Usage of *fast*NLO in PDF fits

Introduction

Proton Structure in the LHC Era - School and Workshop
30. September 2014

Daniel Britzger, Georg Sieber, Klaus Rabbertz

Outline

Preparation of Virtual Machine

Introduction

- Motivation
- General concept of fastNLO
- Application to Jet analysis at LHC
- Outlook

Tutorial/Hands-on

- Download/Installation
- Example of table creation using nlojet++ for CMS inclusive jets
- Example of table evaluation and use various PDF sets
- Representation of fastNLO+nlojet++ results with rivet

Q&A

Preparation of Virtual Machine

1. **Download setup script from web**

   `$> wget http://fastnlo.hepforge.org/setup_fastNLO.sh`

2. **Source script**

   `$> source setup_fastNLO.sh`

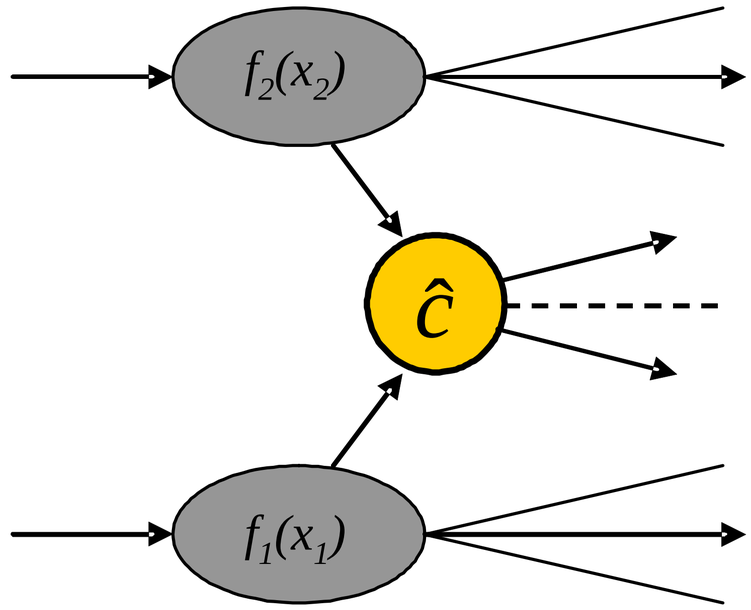
3. **Close your screen**

Basics of QCD cross section calculation

Cross section in hadron-hadron collisions in pQCD

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

- strong coupling α_s in order n
- PDFs of two hadrons f_1, f_2
- Parton flavors a, b
- perturbative coefficent $c_{a,b,n}$
- renormalization and factorization scales μ_r, μ_f
- momentum fractions x_1, x_2



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

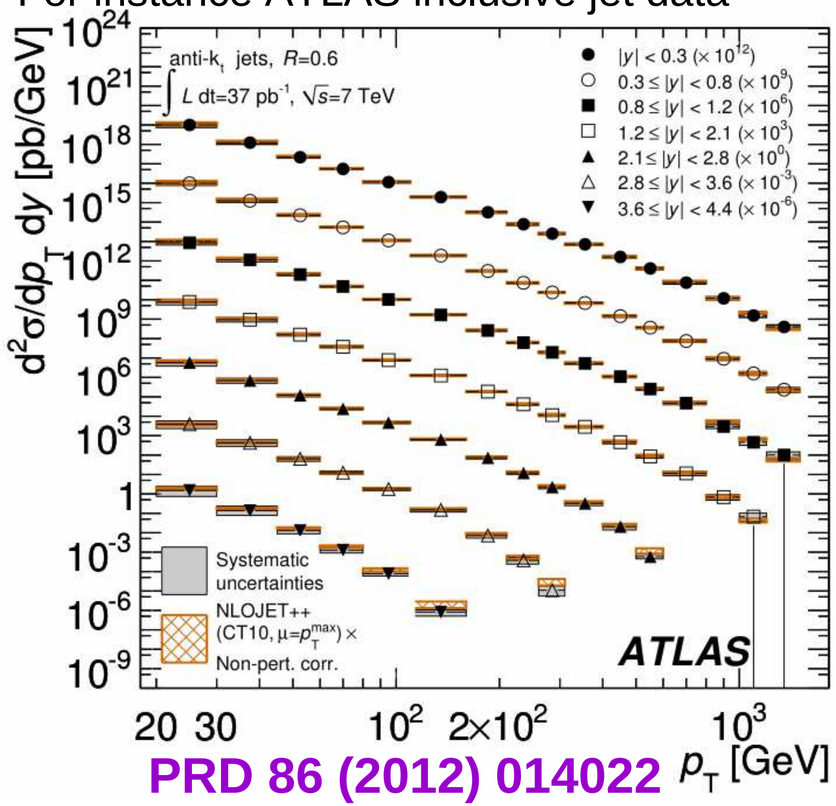
Basics of QCD cross section calculation

Cross section in hadron-hadron collisions in pQCD

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

Application in PDF fits

For instance ATLAS inclusive jet data



PRD 86 (2012) 014022

1. Fit *theory* to *data*:

$$\sigma_{\text{theo}} \approx \sigma_{\text{exp}}$$

2. Free parameters of theory in fit could be any theory parameter
3. Typically for PDF fits
   a) Fix perturbative coefficients
   b) $\alpha_s(M_Z)$ could be free parameter or not
   c) Fit PDFs: $f_{1,a}$, $f_{2,b}$

Goal: Provide theory coefficients $c_{a,b,n}$ such that they can be used in a (PDF) fit

A users view on (N)NLO calculations

Many (N)NLO calculations take months (or even years) for reasonable precision

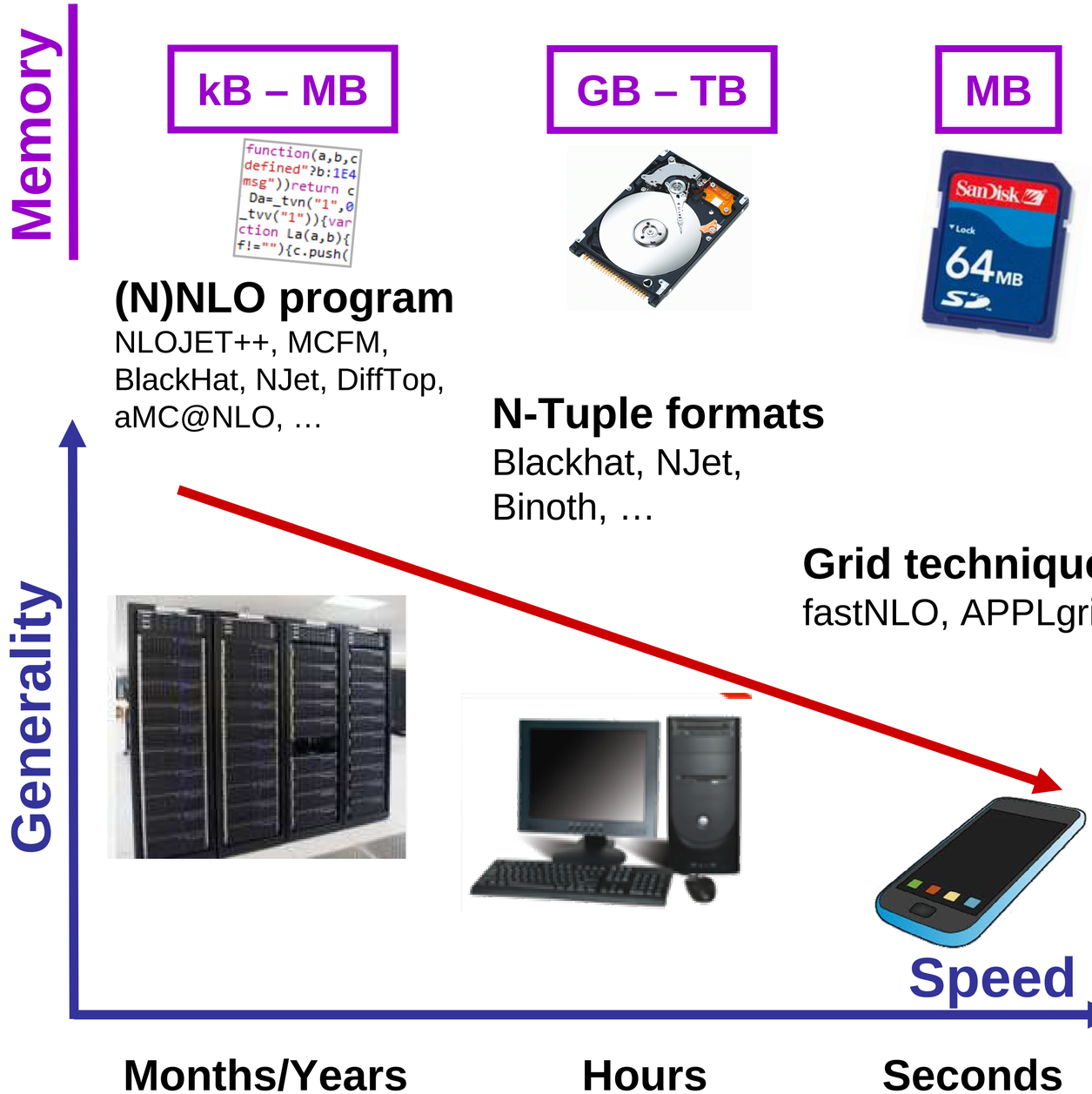
Recent developments require different tools for e.g. real or virtual contributions

Requirements for PDF fit

- Repeated cross section calculations within O(s) or O(ms)
- Standardized format for various processes

Several tools for specific applications available

- (N)NLO program standalone
- ‘N-Tuple’ files
- Grid techniques



Sketch adapted from D. Maitre

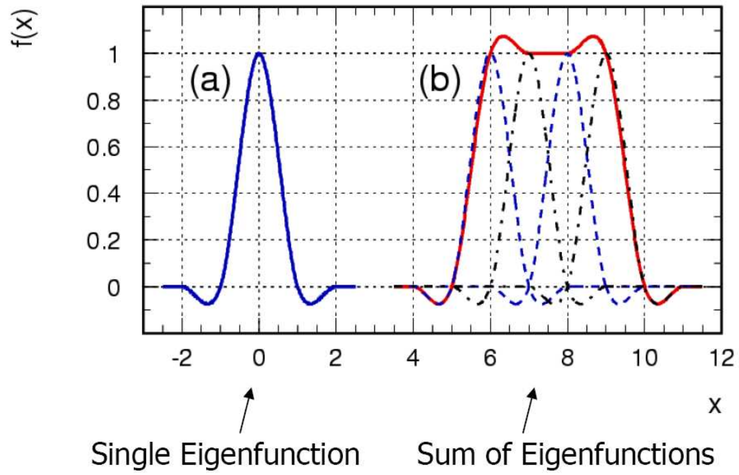
fastNLO working principle

1. **Introduce a set of nodes *j* (‘grids’)**
2. **Replace PDF functions in (N)NLO code with**

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

3. **Interpolation kernel must fulfill**

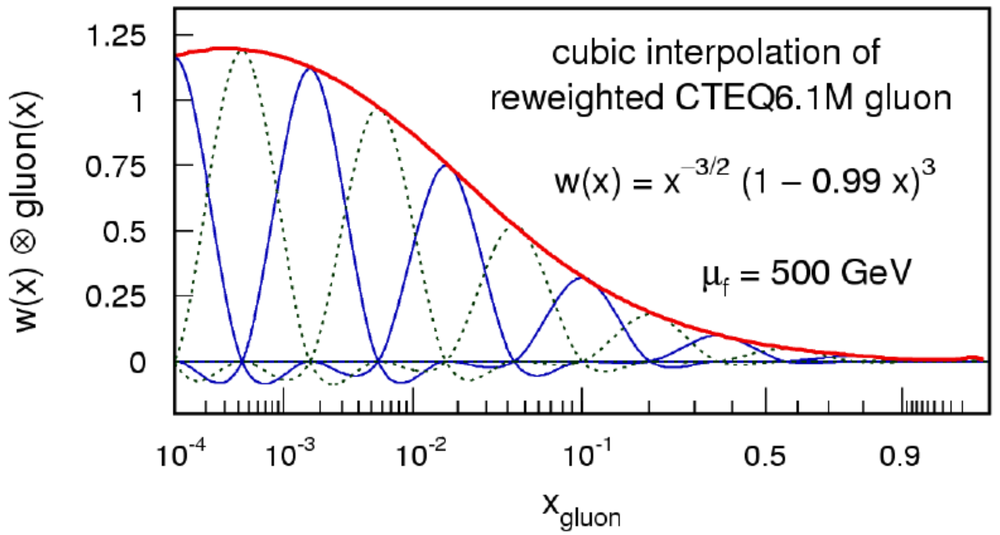
$$\sum_i E_i(x) = 1\,, \qquad E_i(x_j) = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{otherwise} \end{cases}$$



4. **Make use of symmetries for specific process**

$$\sum_{a,b}^{13\times13} f_{1,a}(x_1,\mu_f)\, f_{2,b}(x_2,\mu_f) \rightarrow \sum_k^7 H_k(x_1,x_2,\mu_f)$$

5. **Store coefficients in a table:** $\tilde{\sigma}$
   The cross section for usage in PDF fits can be rewritten as a simple sum



$$\sigma_{hh}^{Bin} = \sum_{i,j,k,n,m} \alpha_s^n(\mu^{(m)}) \cdot H_k(x_1^{(i)}, x_2^{(j)}, \mu^{(m)}) \cdot \tilde{\sigma}_{k,n}^{(i,j)(m)}$$

More details on fastNLO

Storage of coefficients is more general

- Scale dependent contributions are stored separately

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

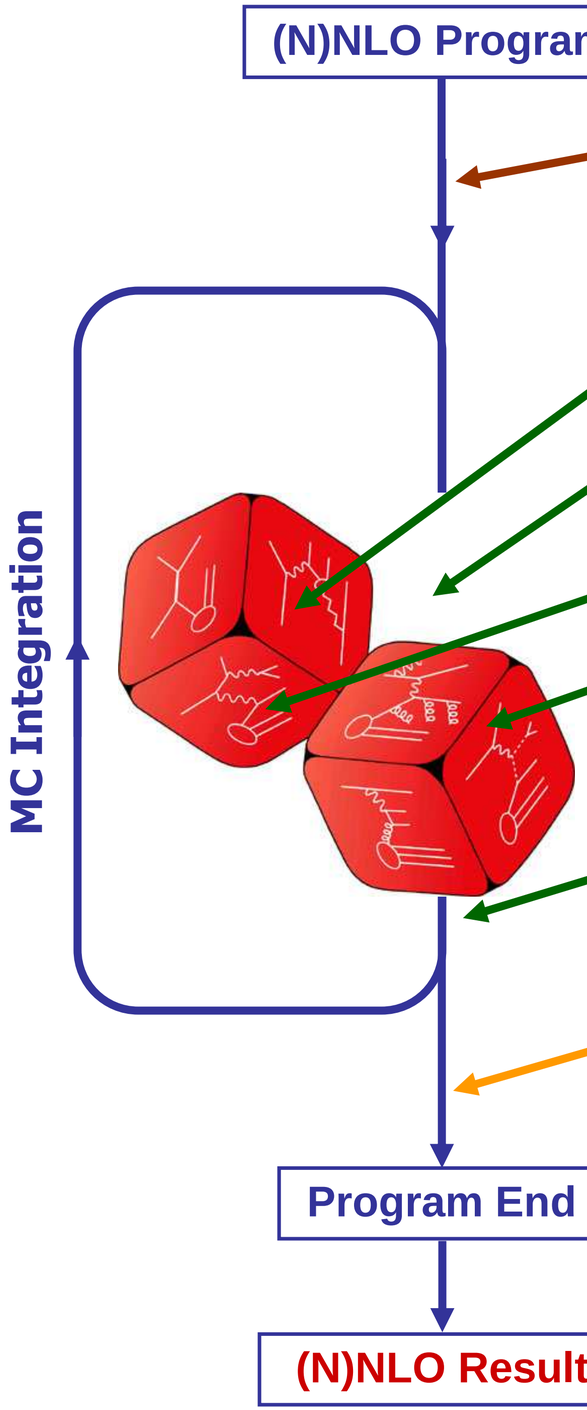
- Store weights: w_0, w_R, w_F, w_{RR}, w_{FF}, w_{RF} for order α_s^{n+2} contributions
- This allows for free choice of renormalization and factorization scale without recalculation of coefficients:
  Two observables can be stored in table which can be employed for calculation of scales

Automated scan of the grids to phase space

- So-called 'warm-up' run: More details in hands-on session

Many other performance and memory size optimizations

Application procedure I: Table creation




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

```
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);
```

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

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

Initialize fastNLO class(es)

Pass the process specific variables during the ‘event loop’ to fastNLO

- Order does not matter
- Many other convenient implementations possible

Pass all information to fastNLO

Set normalization of the MC integration and write table

Minimum implementation:
11 lines of code

Application procedure II: Evaluating tables

Evaluating requires interface to PDF library

- LHAPDF
- PDF fitting framework
- QCDNUM
- …

Intermediate step:
Merge/Append LO tables with NLO tables

Strong coupling evolution can be provided by external program or with shipped code

Usage in your program if you want to evaluate table file fnl1014.tab

```
#include <fastNLOLHAPDF.h>

[...]
// FastNLO example code in c++ for reading CMS incl.
// jets (PRL 107 (2011) 132001) with CT10 PDF set

fastNLOLHAPDF fnlo("fnl1014.tab","CT10.LHgrid",0);
fnlo.PrintCrossSections();                    // Print cross section to screen
vector<double> cs = fnlo.GetCrossSection();   // Access cross sections for later usage
```

Standalone program(s) available: `fnlo-tk-cppread` or `fnlo-tk-example`

More options (like scaling variations, α_s settings, etc…) discussed in hands-on session

Further information

http://fastnlo.hepforge.org/

Summary

fastNLO is a tool for enabling the usage of time consuming theory prediction in (PDF) fits

It is not a NLO program or a MC generator

For the usage of fastNLO two steps are required

1. Create table using the fastNLO toolkit code together with an (N)NLO program (fastNLOCreate class) or download tables on fastnlo.hepforge.org
2. Evaluate table and calculate cross sections using fastNLO stand-alone program or use fastNLO within fitting framework (fastNLOReader class)

The usage of fastNLO tables in HERAFitter is explained in other tutorials

More information, references and documentation is found at

http://fastnlo.hepforge.org

Outline

Preparation of Virtual Machine

Introduction

- Motivation
- General concept of fastNLO
- Application to Jet analysis at LHC
- Outlook

Tutorial/Hands-on

- Download/Installation
- Example of table creation using nlojet++ for CMS inclusive jets
- Example of table evaluation and use various PDF sets
- Representation of fastNLO+nlojet++ results with rivet

Q&A