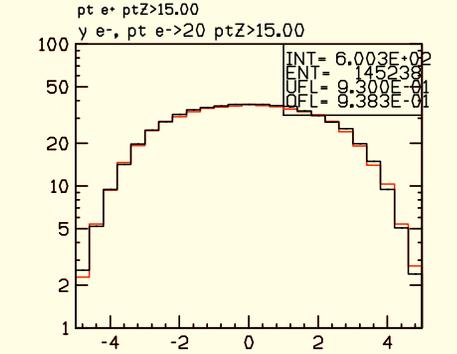
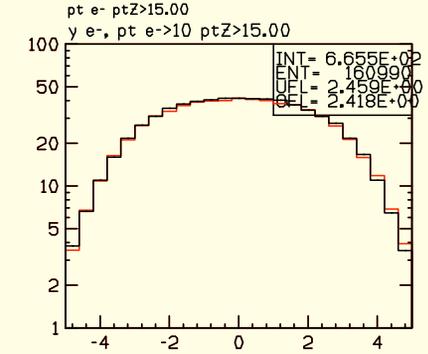
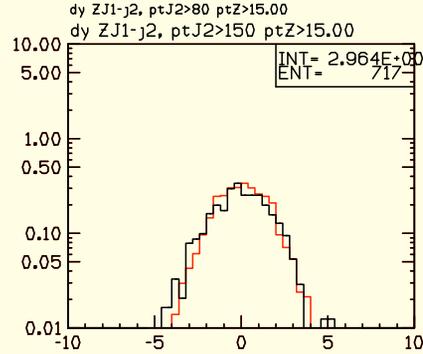
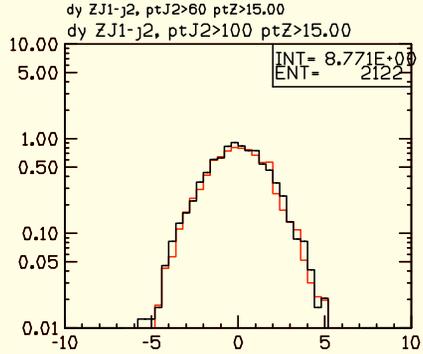
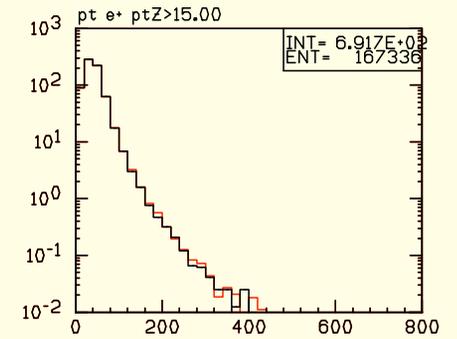
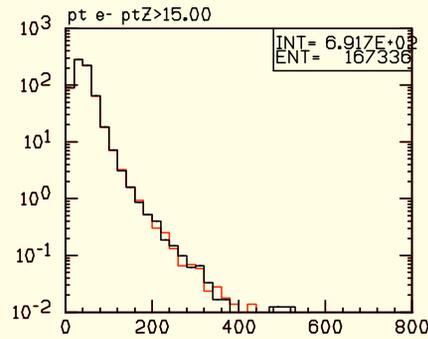
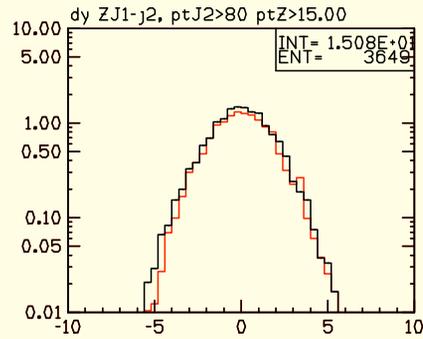
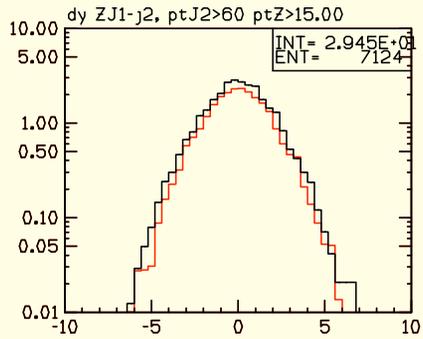


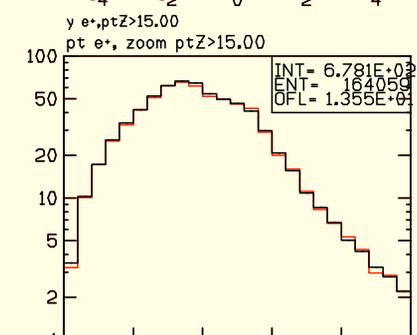
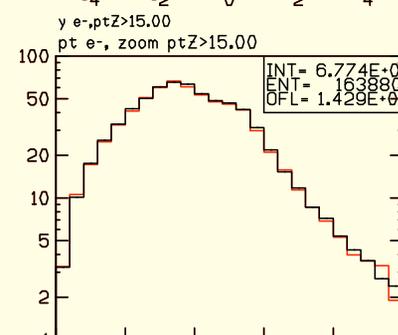
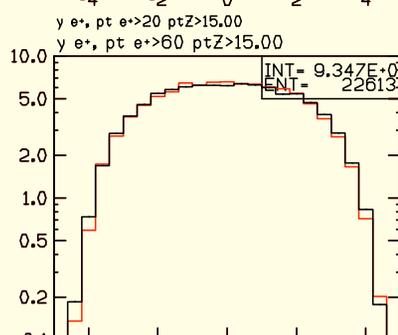
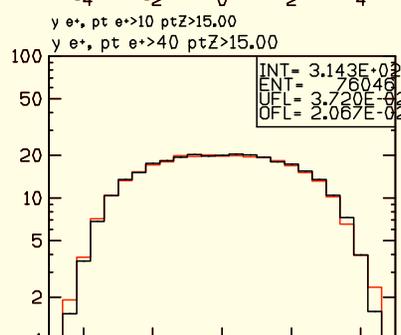
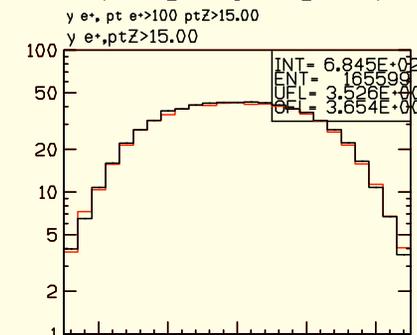
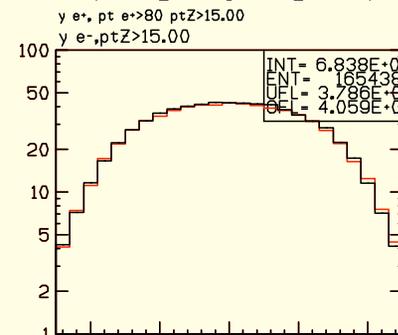
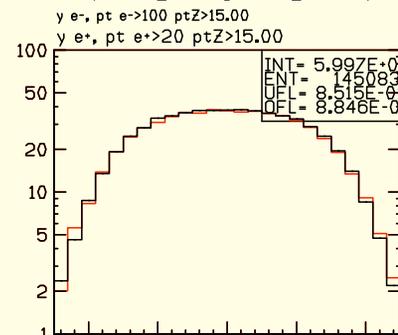
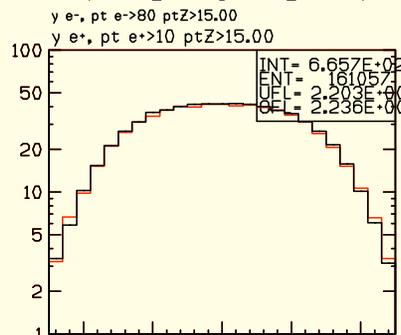
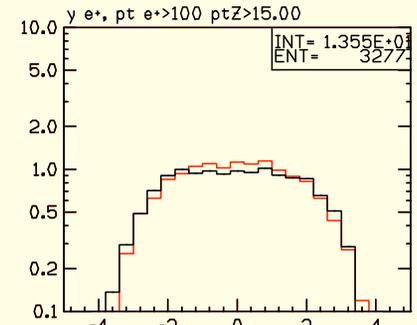
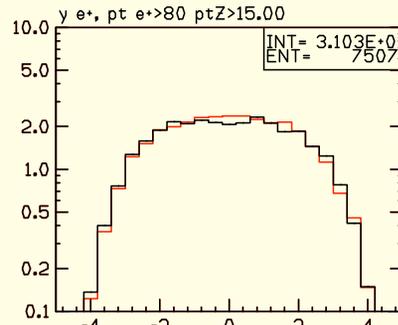
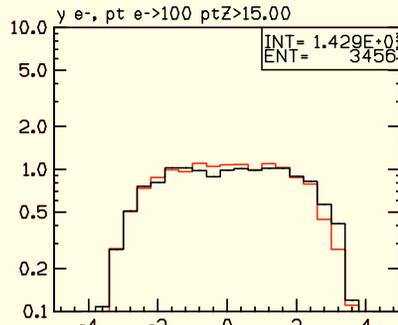
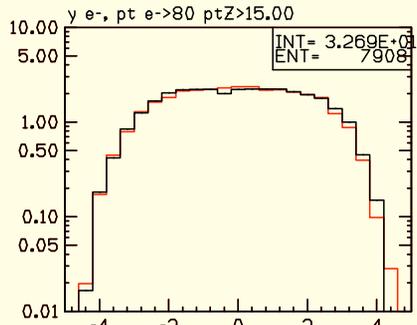
Everything seems to work ...

Now compare POWHEG+HERWIG with NLO (red)

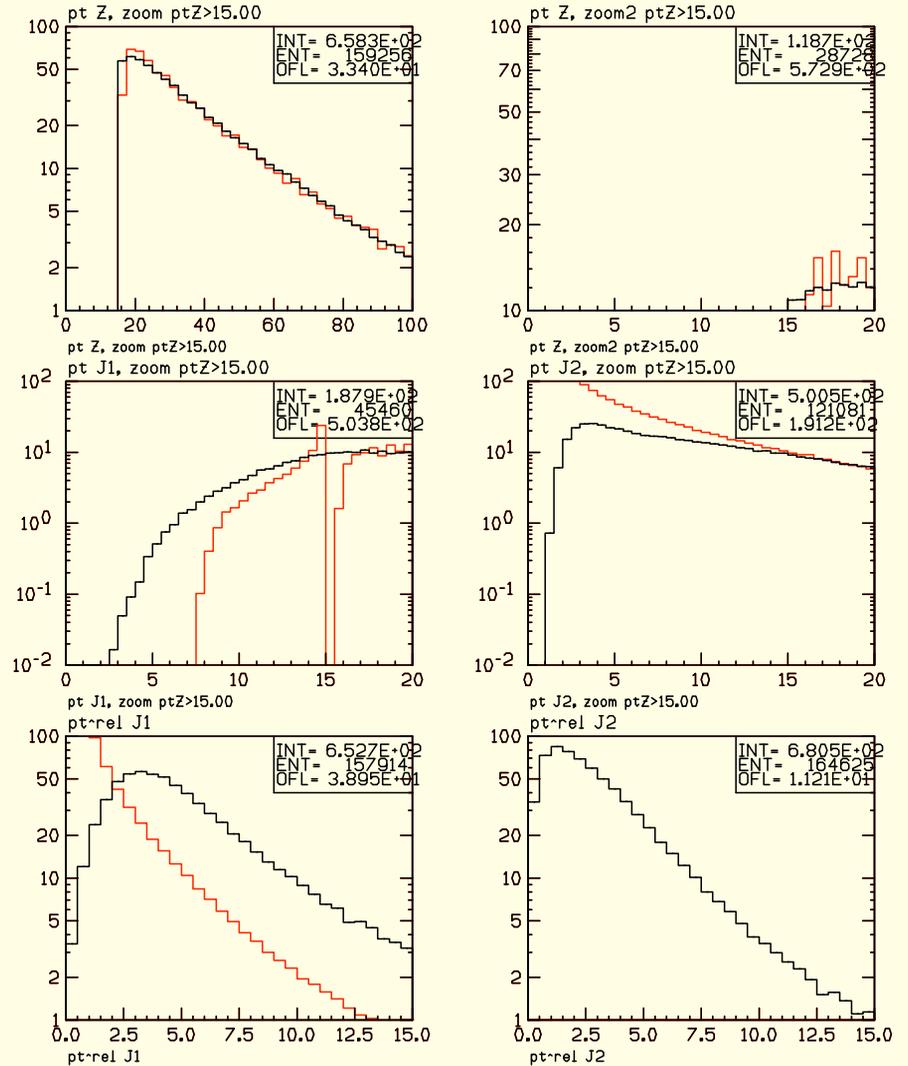








Distributions sensitive to more than two jet show noticeably different.
 All others in agreement with NLO



Summary

- A path to full automation of POWHEG implementations of arbitrary NLO calculation is open
- Soon the code will be out
- Contact us if you would like to implement a process