fastNLO

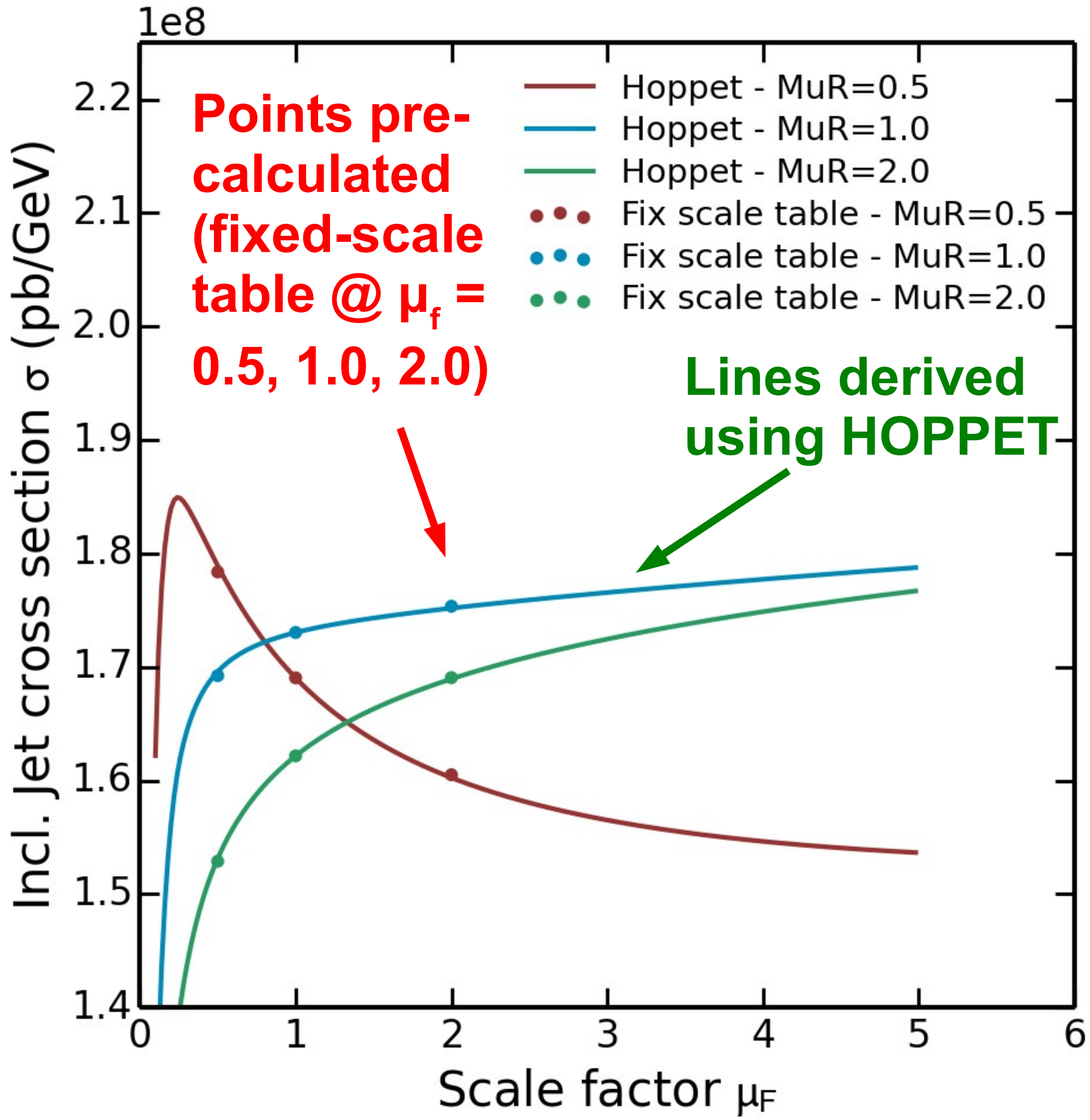
Recent Developments

Daniel Britzger, **Klaus Rabbertz,** Georg Sieber, Fred Stober, Markus Wobisch
(DESY, KIT, KIT, Uni Hamburg, Louisiana Tech University)

Karlsruhe Institute of Technology

Use of HOPPET for μ_f variation

- **fastNLO uses extra tables for μ_f variation with fixed scale factors**
  - **straightforward also @ NNLO**
  - **avoids additional integrations**
  - **increases table size**
- **In fastNLO v2.3 can also use HOPPET for μ_f variation**
  - **Continuous fast variation at NLO**
  - **Same method as used in APPLgrid**



APPLgrid, T. Carli et al., EPJC 66 (2010) 503.

Calculations with fastNLO in NNLO

- Problem
  - Scale variations become more difficult in NNLO than in NLO
- Current available implementations for NLO calculations

Renormalization scale variations

  - Scale variations applying RGE
    - Use LO matrix elements times $n\beta_0 \ln(c_r)$
  - Flexible-scale implementation
    - Store scale-independent weights:

Factorization scale variations

  - Calculate LO DGLAP splitting functions using HOPPET
  - Store coefficients for desired scale factors
  - Flexible-scale implementation

- Scale variations for NNLO calculations
  - renormalization scale variations become more complicated
  - NLO splitting functions are needed for factorization scale variations e.g. with HOPPET
    - Calculations become slower again => Not desired for fast repeated calculations

Flexible-scale tables

- Storage of scale-independent weights enable full scale flexibility also in NNLO
  - Additional logs in NNLO

$$\omega(\mu_R,\mu_F) = \underbrace{\omega_0 + \log(\mu_R^2)\omega_R + \log(\mu_F^2)\omega_F}_{\text{log's for NLO}} + \underbrace{\log^2(\mu_R^2)\omega_{RR} + \log^2(\mu_F^2)\omega_{FF} + \log(\mu_R^2)\log(\mu_F^2)\omega_{RF}}_{\text{additional log's in NNLO}}$$

  - Store weights: $w_0, w_R, w_F, w_{RR}, w_{FF}, w_{RF}$ for order α_s^{n+2} contributions

- Advantages
  - Renormalization and factorization scale can be varied *independently* and by *any* factor
    - No time-consuming 're-calculation' of splitting functions in NLO necessary
  - Only small increase in amount of stored coefficients

- fastNLO implementation
  - *Two* different observables can be used for the scales
    - e.g.: H_T and $p_{T,max}$
    - or e.g.: p_T and $|y|$
    - …
  - *Any function* of those *two observables* can be used for calculating scales

'Flexible-scale concept': Best choice for performant NNLO calculations

Flexible-scale tables in DIS

fastnlo @ HepForge

Tables from H1 multi-jet study use $\sqrt{Q^2}$ and pT

Use of this method in fastNLO dates back to 2011 when going from v1.4 to v2.1. Useful for DIS, now also for pp, e.g. with scales M_Z and pT_Z.

Note: All HERA tables are flexible-scale tables ==> The C++ reader versions must be used.

HERA: ep @ sqrt(s) = 319 GeV		
fnh5001_I1301218	H1 inclusive jet HERA-II (kt and anti-kt); LO, NLO inSPIRE HepData	no RIVET analysis available
fnh5002_I1301218	H1 dijet HERA-II (kt and anti-kt); LO, NLO inSPIRE HepData	no RIVET analysis available
fnh5003kt_I1301218	H1 trijet HERA-II (kt); LO, NLO inSPIRE HepData	no RIVET analysis available
fnh5003ak_I1301218	H1 trijet HERA-II (anti-kt); LO, NLO inSPIRE HepData	no RIVET analysis available
fnh4002_I875006	ZEUS inclusive dijet HERA-I+II (kt); LO, NLO inSPIRE no HepData (Note: This table only works with the new fastnlo_toolkit reader, but not yet with the old fastnlo_reader.)	no RIVET analysis available
fnh5201_I838435	H1 inclusive jets at low Q^2 HERA-I (kt); LO, NLO inSPIRE no HepData (Note: This table only works with the new fastnlo_toolkit reader, but not yet with the old fastnlo_reader.)	no RIVET analysis available
fnh5401_I818707	H1 inclusive jets at high Q^2 HERA-I (kt); LO, NLO inSPIRE no HepData, only normalized x section publ. (Note: This table only works with the new fastnlo_toolkit reader, but not yet with the old fastnlo_reader.)	no RIVET analysis available
fnh5101_I753951	H1 inclusive jets HERA-I (kt); LO, NLO inSPIRE HepData (Note: This table only works with the new fastnlo_toolkit reader, but not yet with the old fastnlo_reader.)	no RIVET analysis available
fnh4401_I724050	ZEUS inclusive jets HERA-I (kt); LO, NLO inSPIRE HepData (Note: This table only works with the new fastnlo_toolkit reader, but not yet with the old fastnlo_reader.)	no RIVET analysis available
HERA: ep @ sqrt(s) = 300 GeV		
fnh4301_I593409	ZEUS inclusive jets HERA (kt); LO, NLO inSPIRE HepData (Note: This table only works with the new fastnlo_toolkit reader, but not yet with the old fastnlo_reader.)	no RIVET analysis available

Flexible-scale tables in DIS

fastnlo @ HepForge

Tables from H1 multi-jet study use $\sqrt{Q^2}$ and pT

Use of this method in fastNLO dates back to 2011 when going from v1.4 to v2.1. Useful for DIS, now also for pp, e.g. with scales M_Z and pT_Z.

Note: All HERA tables are flexible-scale tables ==> The C++ reader versions must be used.

HERA: ep @ sqrt(s) = 319 GeV			
fnh5001_I1301218	H1 inclusive jet HERA-II (kt and anti-kt); LO, NLO		
	inSPIRE	HepData	no RIVET analysis available
fnh5002_I1301218	H1 dijet HERA-II (kt and anti-kt); LO, NLO		
	inSPIRE	HepData	no RIVET analysis available
fnh5003kt_I1301218	H1 trijet HERA-II (kt); LO, NLO		
	inSPIRE	HepData	no RIVET analysis available
fnh5003ak_I1301218	H1 trijet HERA-II (anti-kt); L		
	inSPIRE	HepData	no RIVET analysis available
fnh4002_I875006	ZEUS inclusive dijet HERA-		
	inSPIRE	no HepData	no RIVET analysis available
	(Note: This table only works with the new fastnlo_toolkit reader, but not yet with the old fastnlo_reader.)		
fnh5201_I838435	H1 inclusive jets at low Q^2 HERA-I (kt); LO, NLO		
	inSPIRE	no HepData	no RIVET analysis available
	(Note: This table only works with the new fastnlo_toolkit reader, but not yet with the old fastnlo_reader.)		
fnh5401_I818707	H1 inclusive jets at high Q^2 HERA-I (kt); LO, NLO		
	inSPIRE	no HepData, only normalized x section publ.	no RIVET analysis available
	(Note: This table only works with the new fastnlo_toolkit reader, but not yet with the old fastnlo_reader.)		
fnh5101_I753951	H1 inclusive jets HERA-I (kt); LO, NLO		
	inSPIRE	HepData	no RIVET analysis available
	(Note: This table only works with the new fastnlo_toolkit reader, but not yet with the old fastnlo_reader.)		
fnh4401_I724050	ZEUS inclusive jets HERA-I (kt); LO, NLO		
	inSPIRE	HepData	no RIVET analysis available
	(Note: This table only works with the new fastnlo_toolkit reader, but not yet with the old fastnlo_reader.)		
HERA: ep @ sqrt(s) = 300 GeV			
fnh4301_I593409	ZEUS inclusive jets HERA (kt); LO, NLO		
	inSPIRE	HepData	no RIVET analysis available
	(Note: This table only works with the new fastnlo_toolkit reader, but not yet with the old fastnlo_reader.)		

Click

Tables in Durham HepData

Following a discussion I had with Frank Krauss and a follow-up at the Benasque PDF Workshop, tables can now be stored with data

The Durham HepData Project

Durham University

REACTION DATABASE • DATA REVIEWS • PDF PLOTTER   ABOUT HEPDATA • SUBMITTING DATA

Reaction Database Full Record Display

View short record or as: input, plain text, AIDA, PyROOT, YODA, ROOT, mpl, ScaVis or MarcXML

ANDREEV 2014 — Measurement of Multijet Production in ep Collisions at High Q^2 and Determination of the Strong Coupling alpha_s

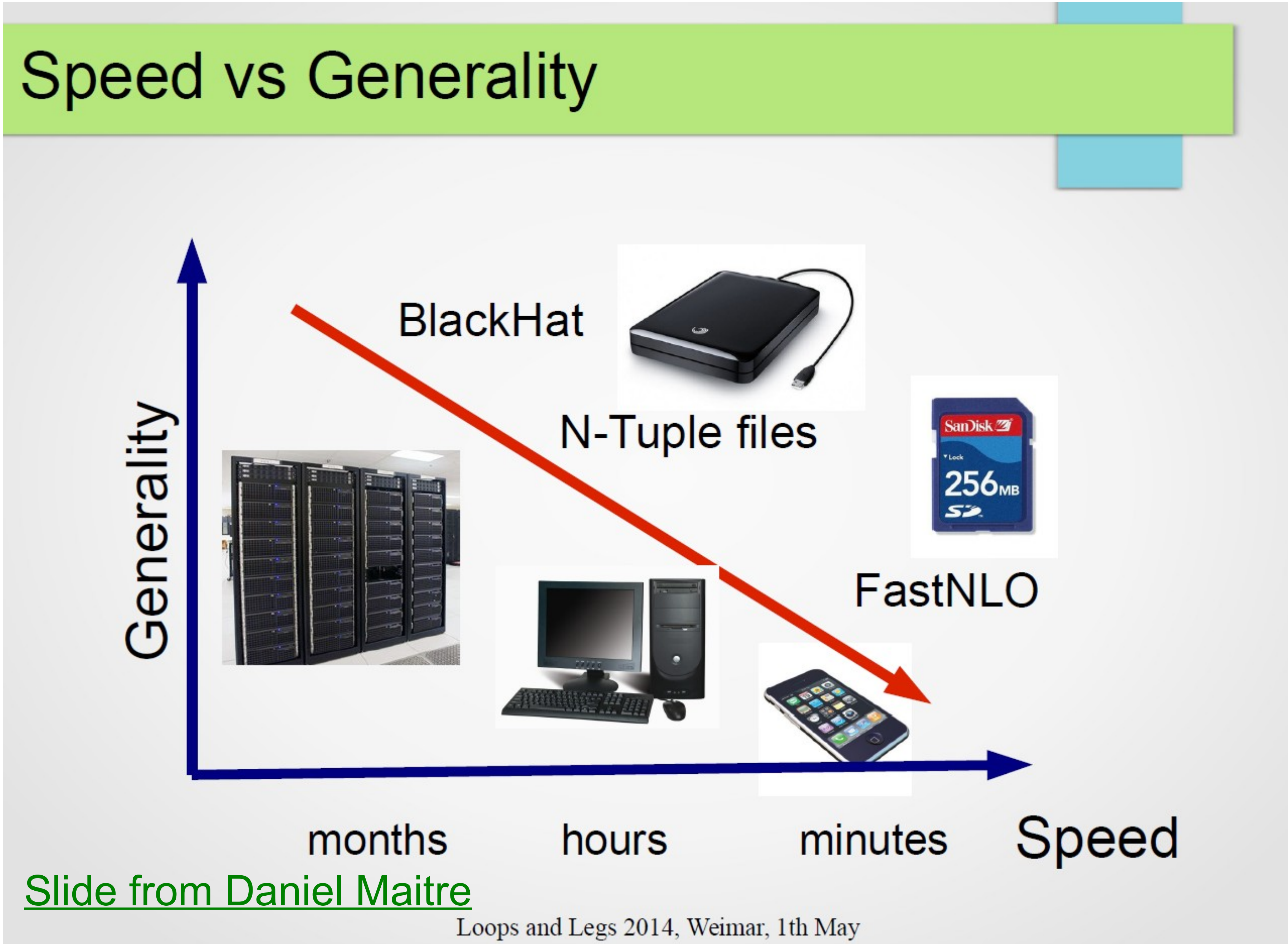
Experiment: **DESY-HERA-H1 (H1)**
Preprinted as **DESY-14-089**
Archived as: **ARXIV:1406.4709**
Record in: **INSPIRE**
Record in: **CERN Document Server**

...

Link to fastNLO v2.1 table (inclusive jet, kT and anti-kT)
Link to fastNLO v2.1 table (dijet, kT and anti-kT)
Link to fastNLO v2.1 table (trijet, kT)
Link to fastNLO v2.1 table (trijet, anti-kT)

Thanks to Graeme Watt

Use with BlackHat N-Tuples

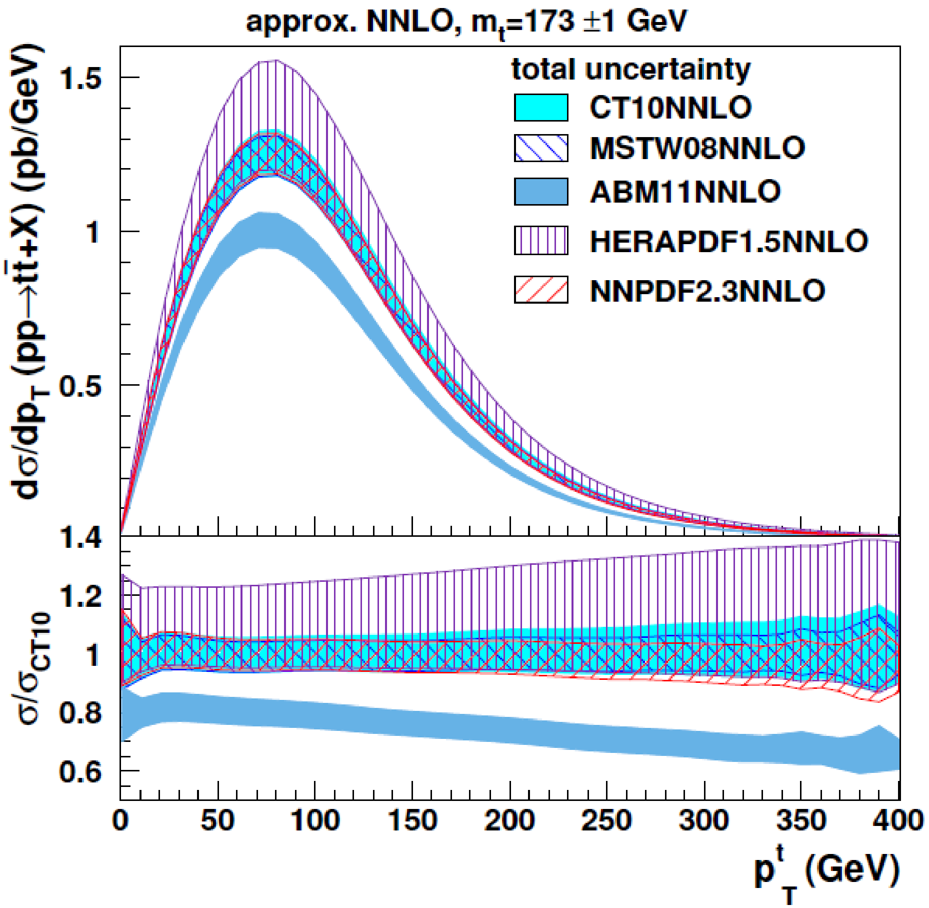


Slide from Daniel Maitre

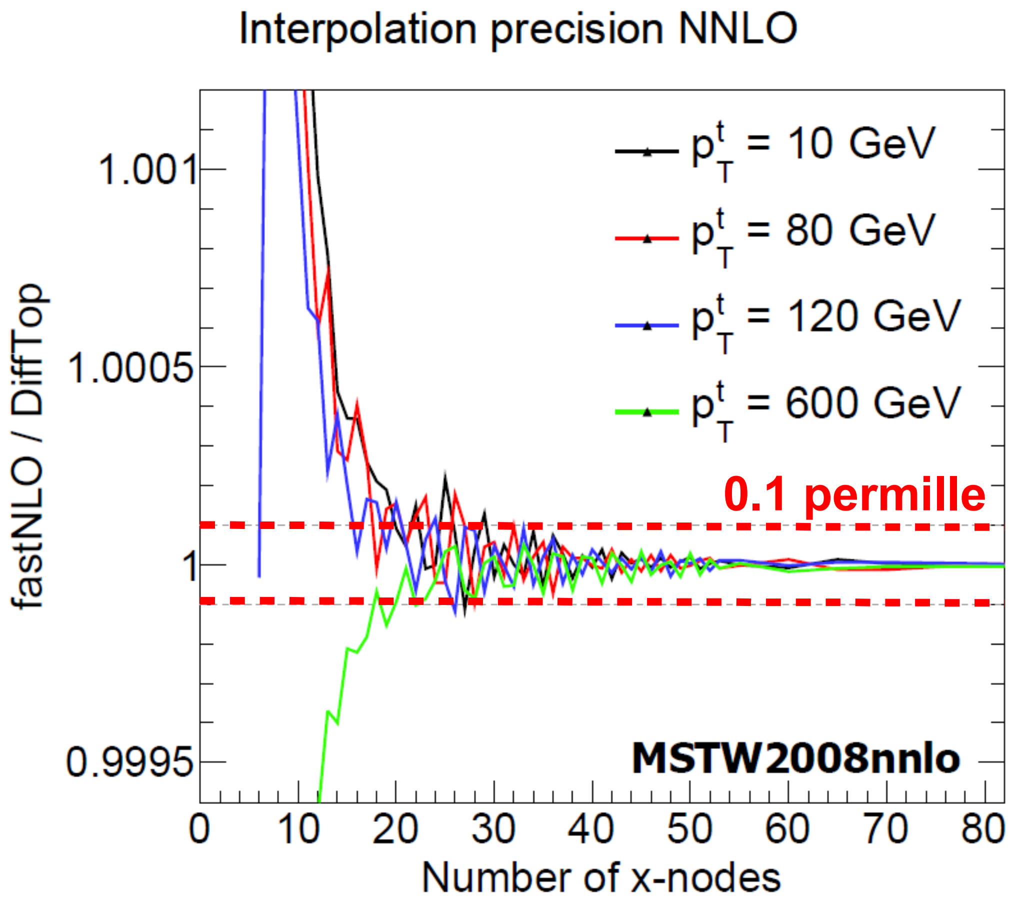
Use with DiffTop

Differential ttbar in approx. NNLO: $d\sigma/dp_T$, $d\sigma/dy$

(total uncertainty: quadr. sum of PDF, scale, α_s, m_t variations)



Precision study of fastNLO tables over DiffTop standalone vs. no. of x nodes



786 repeated calculations needed including (separate) variation of m_t

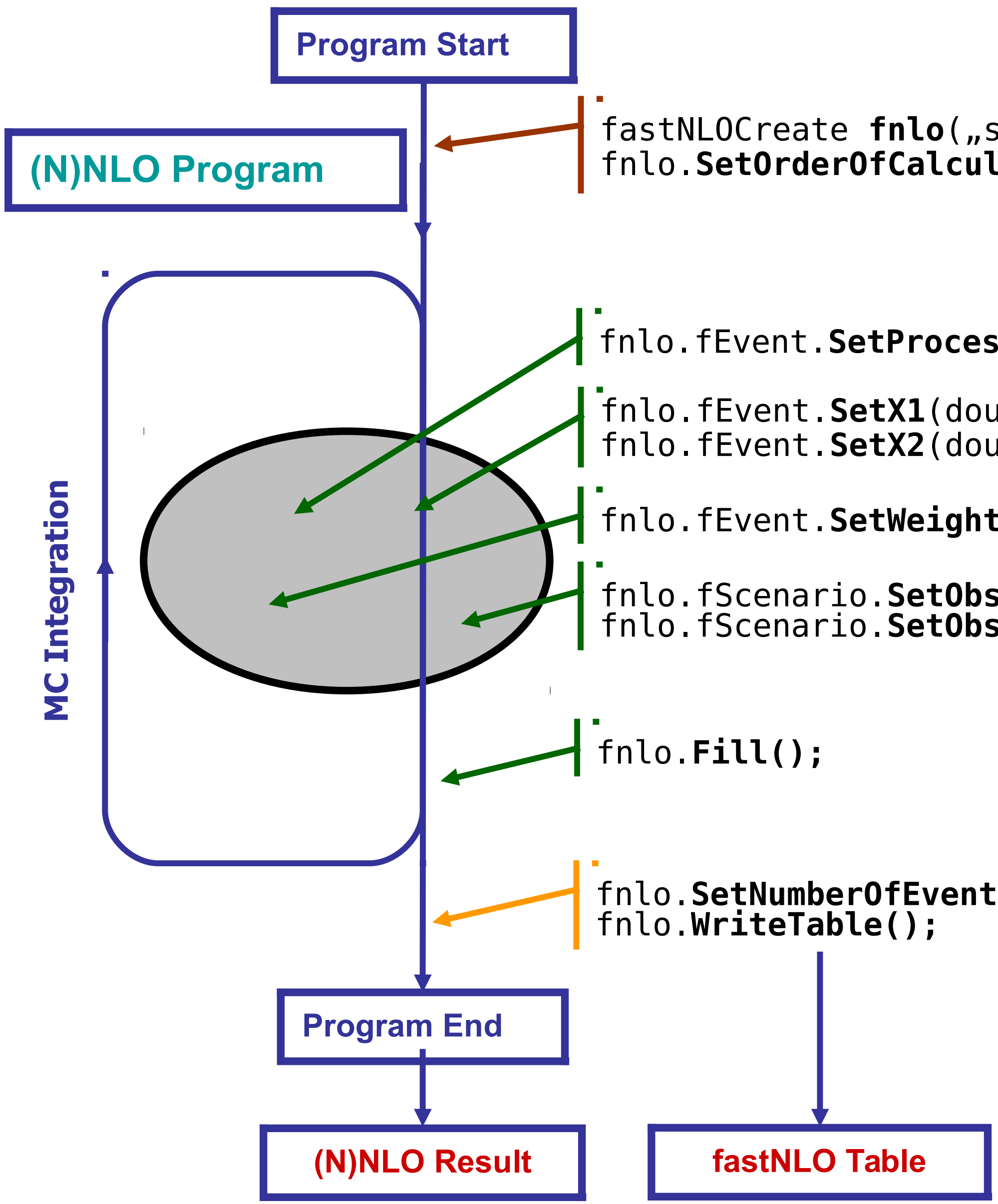
DiffTop, M. Guzzi et al., JHEP01, 2015.

Perfect agreement for probed x-range of $2 \cdot 10^{-3} < x < 1$

Developments

- 1. Prepared Toolkit library for creating & evaluating fastNLO interpolation tables
  - Independent of any generator
- 2. Facilitated use with extensible steering files
- 3. Being asked at DIS we put together an example of Fortran-based access to the C++ library
- 4. Interface even more theory programs …
  - NLO for higher multiplicities
  - NNLO e.g. Z+jet, jets
  - ...

Simple example for use of Toolkit




```
fastNLOCreate fnlo(„steering.str“);
fnlo.SetOrderOfCalculation(int order);
```

Initialize fastNLO class(es)

```
fnlo.fEvent.SetProcessID(int id);
fnlo.fEvent.SetX1(double x1);
fnlo.fEvent.SetX2(double x2);
fnlo.fEvent.SetWeight(double w);
fnlo.fScenario.SetObservable0(double pt);
fnlo.fScenario.SetObsScale1(double s1);
```

Pass the process specific variables during the 'event loop' to fastNLO
- **Order does not matter**
- **Many other convenient implementations possible**

```
fnlo.Fill();
```

Pass all information to fastNLO

```
fnlo.SetNumberOfEvents(double nevents);
fnlo.WriteTable();
```

Set normalization of the MC integration and write table

Minimum implementation: 11 lines of code

Convenient implementation of fastNLO into any (N)NLO program possible!

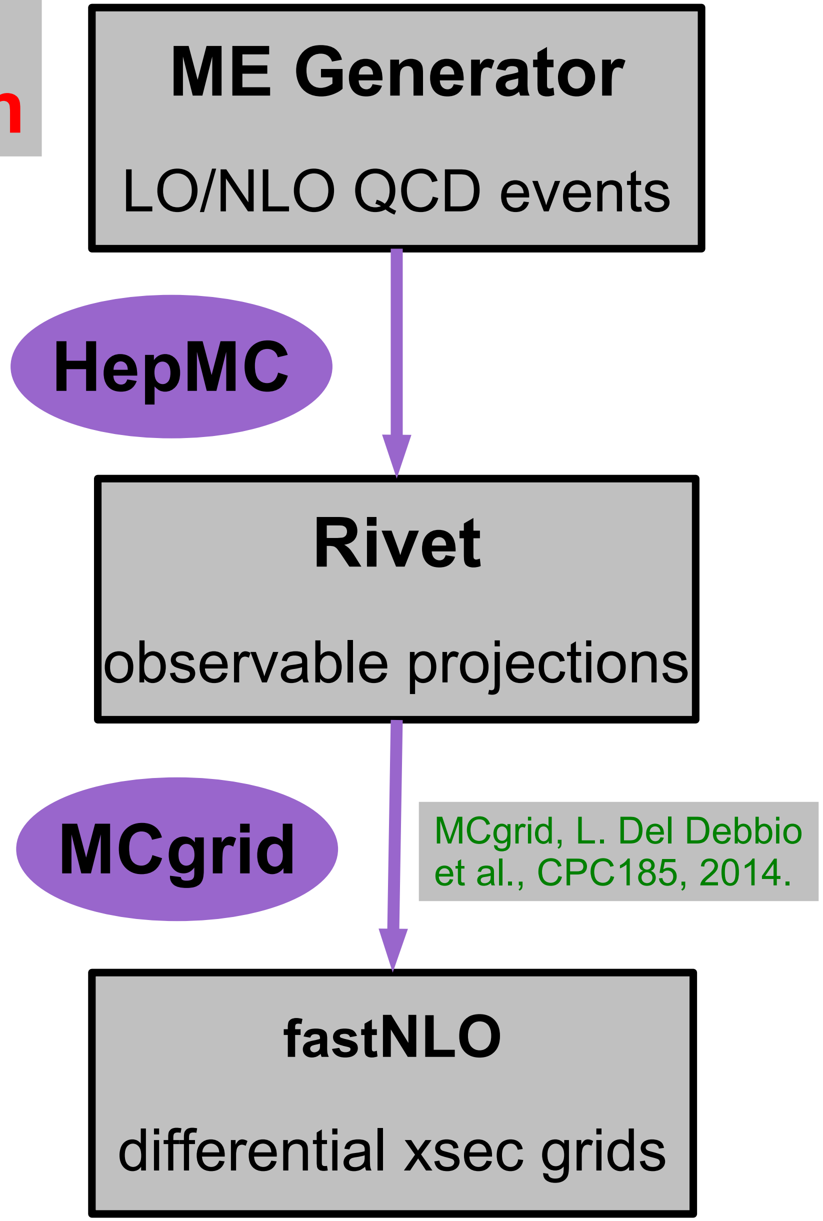
Use with Sherpa2 & MCgrid2

In collaboration with Enrico Bothmann & Steffen Schumann

fastNLO Toolkit access implemented:

- **Events generated with Sherpa 2.1.1/2.2.0**
- **Two analyses from Rivet 2.2.0 tested**
- **MCgrid 2.0 for cross section projection into grids (to be released)**
- **Same toolkit functions accessed either via direct calls from MCgrid-enabled Rivet analysis or via steering file**
- **Usable with large number of processes available via Sherpa and one-loop generators like ...**

Sherpa, T. Gleisberg et al., JHEP02, 2004; JHEP02, 2009.
BlackHat, C.F. Berger et al., PRD78, 2008.
GoSam, G. Cullen et al., EPJC72, 2012.
OpenLoops, F. Cascioli et al., PRL108, 2012.
NJET, S. Badger et al., CPC184, 2013.
...



Snippets of Rivet+MCgrid analysis

```
#include "Rivet/Analysis.hh"
#include "mcgrid/mcgrid.hh"
...
namespace Rivet {

  /// CDF Z boson rapidity modified to generate grid files
  class MCgrid_CDF_2009_S8383952 : public Analysis {
  public:
...
using namespace MCgrid;
Histo1DPtr _hist_yZ;  // Rivet histogram
gridPtr _grid_yZ;     // Corresponding grid

// Init phase
subprocessConfig subproc("DY-ppbar.str", BEAM_PROTON, BEAM_ANTIPROTON);
fastnloGridArch arch(50, 1, "Lagrange", "OneNode", "sqrtlog10", "linear");
fastnloConfig config(0, subproc, arch, 1960.0);
_hist_yZ = bookHisto1D(2, 1, 1);                    // Book Rivet
_grid_yZ = bookGrid(_hist_yZ, histoDir(), config);  // Book MCgrid/fastNLO

// Analyse phase
PDFHandler::HandleEvent(event, histoDir());  // Update subprocess statistics
_hist_yZ->fill(yZ, weight);                  // Fill Rivet
_grid_yZ->fill(yZ, event);                   // Fill MCgrid/fastNLO

// Finalise phase
scale(_hist_yZ, normalisation);              // Scale Rivet
_grid_yZ->scale(normalisation);              // Scale MCgrid/fastNLO
PDFHandler::CheckOutAnalysis(histoDir());    // Finalise
```

Setup Rivet with MCgrid

Book & config grid and histos

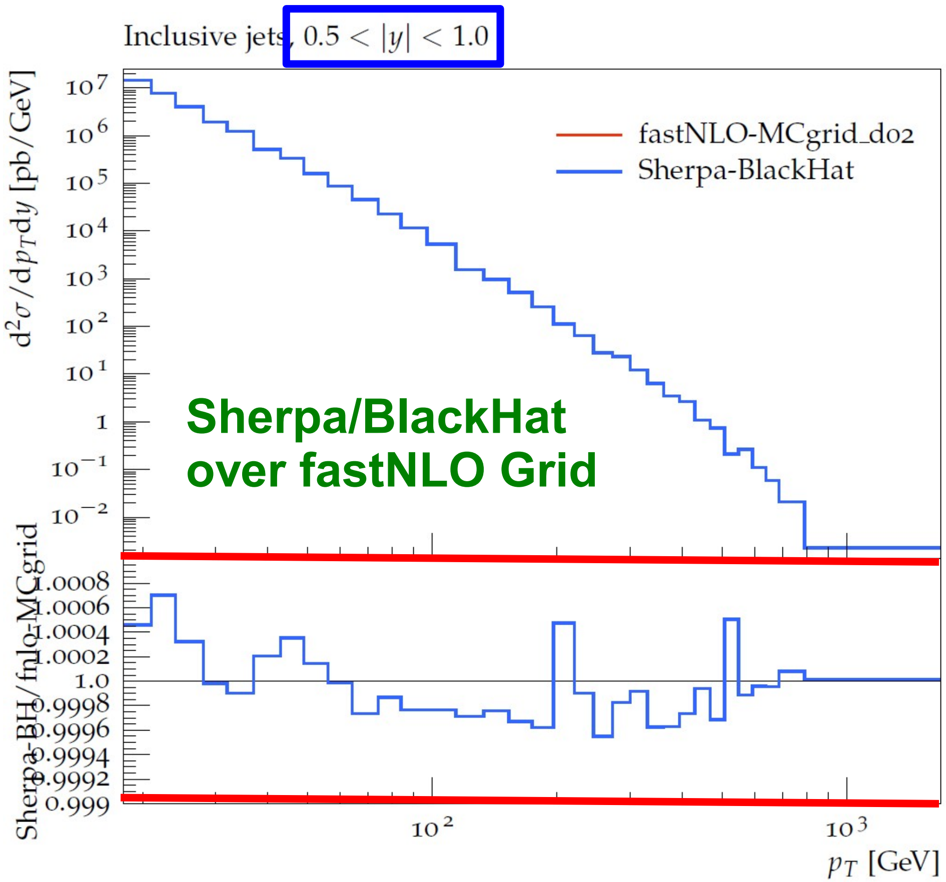
Fill events in event loop.

Final check out, normalize, write table.

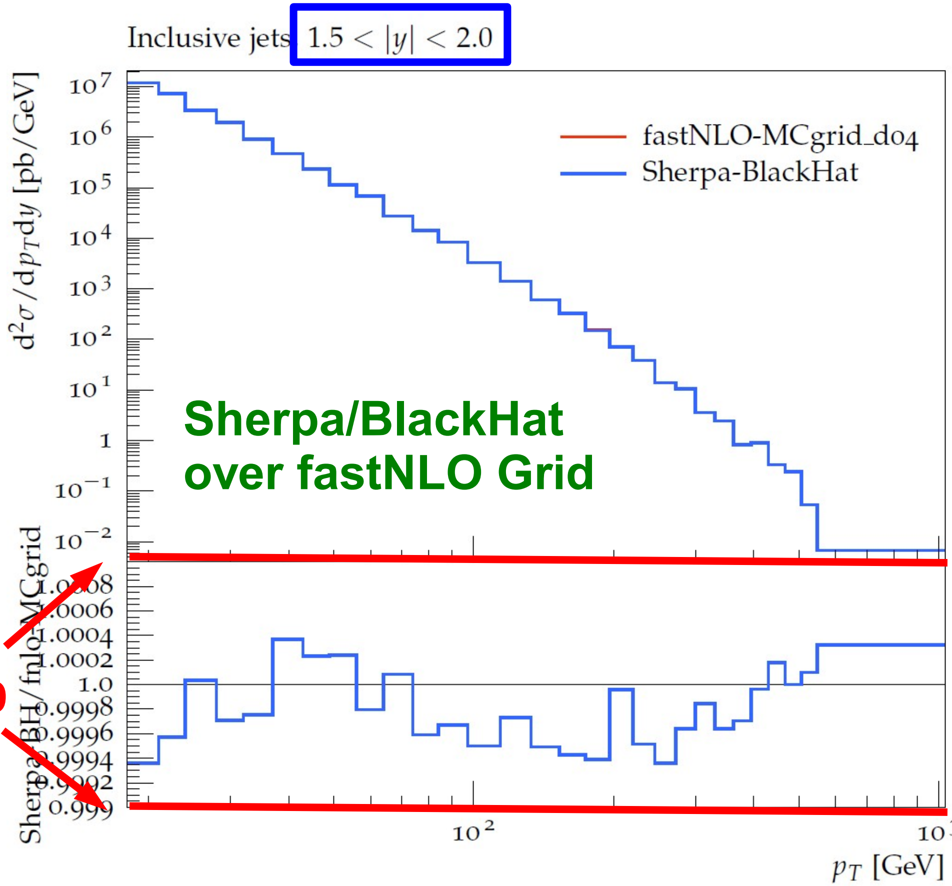
Test with inclusive Jets

Previously:
Drell-Yan Z rapidity @ Tevatron
- 1M (phase space) / 10M (fill) events
- Constant scale → interpolation in x only
- Agreement at sub-permille level

NEW HERE:
Inclusive Jets @ LHC
- 100M (NLOJet++ ph. sp.) / 4M (fill) events
- Dynamic scale → interpolation in x & Q
- Problem in interface MCgrid-fastNLO fixed
- Agreement at sub-permille level



±1‰

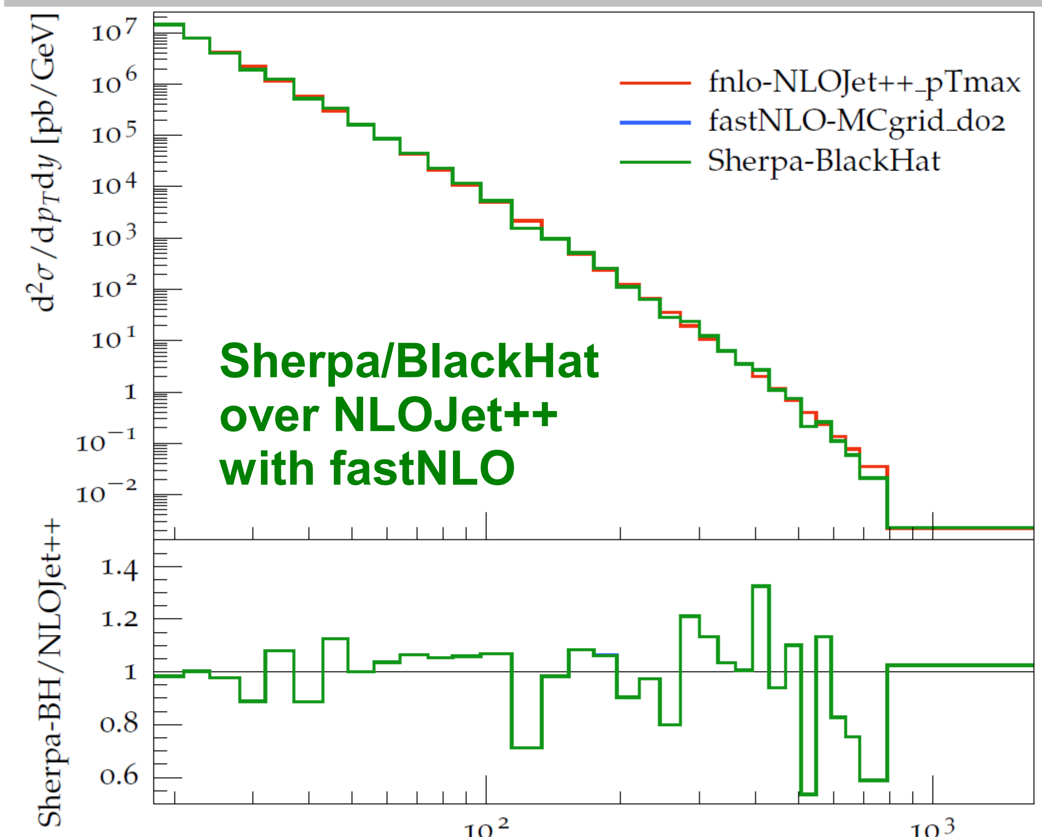


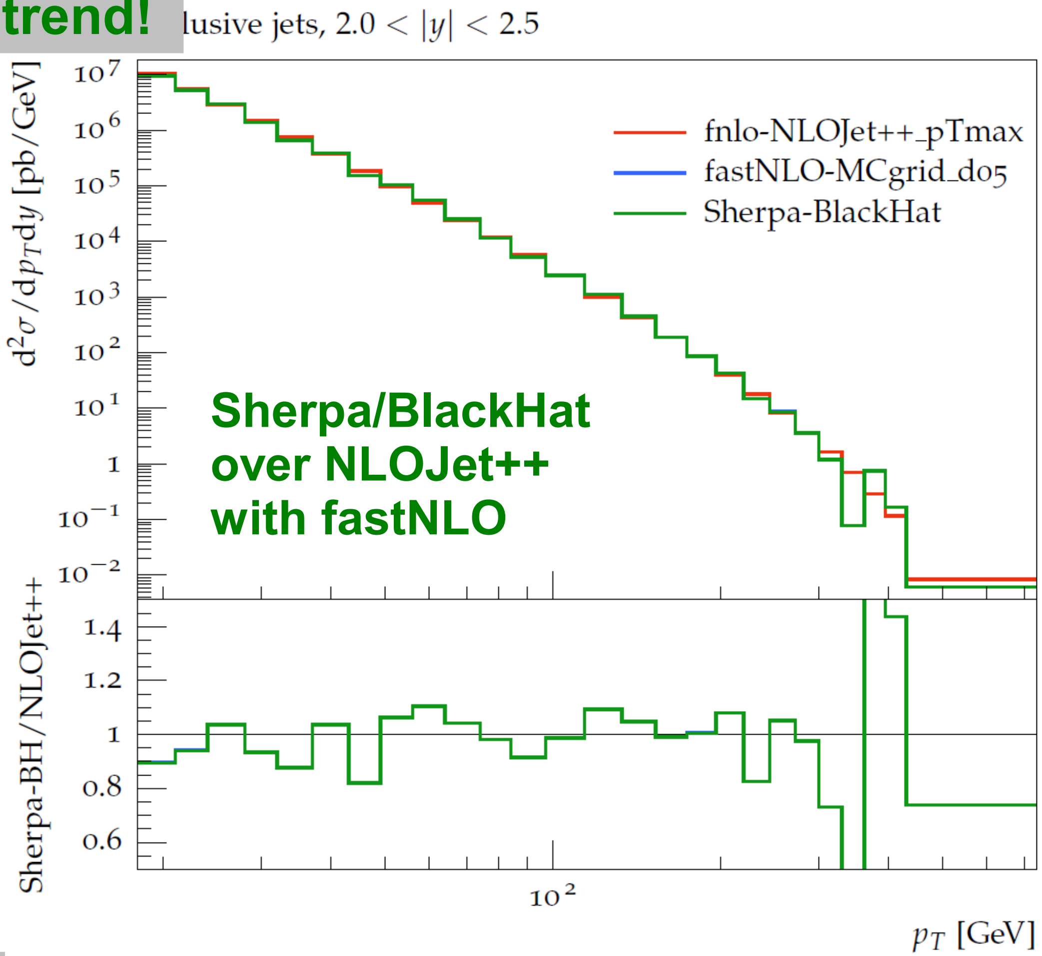
Comparison to NLOJet++

Sherpa/BlackHat inclusive Jets:
- 100M (NLOJet++ ph. sp.) / **4M fill events**
- ~ only some hours on 4 cores of my Laptop!

NLOJet++ inclusive Jets:
- 100M (phase space) / **4G fill events**
- ~ 3000h on cluster using 400 nodes
- **Large stat. fluctuations, but no apparent trend!**

Note: Both calculations use an **event-wise** dynamical scale, pTmax, **jet-wise** scales not possible currently with Sherpa-MCgrid

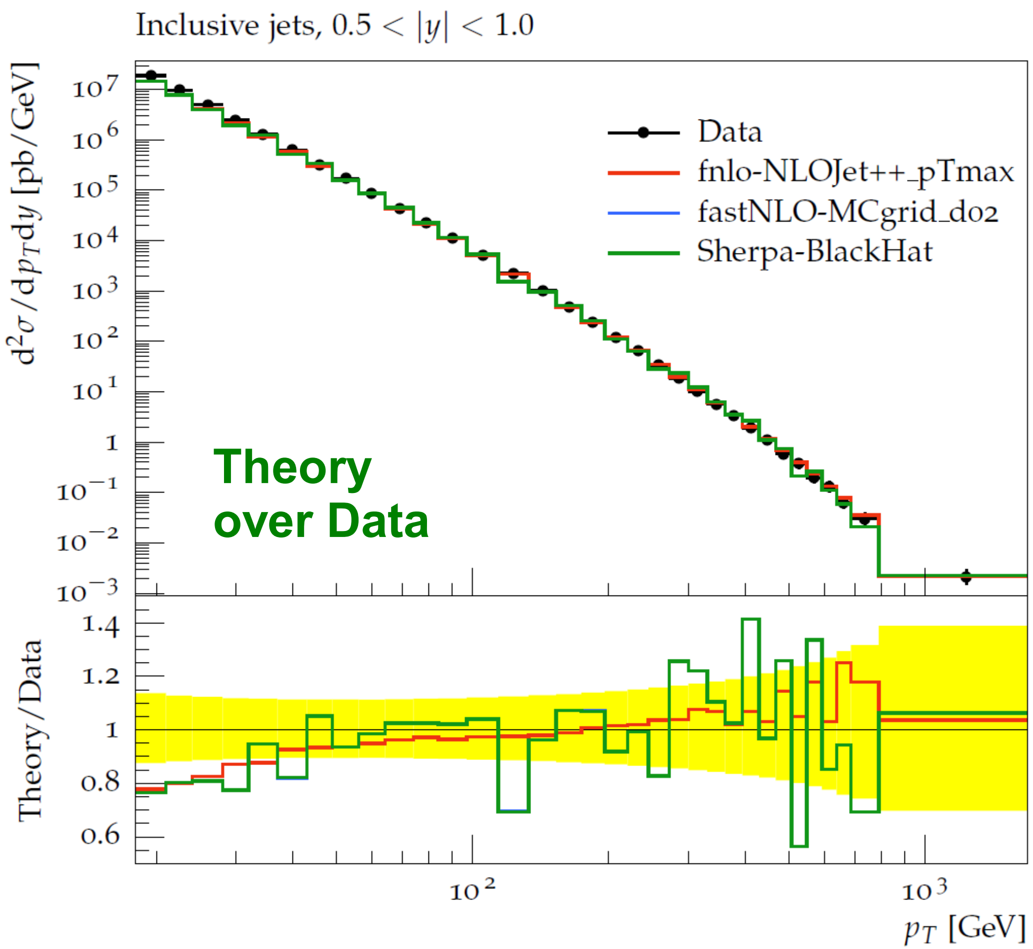


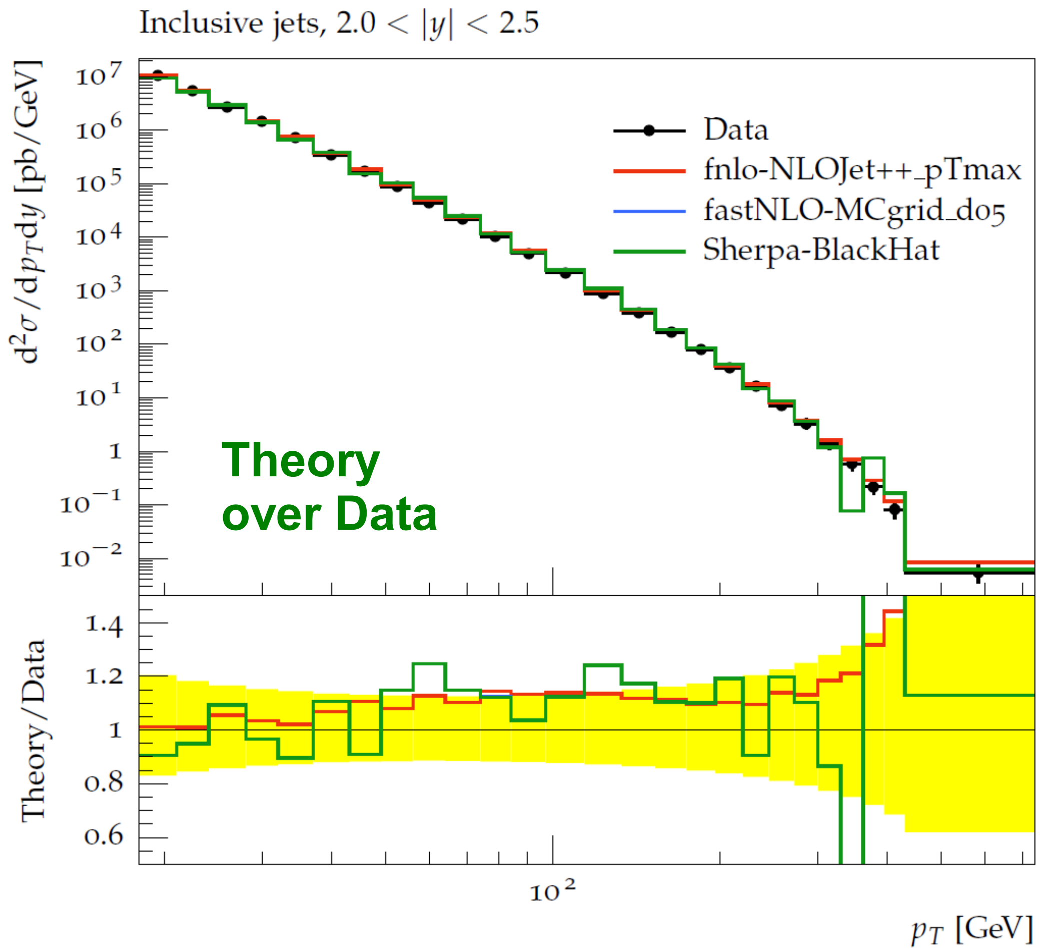


NLOJet++, Z.Nagy, PRD68 2003, PRL88 2002

Comparison to Data

Agreement with NLOJet++ and Data
except at low pT since NP corrections not included here
(Stat. fluctuations still there, of course.)





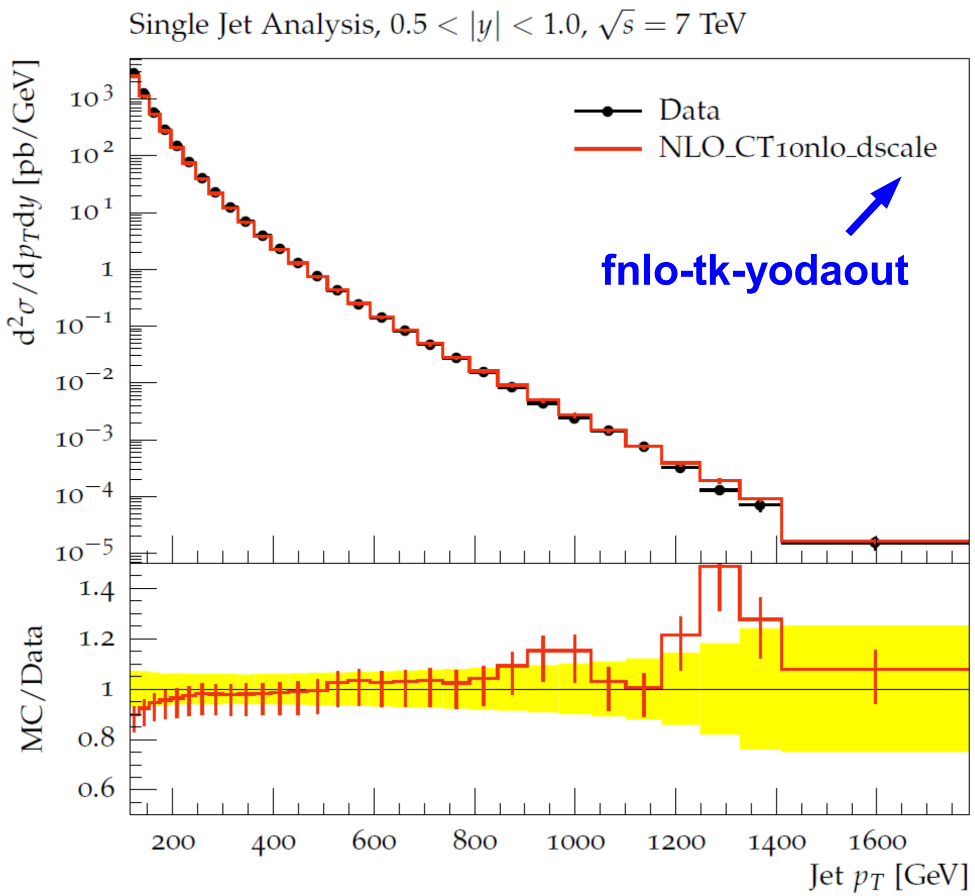
Use with Rivet 2 & YODA Format

Started as a CERN Summer student (S. Tyros)project last year with Peter Skands.

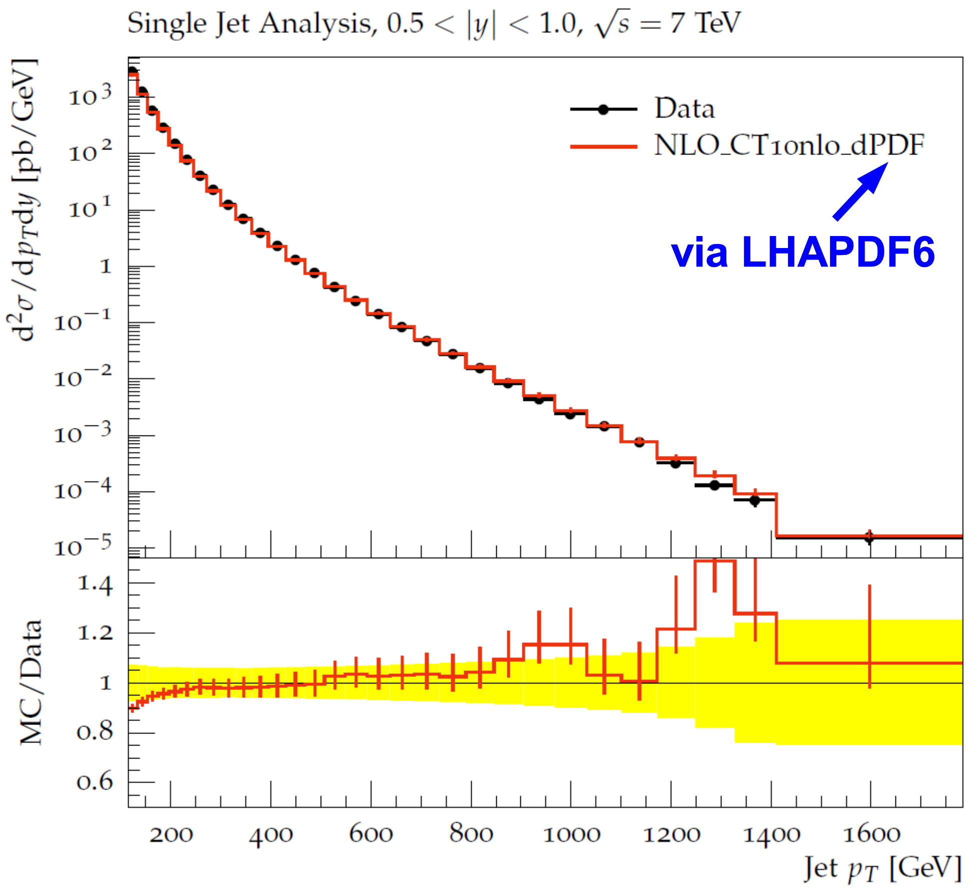
RIVET, A. Buckley et al., CPC184 (2013), rivet.hepforge.org, yoda.hepforge.org.

Can be used to provide NLO histograms with uncertainty to MCPLOTS web site.

NLO with scale uncertainty / Data



NLO with PDF uncertainty / Data



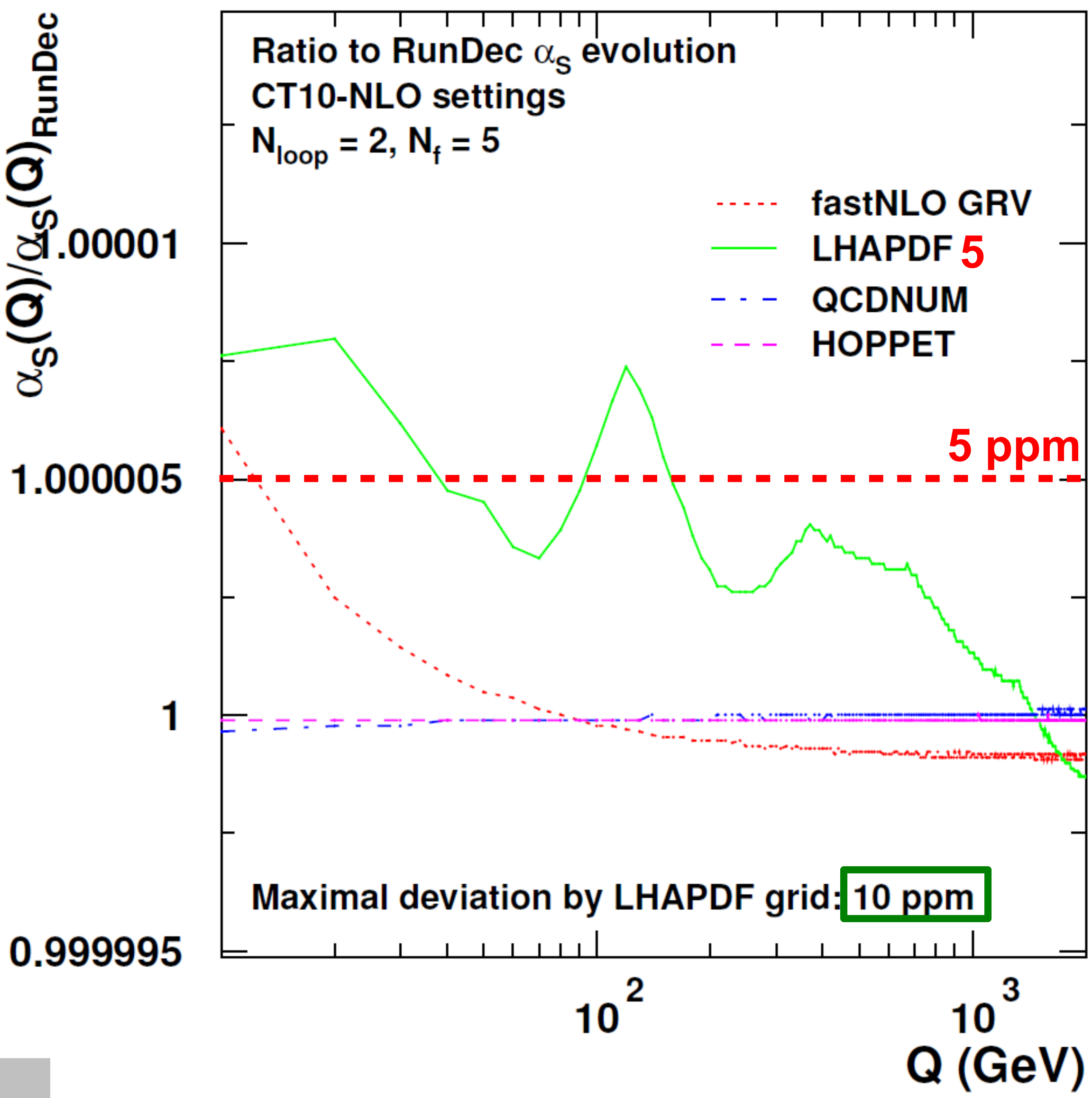
Outlook

- The toolkit provides simple access to full capability of fastNLO
- Creating, filling, reading, and evaluating fast interpolation tables in the fastNLO format
- A simplified interface to NLOJet++ is publically available
- Flexible-scale table format ideally suited for NNLO
- Tested at (approx.) NNLO with DiffTop and by BlackHat
==> first applications @ NNLO
- Other theory programs can be/have been interfaced
- Demonstrated new application with MCgrid and Sherpa
- Will be synchronized with new release of MCgrid
- Progress with further theory interfaces …!

Backup Slides

Use of alternative α_s evolutions

- LHAPDF5/6
- CRunDec 08/2012
  - included in fastNLO
- QCDNUM v17-00-06
  - … `[--with-qcdnum=/path/...]`
  - `Makefiles adapted, need -fPIC on x86_64 systems`
- HOPPET v1.1.5
  - … `[--with-hoppet=/path/...]`



RunDec, B. Schmidt, M. Steinhauser, CPC183, 2012;
K. Chetyrkin, J. Kühn, M. Steinhauser, CPC133, 2000.
QCDNUM, M. Botje, CPC182, 2011.
HOPPET, G. Salam, J. Rojo, CPC180, 2009.

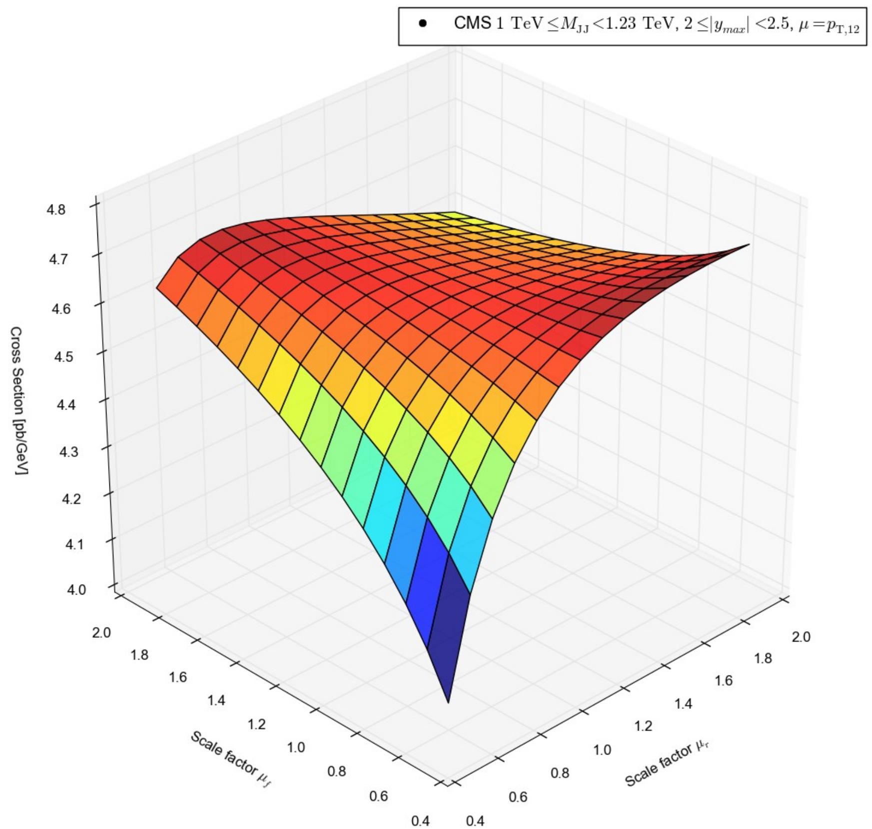
Excerpt of steering.str

```
ScenarioName fnl2342b_I902309_v23_flex          # Name and describe scenario
ScenarioDescription {
    "d2sigma-jet_dpT_dy_[pb_GeV]"
…
JetAlgo                          2               # fastjet jet algorithm: 0,1,2=kT,CA,anti-kT
Rjet                             0.5             # Jet size parameter: Required for all jets
ptjmin                          18.              # Minimal jet pT
yjmin                            0.0             # Minimal jet rapidity
yjmax                            3.0             # Maximal jet rapidity
… extensible
LeadingOrder                       2             # Number of jets for the LO process
DifferentialDimension              2             # Dimensionality of binning
DimensionLabels {                                # Labels (symbol and unit) for dimensions
   "|y|"                                         # Defines the observables to be calculated!
   "pT_[GeV]"
}

FlexibleScaleTable               true            # Create table fully flexible in mu_f
ScaleDescriptionScale1           "pT_jet_[GeV]" # This defines the scale to be used
ScaleDescriptionScale2           "pT_max_[GeV]" # Specify 2nd scale name and unit

DoubleDifferentialBinning {{
  1stDimLo    1stDimUp    "----- Array of bin-grid for 2nd dimension -----"
    0.0         0.5          18.   21.   24.   28.   32.   37.   43.   49.   56. …
…
}}
```

Running any other scenario can be as simple as adapting some kinematical cuts & binning, often not even a recompile necessary!

Demo plot using Python extension

- Python extension available
  - … [--enable-pyext]
- Easy example plotting 2D scale dependence:




```
#! /usr/bin/env python2

from fastnlo import fastNLOLHAPDF

import matplotlib
import matplotlib.pyplot as plt
from matplotlib import cm
from mpl_toolkits.mplot3d import axes3d
import numpy as np

fnlo = fastNLOLHAPDF('fnlotable.tab')
fnlo.SetLHAPDFFilename('CT10nlo.LHgrid')
fnlo.SetLHAPDFMember(0)

mufs = np.arange(0.1, 1.5, 0.10)
murs = np.arange(0.1, 1.5, 0.10)
xs = np.zeros((mufs.size, murs.size))

for i, muf in enumerate(mufs):
    for j, mur in enumerate(murs):
        fnlo.SetScaleFactorsMuRMuF(mur, muf)
        fnlo.CalcCrossSection()
        xs[i][j] = np.array(fnlo.GetCrossSection())[0]

fig = plt.figure(figsize=(13,13))

… plotting details
ax.set_ylabel('Scale factor $\mu_F$')
ax.set_xlabel('Scale factor $\mu_R$')
ax.set_zlabel('Cross Section [pb/GeV]')
plt.show()
… plotting details
```

Setup Python with fastNLO

Select table, PDF & mem.

Define μ_r, μ_f ranges

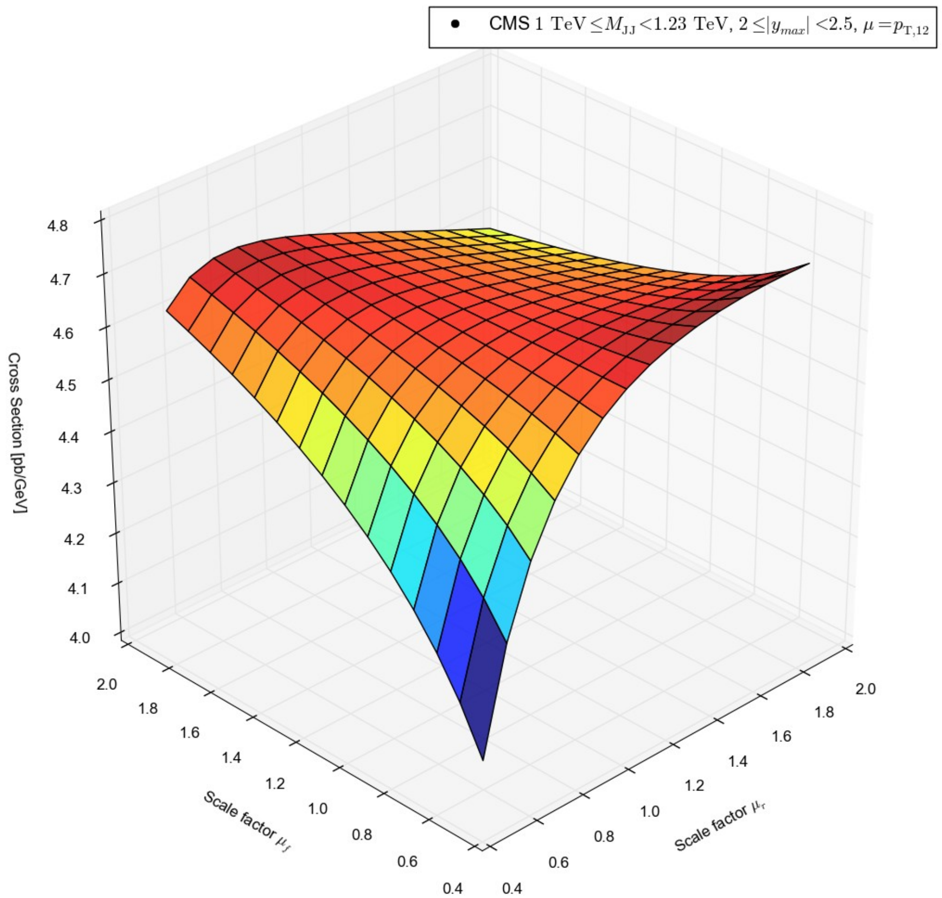
Loop over μ_r, μ_f

Plot

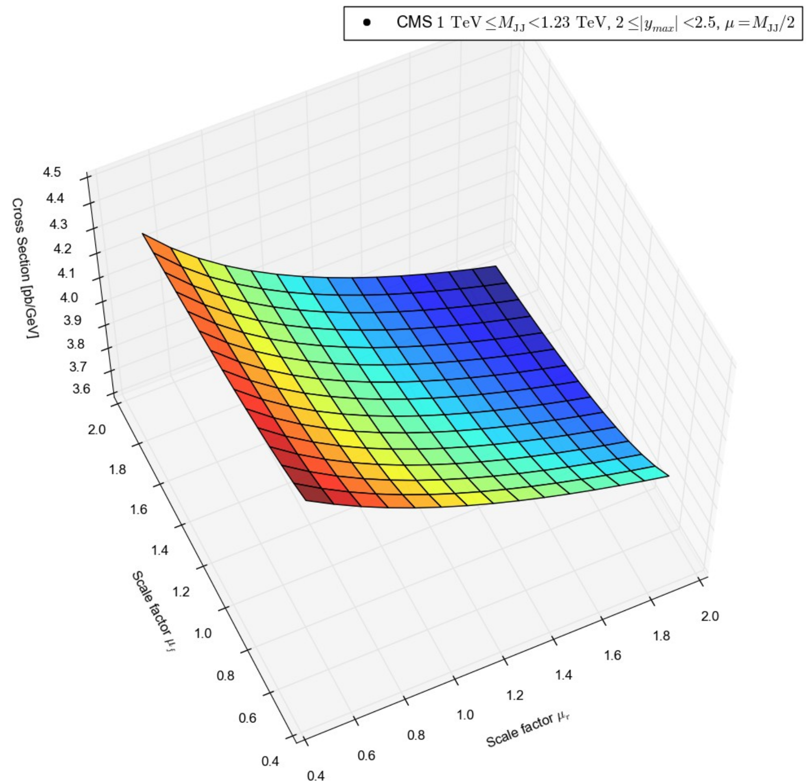
Extra slide: CMS dijet mass

Central scale: μ = <pT$_{1,2}$>

Outer |y$_{max}$| bin!



Central scale: μ = M$_{JJ}$/2



Derived from one fastNLO flexible-scale table

Extra slide: ATLAS dijet mass

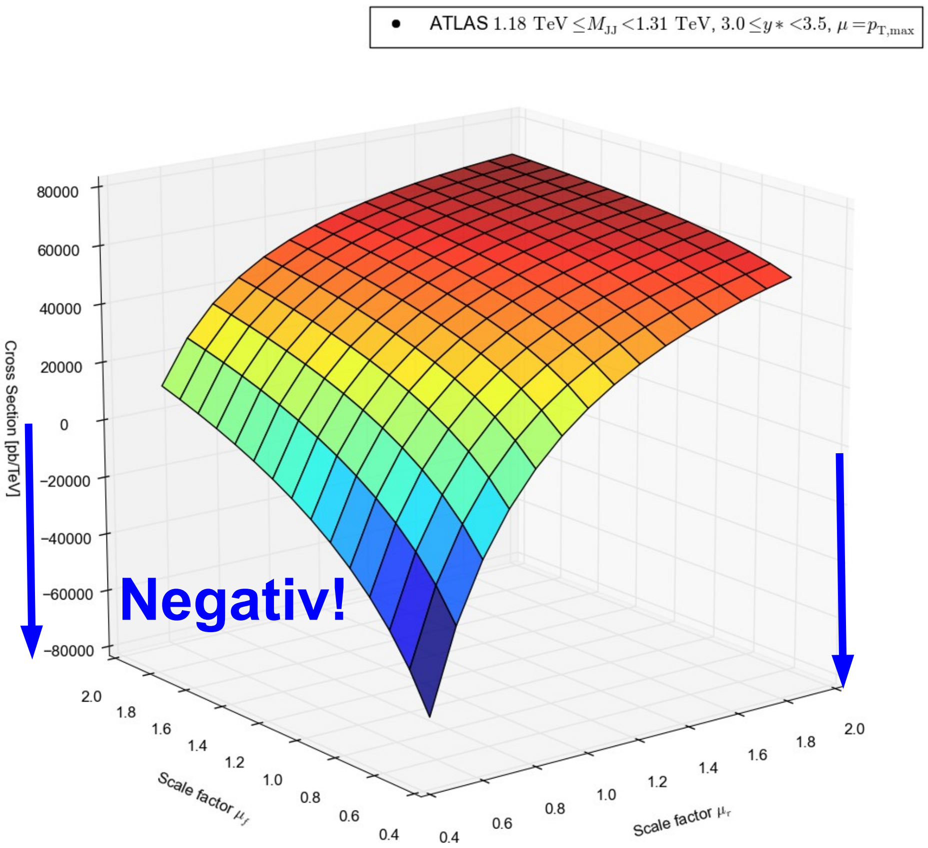
Central scale: μ = pT_{max}

Outer y* bin!

+80k

-80k

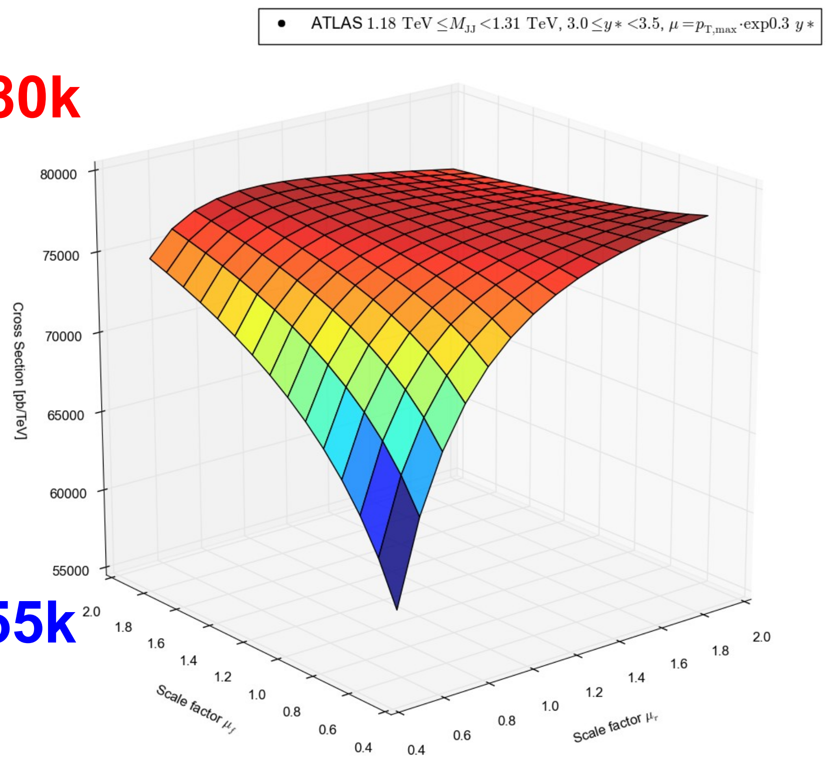
Negativ!



Central scale: μ = pT_{max}· exp(0.3 y*)

+80k

+55k



Derived from one fastNLO flexible-scale table