Extension of fastNLO to arbitrary processes

Daniel Britzger, **Klaus Rabbertz,** Georg Sieber, Fred Stober,Markus Wobisch
(DESY, KIT * 3, Louisiana Tech University)

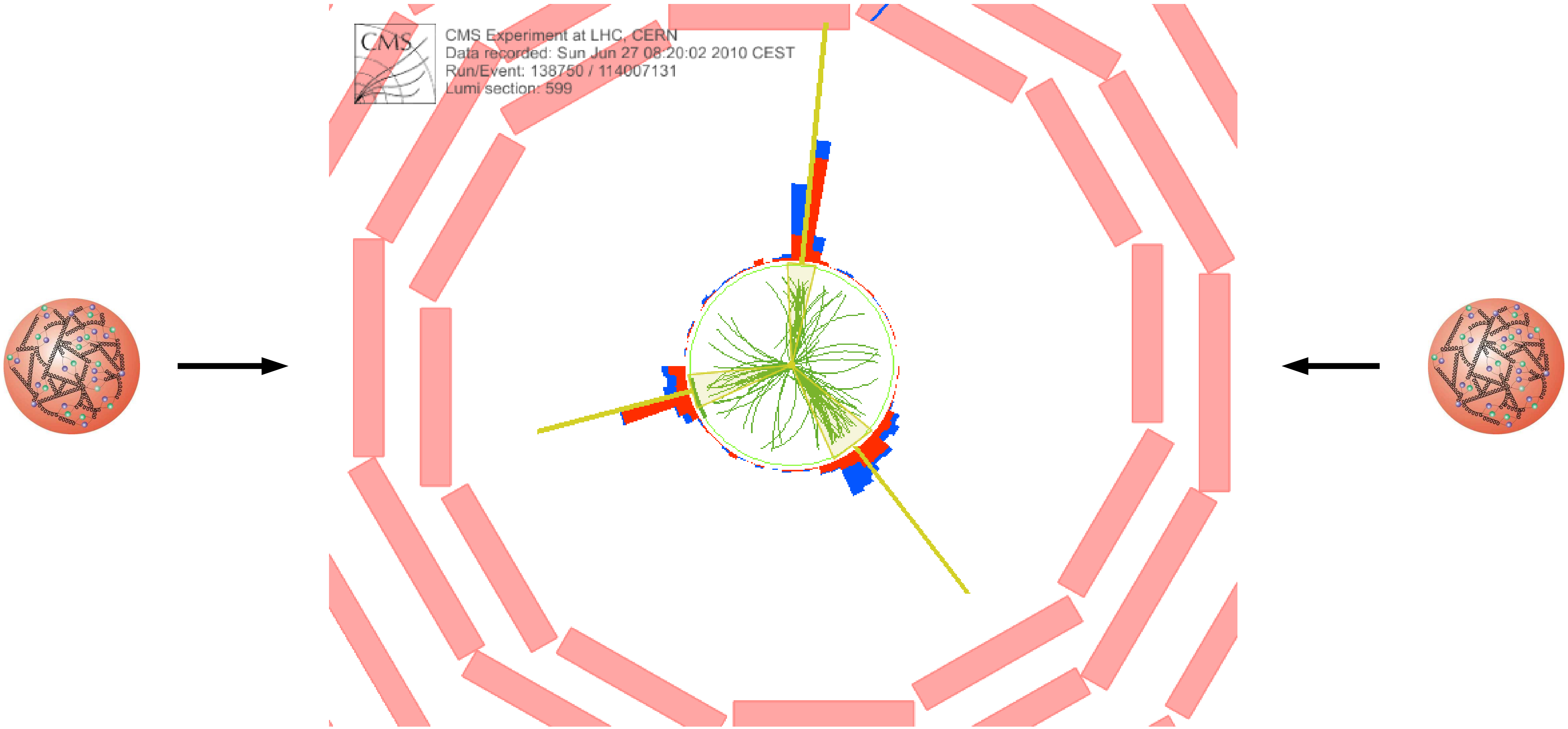
GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

Karlsruhe Institute of Technology

PHYSICS AT THE TERA SCALE
Helmholtz Alliance

Outline

- Introduction & History
- Application & Latest Status
- Current Developments
- Outlook



Motivation

- Interpretation of experiment data relies on:
  - Availability of reasonably fast theory calculations
  - Often needed: Repeated computation of same cross section
- Examples for a specific analysis:
  - Estimate accuracy of perturbative QCD (scale uncertainties)
  - Use of various PDFs (AB(K)M, HERAPDF, CTEQ, MSTW, NNPDF, …)
  - Determine PDF uncertainties (PDF error sets)
  - Use data set in fit of PDFs and/or $\alpha_s(M_Z)$
- Sometimes NLO predictions can be computed fast
- But some are very slow, esp. jets, O(1000s CPU h)
- Need procedure for fast repeated computations of NLO cross sections
- Even more so at NNLO when available!

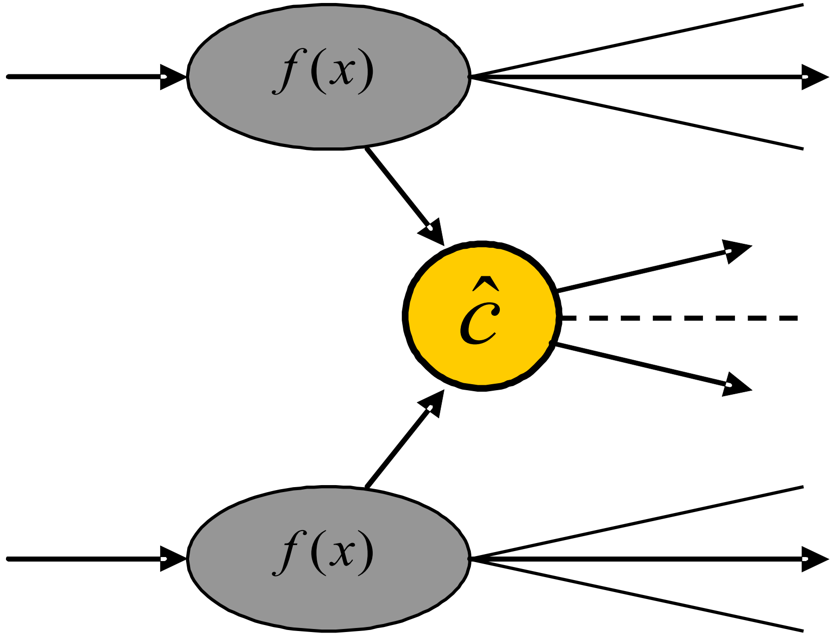
See talk from Nigel!

Jet Cross-Sections

- **Jet production in hadron-hadron collisions depends on**

$$\sigma = \sum_{a,b,n} \int_0^1 dx_1 \int_0^1 dx_2 \alpha_s^n(\mu_r) \cdot c_{a,b,n}(x_1, x_2, \mu_r, \mu_f) \cdot f_{1,a}(x_1, \mu_f) f_{2,b}(x_2, \mu_f)$$

- **strong coupling α_s to order n**
- **PDFs of two hadrons f_1, f_2**
- **Parton flavors a, b**
- **perturbative coefficients $c_{a,b,n}$**
- **renormalization and factorization scales**
- **Parton momentum fractions x**



PDF and α_s are external input
Perturbative coefficients are independent from PDF and α_s

Idea: Avoid folding integrals and factorize the PDFs and α_s

Use interpolation kernel

- Introduce set of n discrete **x-nodes**, x_i's being equidistant in a function f(x)
- Take set of **Eigenfunctions $E_i(x)$** around nodes x_i

→ Interpolation kernels

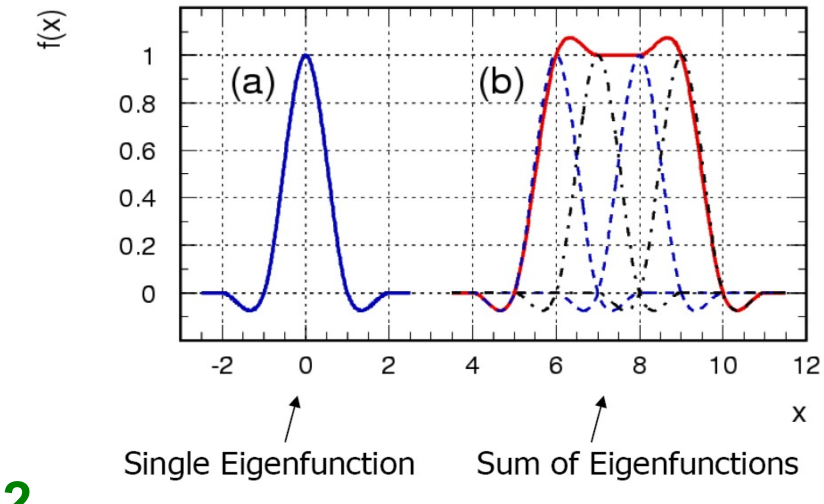
- Actually a rather old idea, see e.g.
  C. Pascaud, F. Zomer (Orsay, LAL), LAL-94-42

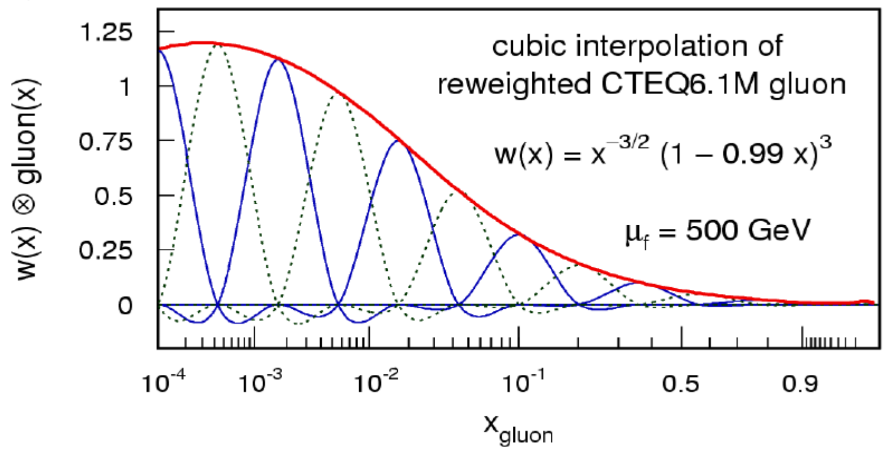
→ Single PDF is replaced by a linear combination of interpolation kernels

$$f_a(x) \cong \sum_i f_a(x_i) \cdot E^{(i)}(x)$$

→ Then the integrals are done only once

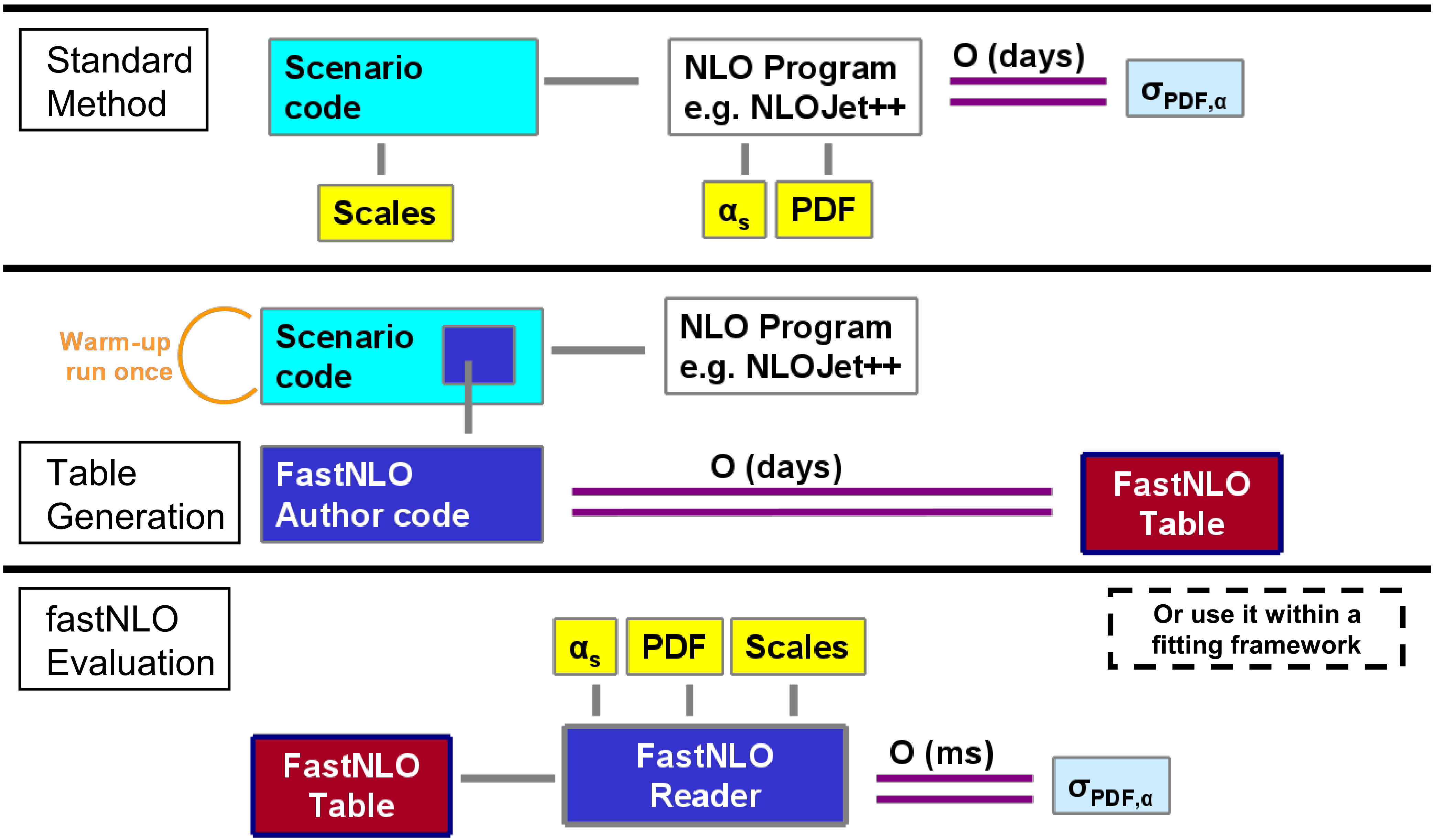
→ Afterwards only summation required to change PDF





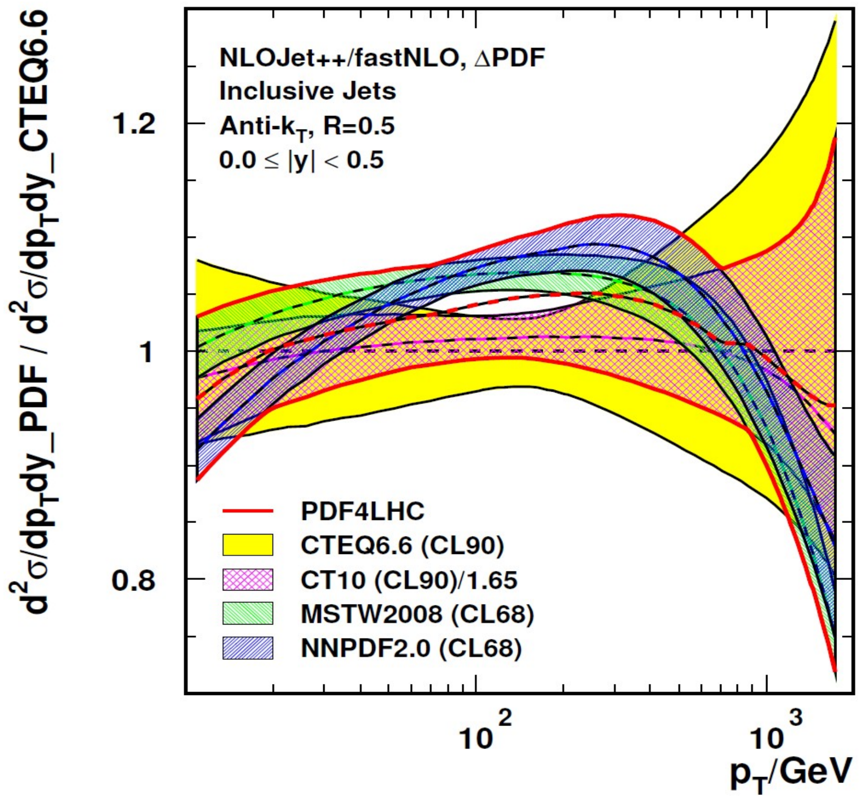
Store a table with the convolution of the pert. coefficients with the interpolation kernel

Conceptual Overview



Example Applications

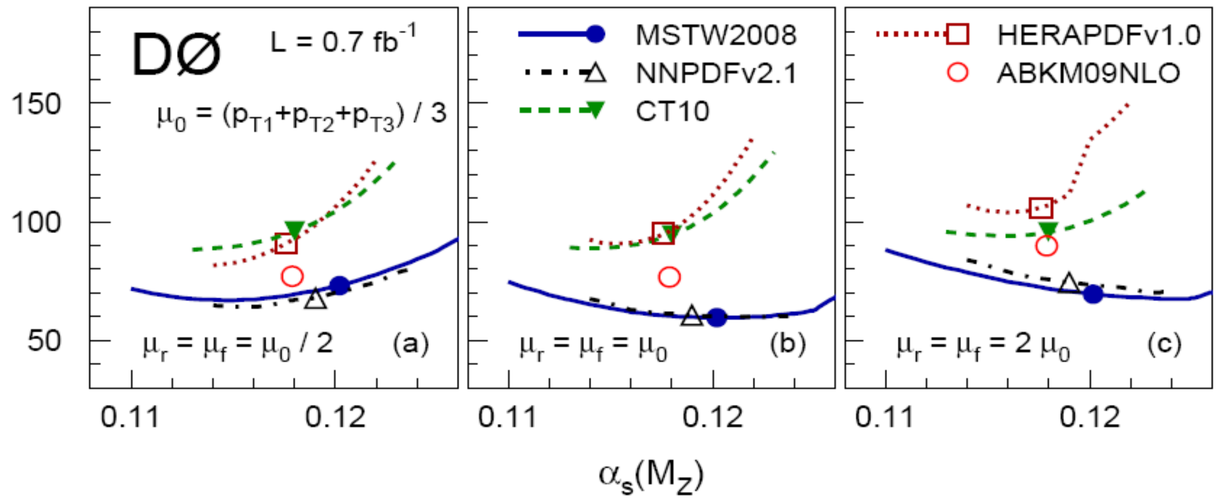
- **CMS inclusive jets**
  - Study of PDF dependence
  - Determination of PDF envelopes
  - PDF-error prediction à la PDF4LHC (not recommended ...)



239 repeated NLO calculations

D0 three-jet invariant mass

- Study of PDF dependence
- Study of scale dependence
  $\mu_r = \mu_f = (p_{T1} + p_{T1} + p_{T1})/3$
  $\mu = 2.0 \times \mu_0$
  $\mu = 0.5 \times \mu_0$
- Study of α_s dependence using α_s dependent PDF sets

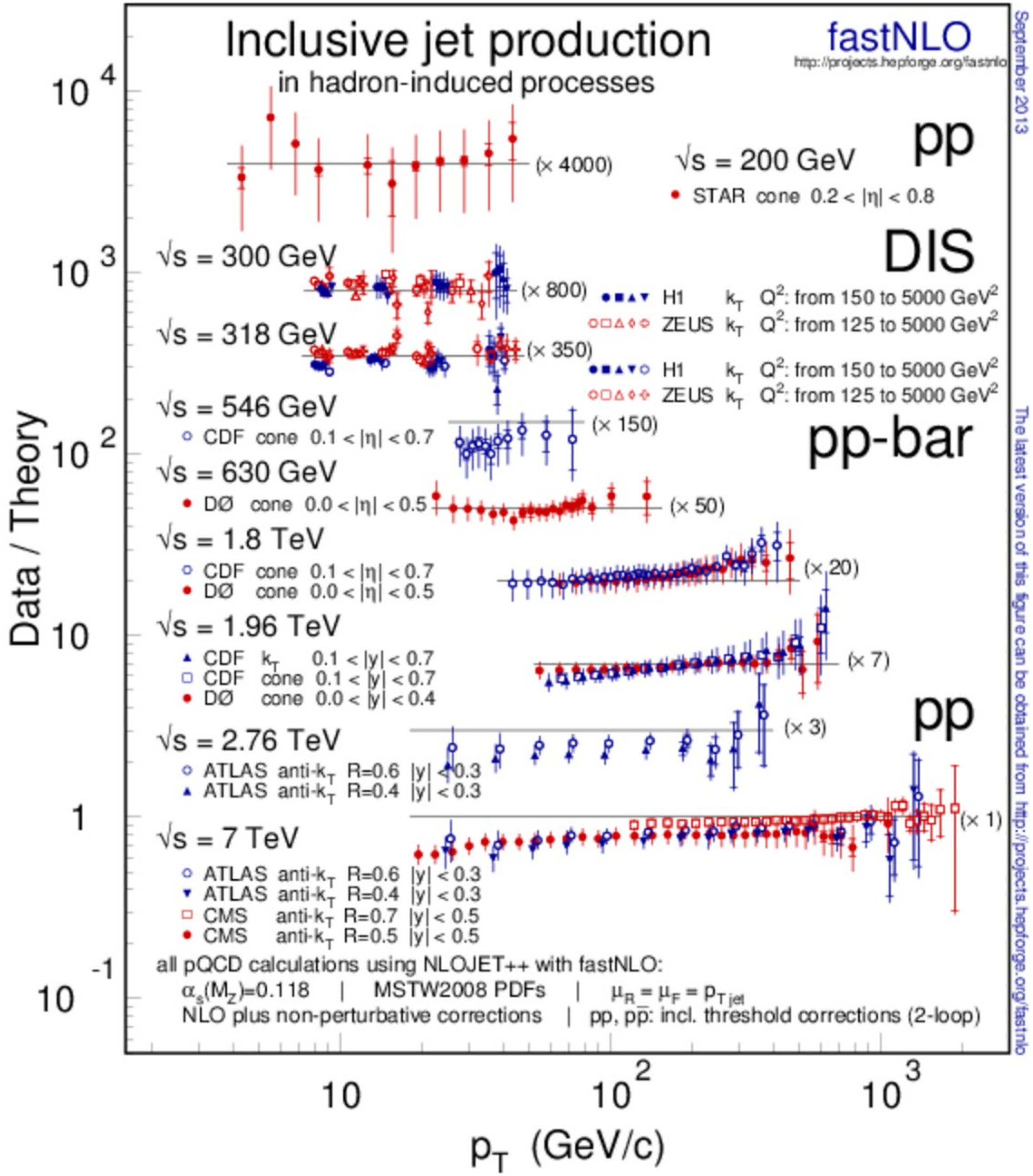


3138 repeated NLO calculations

Each rederivation takes fractions of a second!
Didn´t count any more for the fits presented next ...

Jets Compilation Plot in PDG Book

- **Comparison of jet data from**
  - **STAR at RHIC**
  - **H1 and ZEUS at HERA**
  - **CDF and D0 at Tevatron**
- **Compatible with NLO pQCD**
- **Updated last month with ATLAS 2010 and CMS 2010 & 2011 published LHC measurements**



New in fastNLO Version 2.1

- **Features of pre-computed fastNLO tables:**
  - Automatic adjustment of phase space boundaries
  - Flexible # x-nodes for analysis bins
  - Improved interpolation in ren./fact. scales
  - Arbitrary number of dimensions for binning of observable
  - Support for diffractive PDFs

FastNLO Table

- **Features of fastNLO reading tools:**
  - Easy to install (autotools)
  - Comes as a library linkable from other programs + one example executable
  - Easy implementation of new interfaces
  - Easy to implement in fitting codes and to interface PDFs
  - Independent C++ and Fortran versions
    - agreement at double precision $O(10^{-10})$

FastNLO Reader

Reader_f

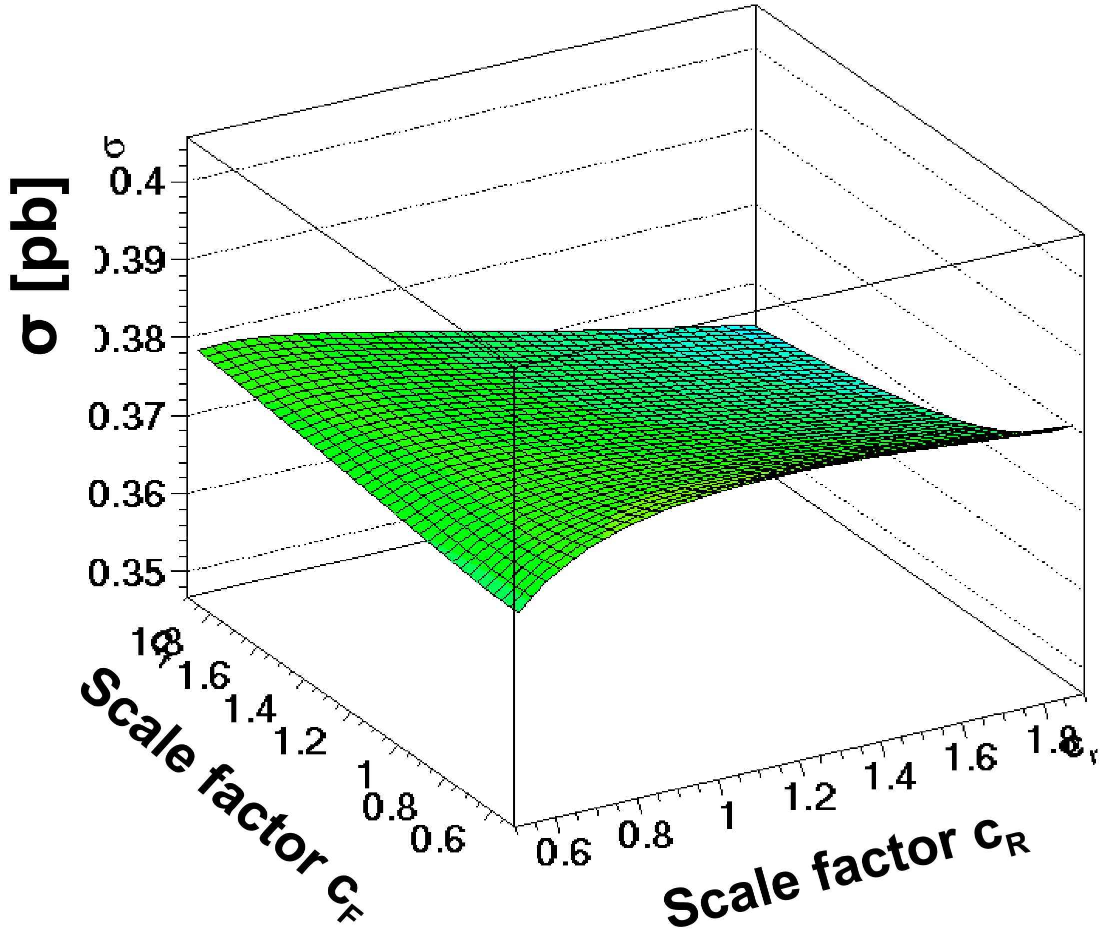
Reader_cc

- **Latest release fastnlo_reader_2.1.0_1488 including some fixes for threshold corrections in C++**

No further dependencies (No ROOT, No CERNLIB, etc…)

New in v2: Scales

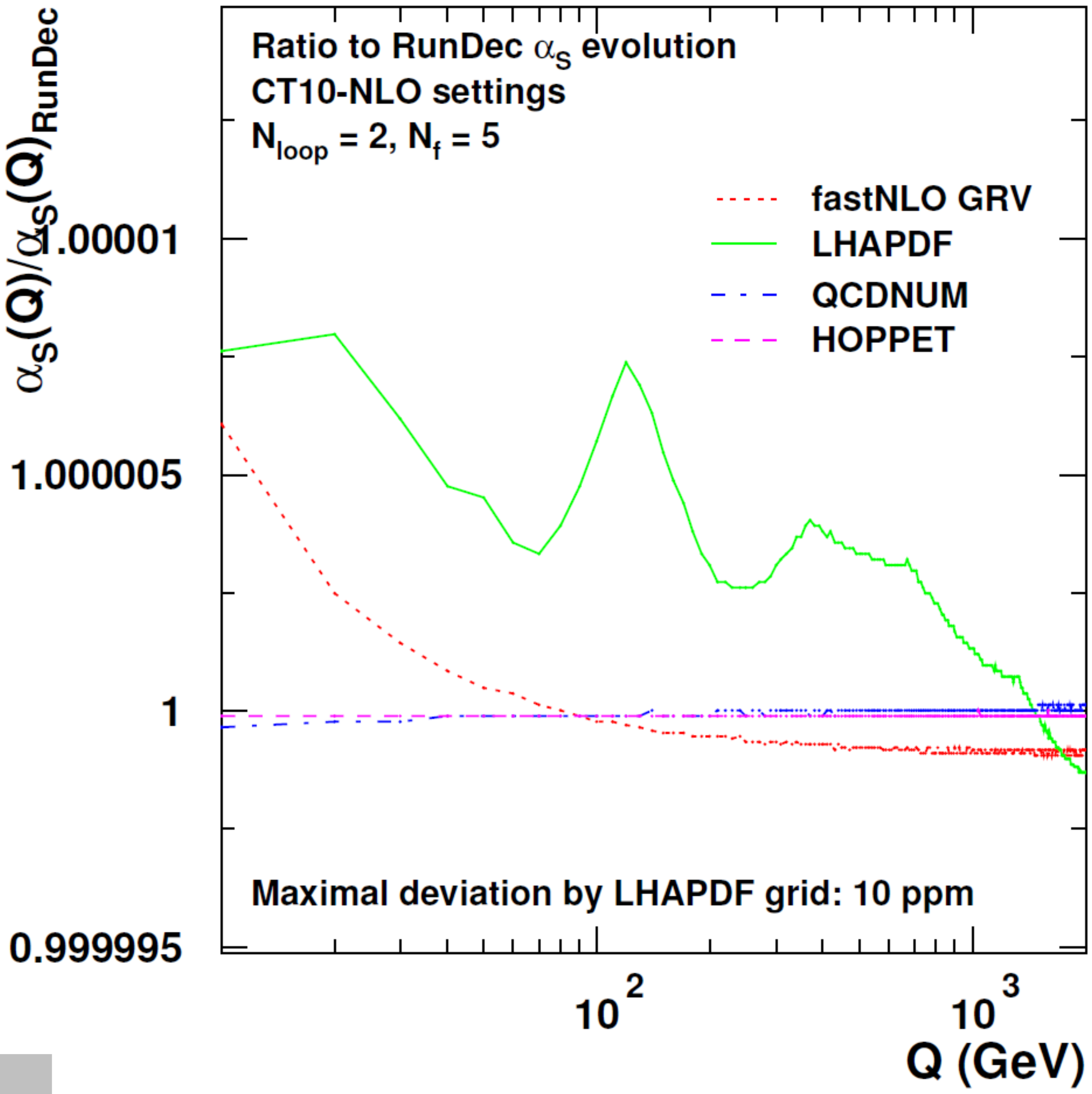
- Scales can be **functions of multiple observables**
  - e.g. for DIS jets
    Scale observables are p_T and Q^2
- **Functional form of combination can be changed**
  - Scales can be
    $\mu_r^2 = (Q^2 + p_T^2) / 2$
    $\mu_r^2 = Q^2$
    $\mu_r^2 = p_T^2$
    $\mu_r^2 = 0.8\, Q^2 + 0.3\, p_T^2 + Q \cdot p_T$
- **Independent scale variations with arbitrary factors of μ_r and μ_f are possible**
  $\mu_R^2 = c_R^2 \times (Q^2 + p_T^2) / 2$
  $\mu_F^2 = c_F^2 \times Q^2$
- **Extensively tested and used in new HERA results to come, not yet exploited for hh tables**



More flexibility for studies of scale dependencies

Next fastNLO reader release

- One further release imminent:
  - with optional Python interface to C++ library
  - with optional interface to alternative α_s evolution code from
    - CRunDec (included already since a longer time)
    - QCDNUM (used in HERAFitter)
    - HOPPET (used by CTEQ, NNPDF(?))



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.

fastNLO page at HepForge

New version with plotting tool exists (G. Sieber), needs some optimization

FastNLO is hosted by Hepforge, IPPP Durham

fastNLO

fast pQCD calculations for hadron-induced processes

Home | Documentation | Scenarios | Code | Interactive (maintenance) | Links

General concept

The fastNLO project provides computer code to create and evaluate fast interpolation tables of pre-computed coefficients in perturbation theory for observables in hadron-induced processes.

This allows fast theory predictions of these observables for arbitrary parton distribution functions (of regular shape), renormalization or factorization scale choices, and/or values of alpha_s(Mz) as e.g.

September 20-30, 2013

New code, tables, and plots available. More work is in progress:

A new release of the fastnlo_reader code (1488) is available. Please go to this page for more info and download.

Tevatron inclusive jets tables have been converted to be used with the new format. Please go to this page for an overview and download. There are more tables in the queue.

Update of "all jets plot" for 2013 available including 2011 ATLAS data at 2.76 TeV and 2011 CMS data at 7 TeV. See the Documentation tab.

Available Tables in v2.1: LHC

Changed table numbering scheme, now contains two parts:

- our internal development number fnlxxxx
- the reference number of the publication in inSPIRE, similar as in RIVET
- makes it easy to connect with relevant publication and HepData files

LHC: pp @ sqrt(s)= 7 TeV		
fnl2332d_I1208923	CMS inclusive jets 2011 (anti-kT R=0.7; pT, y); LO, NLO	
	inSPIRE record	HepData at Durham
fnl2412e_I1208923	CMS dijet mass 2011 (anti-kT R=0.7; Mjj, y_max); LO, NLO	
	inSPIRE record	HepData at Durham
fnl2622f_I1090423	CMS dijet angular 2011 (anti-kT R=0.5; Chi, Mjj); LO, NLO	
	inSPIRE record	HepData at Durham (to be uploaded by CMS)
fnl1016_I1082936	ATLAS inclusive jets 2010 (anti-kT R=0.4; pT, y); LO, NLO, THC-2loop	
	inSPIRE record	HepData at Durham
fnl1015_I1082936	ATLAS inclusive jets 2010 (anti-kT R=0.6; pT, y); LO, NLO, THC-2loop	
	inSPIRE record	HepData at Durham
fnl1014_I902309	CMS inclusive jets 2010 (anti-kT R=0.5; pT, y); LO, NLO, THC-2loop, NPC, Data	
fnl1014_cv21_I902309	CMS inclusive jets 2010 (anti-kT R=0.5; pT, y); LO, NLO; NLOJet++-2.0.1 & fastNLO-1.4.0	
	inSPIRE record	HepData at Durham
fnl2412c_I895742	CMS dijet mass 2010 (anti-kT R=0.7; pT, y_max)	
	inSPIRE record	HepData at Durham

FastNLOCreator v2.2

- 1. Cross check old v1.4 versus new v2.1 tables … Done!
- 2. Cross check new reader code in C++ vs. Fortran … Done!
- 3. Public release of reader code as autotools tarball … Done!
- 4. Transform C++ reader code into linkable library … Done!
- 5. Transform table creation code into linkable library as independent as possible from NLOJet++!
- In progress, first test version exists.
- Make interface available for other N?LO codes.

Basic Interface to other Codes

Generator	Interface	fastNLO
Executable **Calculation of coefficients (weights)** **Calculation of observables and scales** **Phase space definition** **Event count** **Must provide:** **• x-values** **• weights** **• process IDs** **• Observable and scale values**	Weight(s) x-values Process-ID Observable(s) Scale(s) Optional: pass executable specific information to fastNLO during initialization	**Binning** **Bingrid** **Interpolation** **Warmup handling** **Steerfile must provide correct** **• Process dependent information** **• Generator dependent information**

fastNLO library can always be compiled without generator specific code !!

Interface knows about generator specific issues and holds fastNLOCreate instance

Generator has not to be modified !
If generator code is complicated: modify code to pass information to interface

Logic of Table Creation

Initialization step

fastNLOCreate

fastNLOCreate(steerfile)

Initialize fastNLOCreate

All (ALL) information is read from steer-file.

Only quantities, which are given by user to ‘program-steering’ (e.g. LO or NLO run) are passed to the fastNLOCreate class.

fastNLOCreate may also pass steering values to program!

Event loop

fnloEvent **fnloScenario**

fastNLOCreate

Fill(event,scen)

Pass information to fastNLOCreate

For every subprocess/event

End of program

fastNLOCreate

WriteTable()

Write table

(pass number of events to table [event count is left to generator])

Outlook

- Several releases of fastNLO table reading library done. One more to come with optional interfaces to C++ lib via Python and to other α_s evolutions.
- Work on generalized library and interface for table reading AND creation in progress; expect first stable version beginning of next year.
- In particular working on integration of
  - Threshold correction code with Kumar and S. Moch
  - ttbar with M. Guzzi
  - Jets at NNLO with N. Glover, J. Pires, T. Gehrmann, Gehrmann-de Ridder
  - Contact Interaction @ NLO code from J. Gao (he actually implemented his own version of such an interpolation a la fastNLO or APPLGRID)
  - MCFM (available in APPLGRID)

Partonic Subprocesses

- Our test case in 2005/6: Jets @ NLO with NLOJet++
- Don't want to deal with 13 X 13 PDFs

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

- For hh → jets seven relevant partonic subprocesses

$$
\begin{array}{llcll}
1) & gg & \Rightarrow & \text{jets} & \propto H_1(x_1,x_2) \\
2) & qg,\bar{q}g & \Rightarrow & \text{jets} & \propto H_2(x_1,x_2) \\
3) & gq,g\bar{q} & \Rightarrow & \text{jets} & \propto H_3(x_1,x_2) \\
4) & q_iq_j,\bar{q}_i\bar{q}_j & \Rightarrow & \text{jets} & \propto H_4(x_1,x_2) \\
5) & q_iq_i,\bar{q}_i\bar{q}_i & \Rightarrow & \text{jets} & \propto H_5(x_1,x_2) \\
6) & q_i\bar{q}_i,\bar{q}_iq_i & \Rightarrow & \text{jets} & \propto H_6(x_1,x_2) \\
7) & q_i\bar{q}_j,\bar{q}_iq_j & \Rightarrow & \text{jets} & \propto H_7(x_1,x_2)
\end{array}
$$

- Need only seven linear combinations H_i of PDFs

Symmetries

- **In addition, symmetries can be exploited:**

$$H_n(x_1,x_2) = H_n(x_2,x_1) \quad \text{for} \quad n = 1,4,5,6,7$$
$$H_2(x_1,x_2) = H_3(x_2,x_1)$$

- **For hadron anti-hadron collisions, replace:**

$$H_4(x_1,x_2) \leftrightarrow H_7(x_1,x_2)$$
$$H_5(x_1,x_2) \leftrightarrow H_6(x_1,x_2)$$

- **Minimize table size, otherwise number of bins in observable times x_1-, x_2-, μ-nodes, … can quickly get huge!**
- **Very relevant in 2005/6 because of limited disk space in mass production of tables, problem to fit table into memory, Fortran limitations**
- **Cumbersome: Adaptation to be done for each new process**
- **Today: Partially solved using C++ and memory/disk nowadays**
- **Could even try using 13x13 in a first step for new processes**

Reader Code Download

Choose fastNLO version

Latest
Version 2.1

Previous (deprecated)
Version 1.4

Installation

Installation of distribution package:

Via GNU autotools setup (NOT required for installation), in unpacking directory of the *.tar.gz file do:
./configure --prefix=your_local_directory
(should contain LHAPDF installation, otherwise specify separate path via
--with-lhapdf=path_to_lhapdf; see also ./configure --help)
make
make install
make check (not yet implemented)

Requirements:

For the installation of the reader package: LHAPDF
Please use at least version 5.8.9, but not version 6 of LHAPDF. The latter has not yet been tested with fastNLO.
For running the executable: fastNLO table, PDF set from LHAPDF

For more information see the README file.

fastnlo_reader 2.1.0 releases			
1488	ReleaseNotes	ChangeLog	Recommended! Consistent treatment of 1- and 2-loop threshold corrections in C++ & Fortran
1360	ReleaseNotes	ChangeLog	Workaround for uninitialized top PDF in LHAPDF pre 5.8.9b1 removed
1354	ReleaseNotes	ChangeLog	Xmas release including experimental support for diffractive PDFs
1273	ReleaseNotes	ChangeLog	Edition for PDF school 2012 "Proton Structure in the LHC Era" at DESY
1062	ReleaseNotes	ChangeLog	First public release, presented at Marseille HERAFitter Meeting

Available Tables in v2.1: Tevatron

Tables with “cv21” refers to tables produced with the old versions of NLOJet++ 2.0.1 and fastNLO v1.4 that have been converted to the new format. More tables to come. Will be replaced at some point by newer tables from scratch.

Waiting for new tables for HERA in new flexible scale format.

Tevatron: ppbar @ sqrt(s)= 1.96 TeV		
fnt2007midp_cv21_I790693	CDF inclusive jets 2002-2006 (midpoint cone R=0.7; pT, y); LO, NLO, THC-2loop	
	inSPIRE record	HepData at Durham
fnt2009midp_cv21_I779574	D0 inclusive jets 2004/5 (midpoint cone R=0.7; pT, y); LO, NLO, THC-2loop	
	inSPIRE record	HepData at Durham
fnt2004_cv21_I743342	CDF inclusive jets (kT R=0.7; pT, y); LO, NLO, THC-2loop	
	inSPIRE record	HepData at Durham
Tevatron: ppbar @ sqrt(s) = 1.8 TeV		
fnt1001midp_cv21_I552797	CDF inclusive jets 1994/5 (midpoint cone R=0.7; ET, eta); LO, NLO, THC-2loop	
	inSPIRE record	HepData at Durham
fnt1002midp_cv21_I536691	D0 inclusive jets 1994/5 (midpoint cone R=0.7; ET, eta); LO, NLO, THC-2loop	
	inSPIRE record	HepData at Durham

Technical Cross-check

tkdiff between Fortran and C++, ALL differences in color ...!

TkDiff 4.1.4

File Edit View Mark Merge

fnl1014_v14_cv21_fread_1062.log

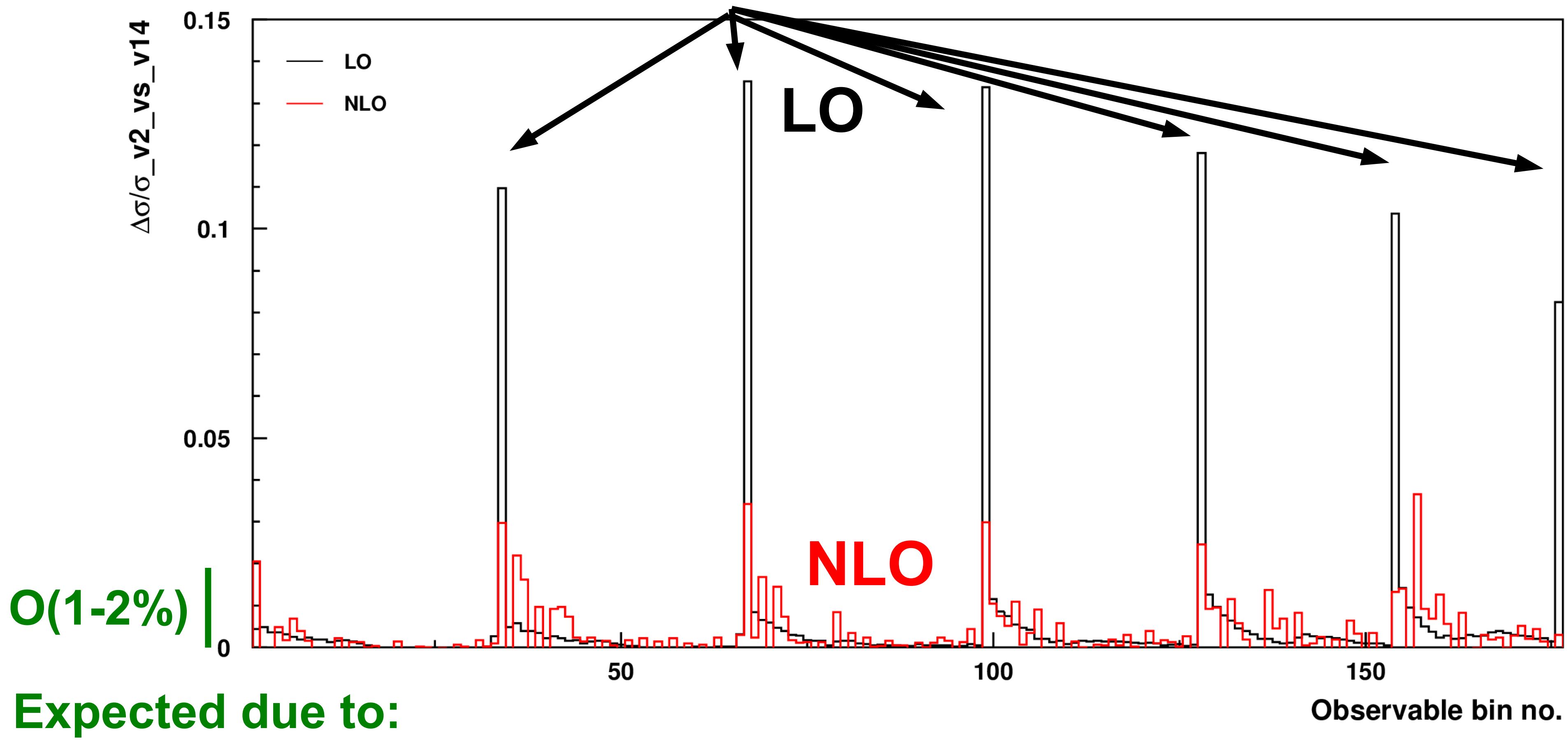
Cross Sections
The scale factor chosen here is: 1.000

IObs	Bin Size	IODim1	[pT_[GeV]	]	IODim2	[|y|	]	LO cross section	NLO cross section	K NLO
1	3.000	1	18.00	21.00	1	0.00E+00	5.00E-01	1.57274581281E+07	1.63402311907E+07	1.03896
2	3.000	2	21.00	24.00	1	0.00E+00	5.00E-01	8.38588042457E+06	8.92499652457E+06	1.06429
3	4.000	3	24.00	28.00	1	0.00E+00	5.00E-01	4.44617619413E+06	4.68895651667E+06	1.05460
4	4.000	4	28.00	32.00	1	0.00E+00	5.00E-01	2.32175304480E+06	2.48739373594E+06	1.07134
5	5.000	5	32.00	37.00	1	0.00E+00	5.00E-01	1.22985606580E+06	1.31501340014E+06	1.06924
6	6.000	6	37.00	43.00	1	0.00E+00	5.00E-01	6.20058716819E+05	6.57353581654E+05	1.06015
7	6.000	7	43.00	49.00	1	0.00E+00	5.00E-01	3.19183821541E+05	3.42274328312E+05	1.07234
8	7.000	8	49.00	56.00	1	0.00E+00	5.00E-01	1.69704477492E+05	1.83046529104E+05	1.07862
9	8.000	9	56.00	64.00	1	0.00E+00	5.00E-01	8.87915718598E+04	9.55576371649E+04	1.07620
10	10.00	10	64.00	74.00	1	0.00E+00	5.00E-01	4.47860610011E+04	4.83398734386E+04	1.07935
11	10.00	11	74.00	84.00	1	0.00E+00	5.00E-01	2.26334926926E+04	2.44897733616E+04	1.08201
12	13.00	12	84.00	97.00	1	0.00E+00	5.00E-01	1.14157974746E+04	1.23657778458E+04	1.08322
13	17.00	13	97.00	114.0	1	0.00E+00	5.00E-01	5.20864150541E+03	5.66705156918E+03	1.08801
14	19.00	14	114.0	133.0	1	0.00E+00	5.00E-01	2.26986160457E+03	2.47492393341E+03	1.09034
15	20.00	15	133.0	153.0	1	0.00E+00	5.00E-01	1.03027761770E+03	1.12801906894E+03	1.09487
16	21.00	16	153.0	174.0	1	0.00E+00	5.00E-01	4.94585929406E+02	5.44349660966E+02	1.10062
17	22.00	17	174.0	196.0	1	0.00E+00	5.00E-01	2.48671425936E+02	2.74189028880E+02	1.10262
18	24.00	18	196.0	220.0	1	0.00E+00	5.00E-01	1.28423986831E+02	1.42067642887E+02	1.10624
19	25.00	19	220.0	245.0	1	0.00E+00	5.00E-01	6.77424165982E+01	7.54903551563E+01	1.11437
20	27.00	20	245.0	272.0	1	0.00E+00	5.00E-01	3.65423220021E+01	4.07221522939E+01	1.11438
21	28.00	21	272.0	300.0	1	0.00E+00	5.00E-01	2.00810227037E+01	2.24894597220E+01	1.11994
22	30.00	22	300.0	330.0	1	0.00E+00	5.00E-01	1.12407556895E+01	1.26477375709E+01	1.12517
23	32.00	23	330.0	362.0	1	0.00E+00	5.00E-01	6.33683801220E+00	7.14852031809E+00	1.12809
24	33.00	24	362.0	395.0	1	0.00E+00	5.00E-01	3.62773698109E+00	4.11706825391E+00	1.13489
25	35.00	25	395.0	430.0	1	0.00E+00	5.00E-01	2.10813697037E+00	2.40328694241E+00	1.14001
26	38.00	26	430.0	468.0	1	0.00E+00	5.00E-01	1.22390945155E+00	1.40066739688E+00	1.144[illegible]2
27	39.00	27	468.0	507.0	1	0.00E+00	5.00E-01	7.14273567995E-01	8.21258466624E-01	1.14978
28	41.00	28	507.0	548.0	1	0.00E+00	5.00E-01	4.22300908307E-01	4.88483283158E-01	1.15672
29	44.00	29	548.0	592.0	1	0.00E+00	5.00E-01	2.49475414467E-01	2.89536598089E-01	1.16058
30	46.00	30	592.0	638.0	1	0.00E+00	5.00E-01	1.47171713316E-01	1.72087362982E-01	1.16930
31	48.00	31	638.0	686.0	1	0.00E+00	5.00E-01	8.71981367924E-02	1.02345684984E-01	1.17371
32	51.00	32	686.0	737.0	1	0.00E+00	5.00E-01	5.16004131315E-02	6.09739215817E-02	1.18166
33	109.0	33	737.0	846.0	1	0.00E+00	5.00E-01	2.39696393032E-02	2.86096708138E-02	1.19358
34	838.0	34	846.0	1684.	1	0.00E+00	5.00E-01	1.64803906607E-03	2.15929120721E-03	1.31022

Identical at $O(10^{-10})$

Cross-check v2 vs. v14

Feature known from discussion with CTEQ:
Small scale offset in highest pT XXL bin → resolved in v2!



Expected due to:
Stat. independent calculations, NLOJet++_2.0.1 → NLOJet++_4.1.3, improved x limits/binning, …