

Recent developments with the APPLgrid project

Mark Sutton

University of Sussex

On behalf of the APPLgrid developers

28th July 2015

What is the APPLgrid project?

- APPLgrid is a fully open source package to build a library of C++ utility classes for performing fast (N)NLO convolutions with PDFs (with an optional FORTRAN interface)
- It uses a customised grid storage for reduced memory usage
- Can be used for fast cross section production with pre-existing grids
 - Arbitrary renormalisation and factorisation scale variation
 - Arbitrary beam-energy rescaling
 - Different PDF sets
 - Any number of multiplicative corrections can be stored and applied in the the grid
 - Allows different input PDFs for each incoming hadron
- Can be used by the user to generate custom grids for different cross sections and processes
- Currently available interfaces for ...
 - NLOjet++ for jet production
 - All processes in MCFM: Electroweak boson production, heavy flavour production, boson + jet production (plus generic interface for all remaining processes)
 - All fixed order NLO processes in aMC@NLO, using the [aMCfast](#) interface, [arXiv:1406.7693](#)
 - Sherpa for fixed NLO processes using MCgrid, [arxiv:1312.4460](#) - in addition an independent native APPLgrid - Sherpa interface is available
- Currently working on an interface to DYNNLO for EW boson production at NNLO
- In the past we have also toyed with interfaces to JETRAD (NLO) and Vrap (NNLO)

applgrid.hepforge.org

APPLgrid - Hepforge

https://applgrid.hepforge.org

APPLgrid is hosted by Hepforge, IPPP Durham

the APPLgrid project

[Home](#) [Downloads](#) [Documentation](#) [Subversion](#) [News](#) [Links](#)

Convolution Code Download
 Current version [applgrid-1.4.70](#)
 Basic example code [here](#)
 The [hoppet](#) code version 1.1.5 for QCD evolution from Gavin Salam and Juan Rojo.

Calculation Code
 MCFM: (use standard mcfm code)
[mcfm-patch](#) (mcfm applgrid patch 0.0.8)
[mcfm-bridge](#) (version 0.0.35 - for mcfm-6.8)
 NLOjet++:
[nlojet++ 4.0.1](#) (applgrid version 0.0.2)
[nlojet++ lhapdf wrapper 1.0.0](#) (applgrid version 0.0.2)
[nlojet++ user module](#) (applgrid version 0.0.2)

Grid Download
 Full details on the [Downloads](#) page:

Grids for $\sqrt{s}=2.76$ TeV:
 ATLAS inclusive jets (2011 - 0.2pb^{-1}) [grids](#)

Grids for $\sqrt{s}=7$ TeV:
 ATLAS inclusive jets (2010 - 17nb^{-1}) [grids](#)
 ATLAS inclusive jets (2010 - 37pb^{-1}) [grids](#)
 ATLAS inclusive dijets (2010 - 37pb^{-1}) [grids](#)
 ATLAS inclusive dijets (2011 - 4.5fb^{-1}) [grids](#)
 ATLAS W+, W- data (2010 - 35pb^{-1}) [grids](#)
 ATLAS Z0 data (2010 - 35pb^{-1}) [grids](#)

Quick Start Guide
[how to run the APPLgrid code](#)

Citation
 Please cite the APPLgrid as
 Eur Phys J C 66 (2010) 503

Welcome
News! New versions of applgrid and the mcfm interface released. See the [News](#) page for more details.
 The APPLgrid project provides a fast and flexible way to reproduce the results of full NLO calculations with any input parton distribution set in only a few milliseconds rather than the weeks normally required to gain adequate statistics.
 Written in C++ (although a fortran interface is included) it can be used for the calculation of any process where the hard subprocess weights from the convolution with the PDF are available from the calculation.
 The user can use existing grids simply to obtain the fast cross sections, as with fastNLO, but the complete project is publicly available should the user wish to generate the grids themselves for new cross sections. In this case, the user interfaces the grid code with NLO calculation, and after running the NLO calculation to achieve the required statistical precision once, the results of the calculation with any different parton distribution set can be calculated, typically in around 1 to 100ms, depending on the size of the grid.
 At present, examples exist for MCFM and nlojet++. Since the user code that needs to run to extract the weights and create the grid may require changes to the NLO calculation code or may be dependent on specific versions of the code, the specific versions of both MCFM and nlojet++ are included here
 An interface to fastNLO (version 1) grids is included.
 The code is under continuous development so please come back soon for more information.

Mark Sutton, Pavel Starovoitov, Tancredi Carli and Gavin Salam - send mail to the authors: applgrid@projects.hepforge.org: Last updated Tue 17 Feb 2015 15:04:19 CET

downloads.htm

Grid Downloads

Here you can download grids for APPLgrid version 1.4.70 for fast cross section evaluation. These grids are fully differential and should require no additional scaling. Each should include the non-perturbative (and any additional) bin-by-bin corrections or K-factors, the application of which is discussed in the relevant papers - see the [Documentation](#) page for more information.

Please note that these grids should be based on the grids that were used for the relevant papers, and were not necessarily created by the the applgrid authors.

LHC: pp @ sqrt(s) = 2.76TeV

ATLAS inclusive jets (2011 data 0.2pb^{-1})

arXiv:1304.4739v1 Inclusive jets anti-KT: Tables 4-10 (R=0.4) and 11-17 (R=0.6)

hepdata

grid tarball
contains atlas-incljets-arxiv-1304.4739/r04/atlas-incljets-eta[1-7].root (R=0.4)
atlas-incljets-arxiv-1304.4739/r06/atlas-incljets-eta[1-7].root (R=0.6)

LHC: pp @ sqrt(s) = 7TeV

ATLAS inclusive jets (2010 data 17nb^{-1})

arXiv:1009.5908v2 Inclusive jets anti-KT: Tables 1-3 (R=0.4) and 4-6 (R=0.6)

hepdata

grid tarball
contains atlas-incljets-arxiv-1009.5908v2/r04/atlas-incljets-eta[1-5].root (R=0.4)
atlas-incljets-arxiv-1009.5908v2/r06/atlas-incljets-eta[1-5].root (R=0.6)

ATLAS inclusive jets (2010 data 37pb^{-1})

arXiv:1112.6297 Inclusive jets anti-KT: Tables 5-11 (R=0.4) and 12-18 (R=0.6)

hepdata

grid tarball
contains atlas-incljets-arxiv-1112.6297/r04/atlas-incljets-eta[1-7].root (R=0.4)
atlas-incljets-arxiv-1112.6297/r06/atlas-incljets-eta[1-7].root (R=0.6)

ATLAS inclusive dijets (2010 data 37pb^{-1})

arXiv:1112.6297 Inclusive dijets anti-KT: Tables 19-27 (R=0.4) and 28-36 (R=0.6)

hepdata

grid tarball
contains atlas-incljets-arxiv-1112.6297/r04/atlas-incljets-eta[1-9].root (R=0.4)
atlas-incljets-arxiv-1112.6297/r06/atlas-incljets-eta[1-9].root (R=0.6)

Code Download

Various code for download is available:

- You can download the latest version (1.4.70) of the standard APPLgrid convolution code [here](#) .
- If you wish to use arbitrary factorisation scale variation, you will need to have hoppet installed first, which you can download from [here](#) .
- There are some simple examples which you can download from [here](#) . These examples requires that you have LHAPDF installed which you can find [here](#) .
- In addition, APPLgrid uses root files for storage, although not internally, so you should also install root which you can find [here](#) .

Calculation Code

If you wish to generate your own grids, we currently have NLOjet++ available for jet production at NLO and MCFM for most other processes.

- For NLOjet++ we have our own custom version 4.0.1 which can be downloaded from [here](#) . This requires the nlojet pdf module that you can download from [here](#) . The user module can be downloaded from [here](#)
- For MCFM, you should download the standard version of MCFM (6.7 or later). To link with applgrid, before installing MCFM, you should download and install the mcfm-bridge code, which you install with the usual ./configure ; make ; make install etc.

After running this you will need to patch the standard installation using the patch file which you should untar in the MCFM base directory and then remove the file

```
src/User/gridwrap.f
```

then just build using make. To link with applgrid, before running make, you should set the LDFlags environment variable to the output from

```
mcfm-bridge --ldflags
```


NB: you should not use the standard install script that comes with MCFM.


Other interfaces are being developed and will be released when available.

Downloads Archive

The old downloads directory can be found [here](#) (Warning, not for the faint of heart)

Open Source



powered by  **trac**

[Login](#) | [Help/Guide](#) | [About Trac](#) | [Preferences](#)

Wiki | Timeline | Roadmap | **Browse Source** | View Tickets | Search


[Last Change](#) | [Revision Log](#)

source:

View revision: View diff against:

Name [▲]	Size	Rev	Age	Author	Last Change
▸ amcatnlo-2.0.0		443	20 months	rojo	trivial edit
▸ amcatnlo-bridge		1234	7 months	sutt	fix autotools
▸ amcfast		1252	7 months	bertone	amcfast-shower executable added
▸ applgrid		1294	9 days	tcarli	corrected spelling mistake in comment
▸ dynnlo		387	22 months	sutt	create directory structure
▸ example		1282	5 months	sutt	fixes for pdfschool
▸ jetmod		591	19 months	sutt	rename
▸ lhpdf-1.0.0		593	19 months	sutt	rename
▸ mcfm-6.0		902	18 months	sutt	fix cmd line arguments
▸ mcfm-6.6		1079	15 months	sutt	fix broken frizione cone epsilon setting
▸ mcfm-bridge		1296	12 hours	sutt	fix runmcfm
▸ mcfm-patch		1298	12 hours	sutt	update makefile
▸ nlojet-4.0.1		589	19 months	sutt	rename
▸ obsolete		367	23 months	sutt	move tags to obsolete directory
▸ sherpa		1268	5 months	tcarli	added steering for ATLAS ttbar+jets
▸ utils		1219	8 months	sutt	improve tagging and preparation
▸ web		1182	9 months	sutt	modify news bg colours

Note: See [TracBrowser](#) for help on using the repository browser.



Powered by Trac 1.0.1
By Edgewall Software.

Visit the Trac open source project at
<http://trac.edgewall.org/>

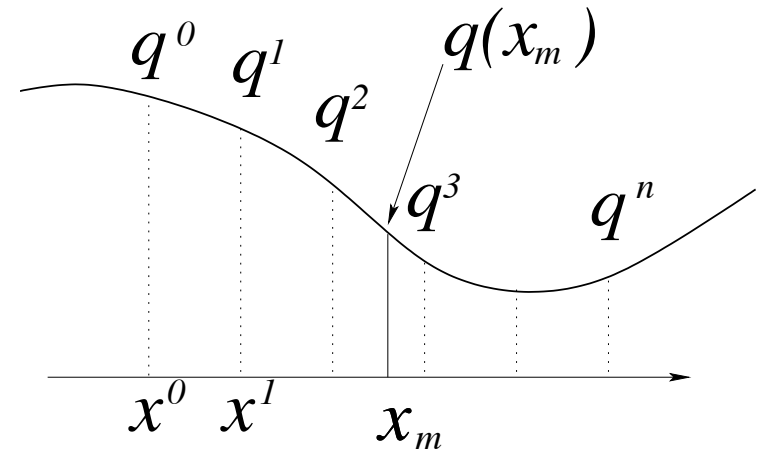
Recap of the Numerical Technique

- For a calculation of a cross section from $m = 1 \dots N$ weights, w_m , from a Monte Carlo integration with momentum fraction x_m and we wish to form the product

$$\sum_m w(x_m) q(x_m)$$

- but we can interpolate our function $q(x_m)$

$$q(x_m) \approx \sum_i q^{(i)} I^{(i)}(x_m - x^{(i)})$$



- such that

$$\begin{aligned} \sum_m w(x_m) q(x_m) &\approx \sum_i q^{(i)} \sum_m w(x_m) I^{(i)}(x_m - x^{(i)}) \\ &\approx \sum_i q^{(i)} W^{(i)} \end{aligned}$$

Recap of the Numerical Technique in DIS

- For a calculation of a cross section from $m = 1 \dots N$ weights, w_m , from a Monte Carlo integration with momentum fraction x_m at scale Q^2 ...

$$\begin{aligned}
 d\sigma &= \sum_p \sum_{m=1}^N w_m^{(p)} \left(\frac{\alpha_s(Q_m^2)}{2\pi} \right)^p q(x_m, Q_m^2) \\
 &= \sum_p \sum_{ij} q(x_{(i)}, Q_{(j)}^2) \left(\frac{\alpha_s(Q_{(j)}^2)}{2\pi} \right)^p \sum_m^N w_m^{(p)} I_i^x(x_m) I_j^{Q^2}(Q_m^2) \\
 &= \sum_p \sum_{ij} q(x_{(i)}, Q_{(j)}^2) \left(\frac{\alpha_s(Q_{(j)}^2)}{2\pi} \right)^p W_{ij}^{(p)}
 \end{aligned}$$

- The $I_i^x(x_m)$ and $I_j^{Q^2}(Q_m^2)$ are interpolating function coefficients for the terms on the grid nodes in x_m and Q^2 .

Proton-Proton Collisions

- For pp collisions need an extra dimension for the PDF of the second colliding hadron

$$d\sigma = \sum_p \sum_{m=1}^N w_m^{(p)} \left(\frac{\alpha_s(Q_m^2)}{2\pi} \right)^p q_1(x_{1m}, Q_m^2) q_2(x_{2m}, Q_m^2)$$

- But there is an implicit summation over parton flavours. Make use of symmetries in the matrix elements to use a vector, of independent weight such that $k = 1 \dots M$ and

$$\sum_{ij=q,\bar{q},g} w_{ij} q_{1i}(x_1) q_{2j}(x_2) = \sum_{k=1}^M w^{(k)} F^{(k)}(x_1, x_2)$$

- so that we have

$$d\sigma = \sum_p \sum_{k=1}^M \sum_{m=1}^N w_m^{(p)(k)} \left(\frac{\alpha_s(Q_m^2)}{2\pi} \right)^p F_m^{(k)}(x_{1m}, x_{2m}, Q_m^2)$$

- Which we can place on a grid in the same way as for DIS.

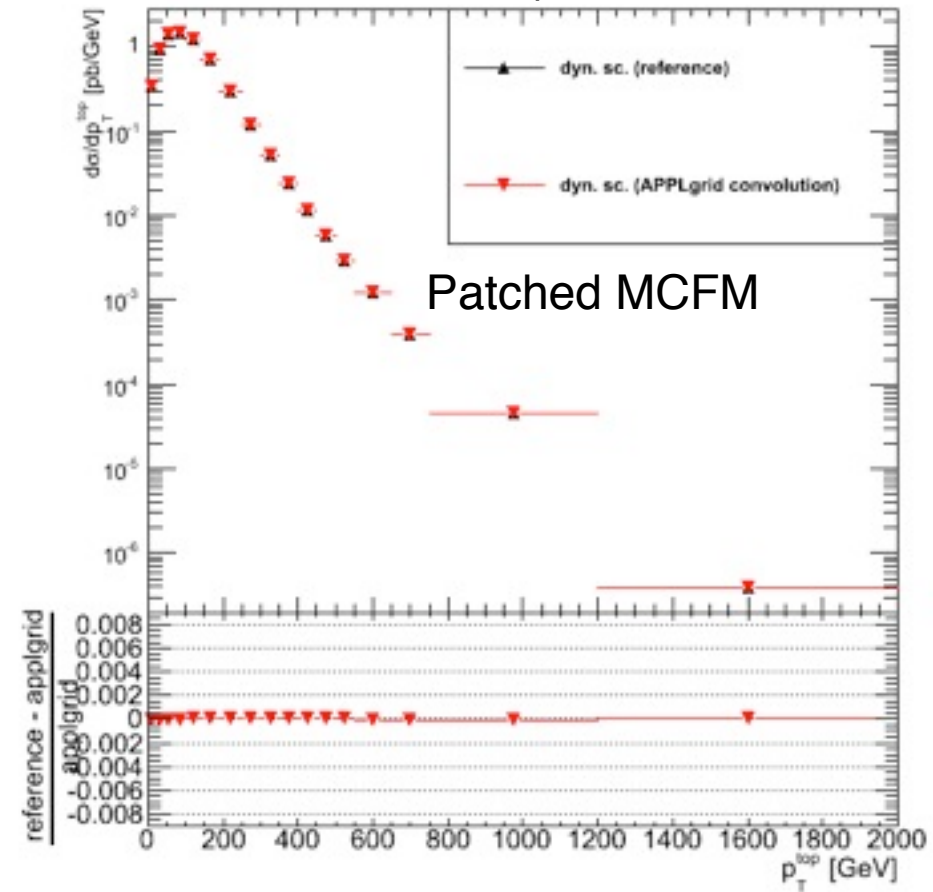
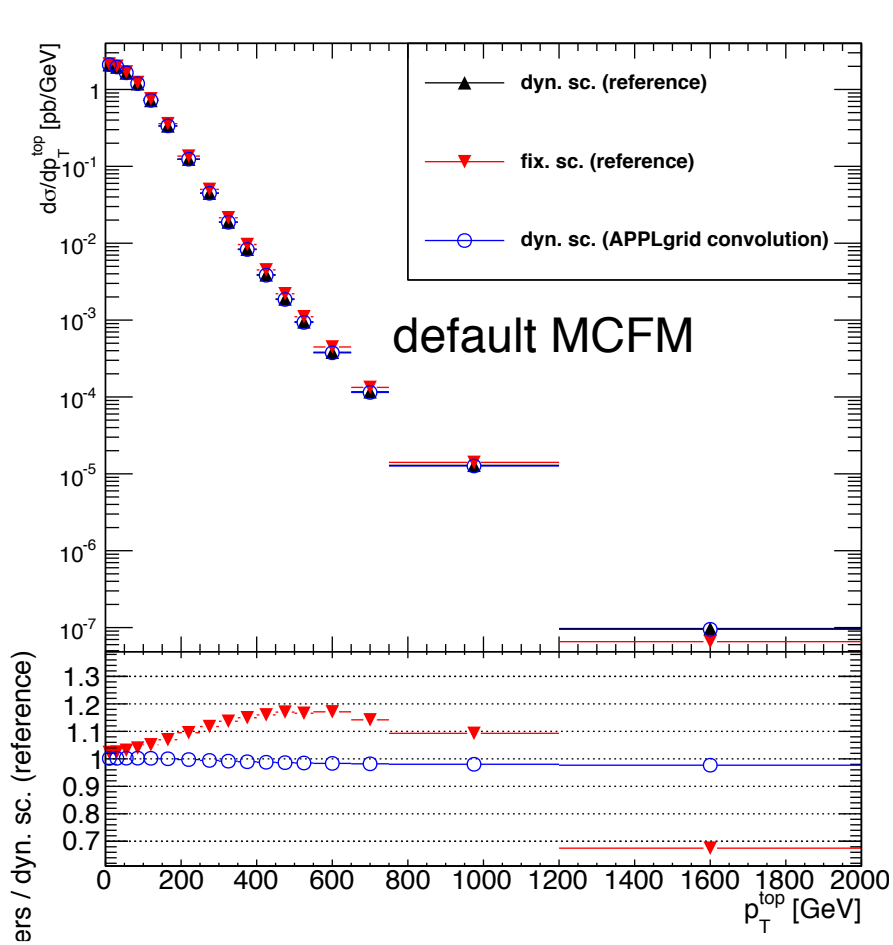
MCFM interface

- Current version 6.8 tested with W, Z production, QQ production (bbar, ttbar) W, Z + jets, W+c, prompt photon production
- Since W and Z production in APPLgrid has been around for a good while and is well known - won't discuss here
- Due largely to the efforts of Pavel Starovoitov and Alberto Guffanti since version MCFM 6.7, modifications required to interface with APPLgrid have been included in the standard MCFM - no longer any need for a custom version
- Have a new paradigm for the interface with MCFM
 - Non obtrusive hook functions based on null function pointers that can be left compiled into the standard MCFM code
 - New bridge package that includes a setup function which initialises the hook functions to external filling routines ...
 - APPLgrid interface included, or excluded simply by linking against the APPLgrid libraries, without requiring any recompilation ...
 - If you don't link against the APPLgrid libraries you get standard MCFM, no MCFM code needs to be compiled
- This paradigm is also used to great success in the aMCfast interface for aMC@NLO

Current MCFM status ...

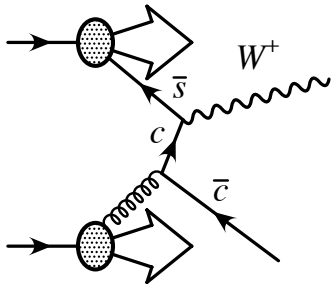
- All MCFM processes, in principle now included
- Those that we have (personally) verified ...
 - EW Boson production ...
 - Inclusive W^\pm production (processes 1, 6)
 - W^\pm + jet production (processes 11, 15)
 - Inclusive Z production (process 31)
 - Z + jet production (processes 41-43)
 - Prompt photon production (processes 280 - 286) - including fragmentation
 - Heavy flavour ...
 - $t\bar{t}$ production (semi-leptonic decay) with/without spin correlations (proc. 141, 142, 144, 145)
 - $t\bar{t}$ production (hadronic decay), with radiative corrections in t/W decays etc (proc. 146 - 151)
 - Inclusive $t\bar{t}$ production (157)
 - Inclusive $b\bar{b}$ production (158)
 - Inclusive $c\bar{c}$ production (159)
 - Combined EW+heavy flavour ...
 - W^\pm + c production (processes 13-16)

MCFM top - dynamic scale patch

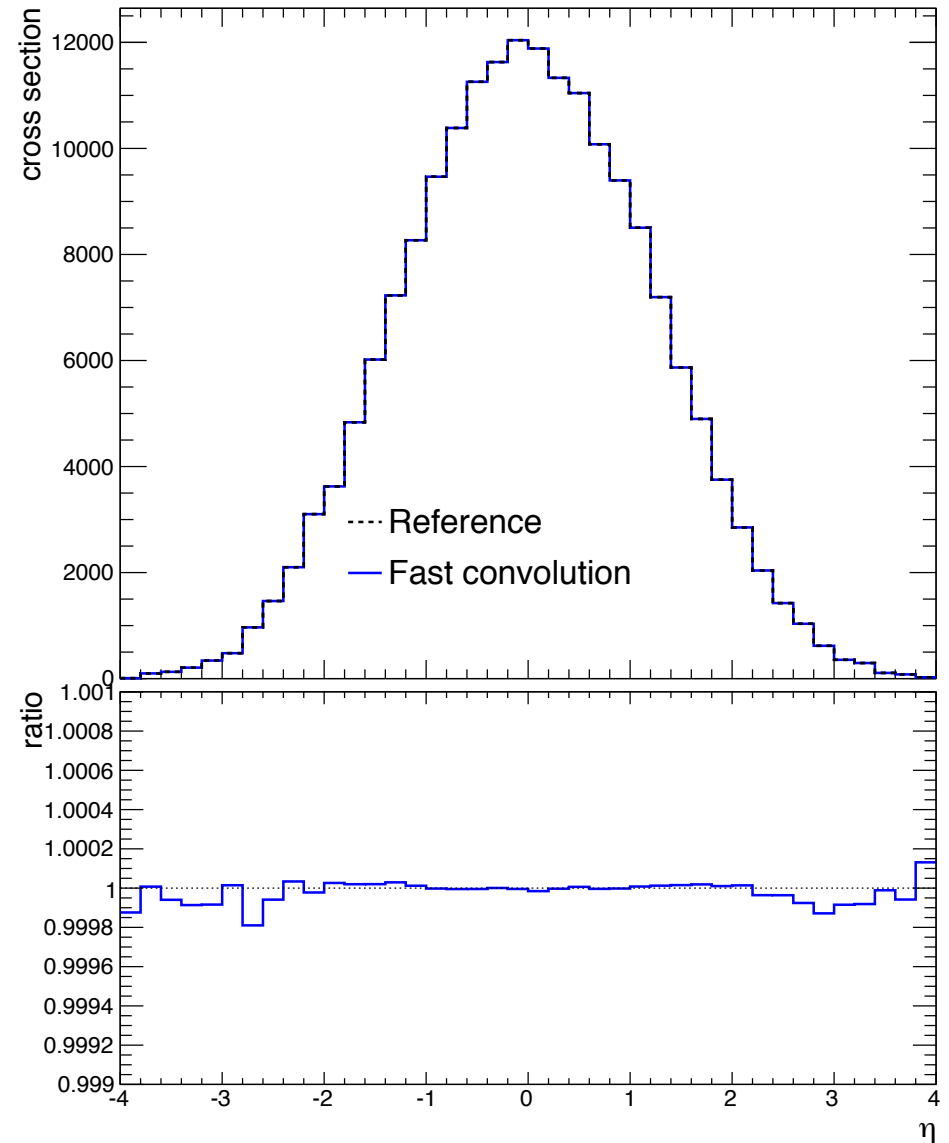


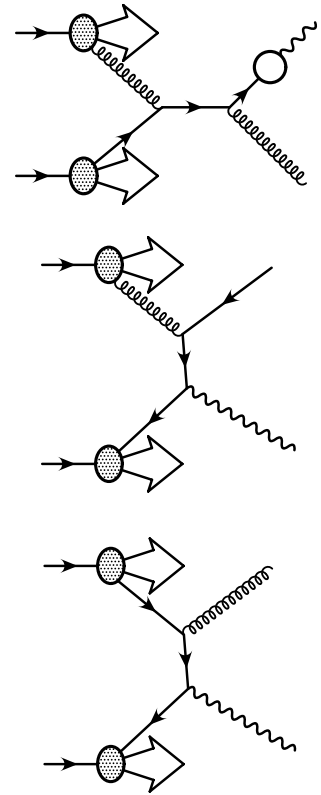
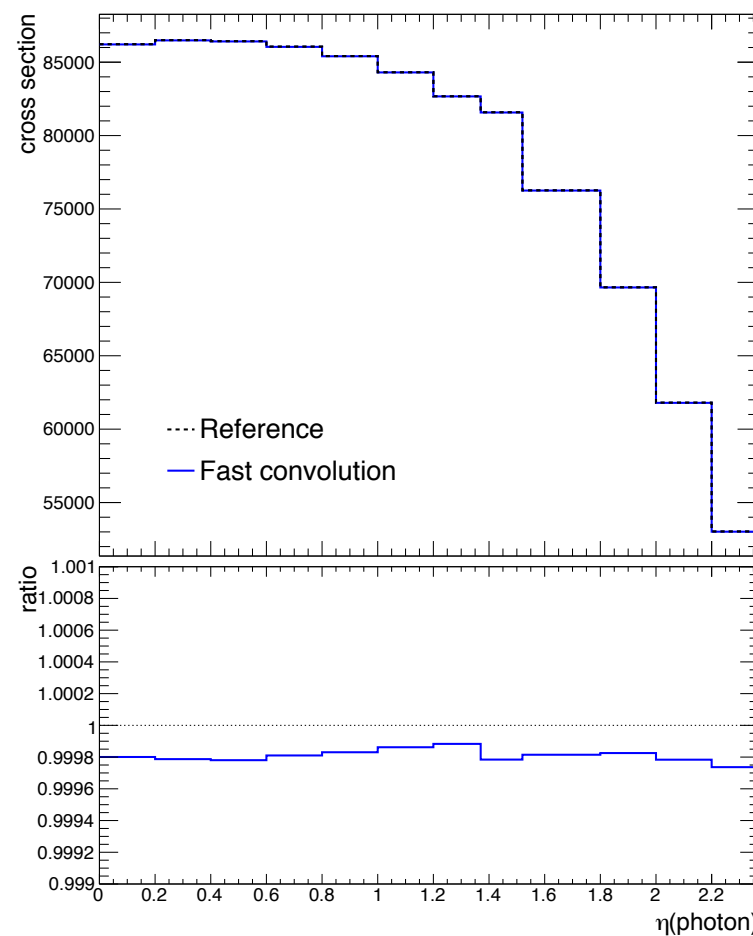
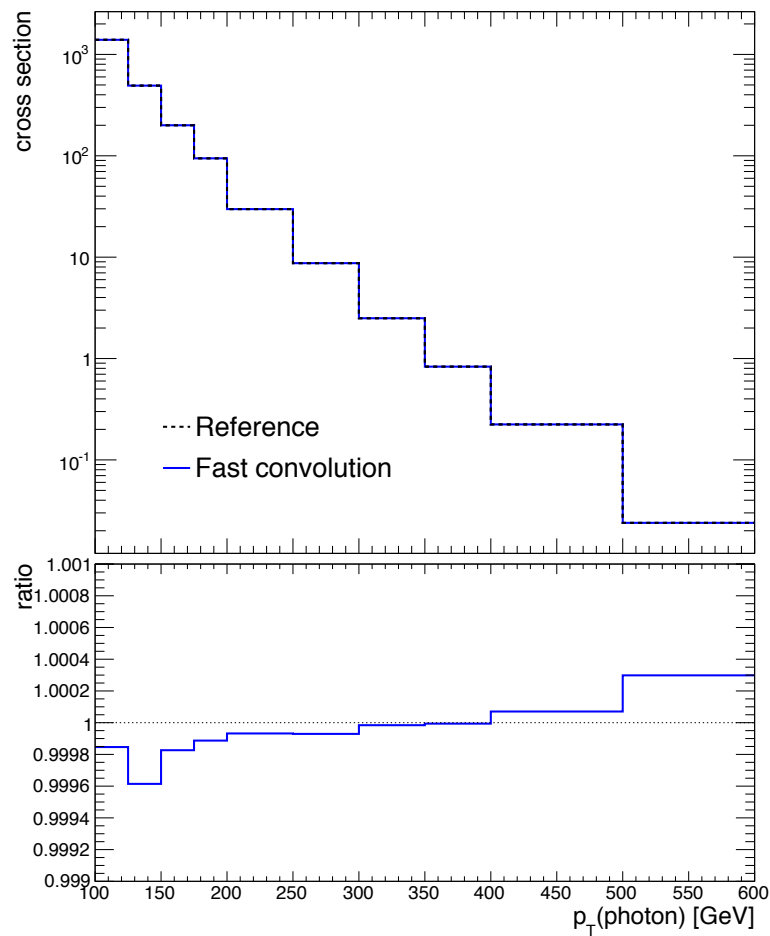
- Without the patch, small $\sim 2\%$ non-closure of the top cross section at high p_T
- With the patch closes to better than 0.1%

Just a selection of interesting processes ...



- W+charm production implemented and well tested
- Bare charm - for realistic comparison with experimental fiducial data some model of hadronisation is required





Prompt photon production (Andrey Sapronov)

- Prompt photon production at NLO + LO fragmentation process
- Small non-closure at the level of 0.02%
- Some validation underway: NLO+NLO fragmentation contribution available from JetPhox - calculating LO \rightarrow NLO fragmentation k-factors

Generic subprocesses

- For generic calculations, can use a representative **basic** configuration containing 11×11 (non-top) subprocesses
- This is rather **too large** for general use, so that we have a feature of the grids that can reduce this to the smallest number of unique subprocesses
- For example, for inclusive photon production with 121 processes, these can be reduced to 6 subprocesses for LO and 33 for NLO

```

0 6  -5 0  -3 0  -1 0  1 0  3 0  5 0
1 6  -5 5  -3 3  -1 1  1 -1  3 -3  5 -5
2 4  -4 0  -2 0  2 0  4 0
3 4  -4 4  -2 2  2 -2  4 -4
4 6  0 -5  0 -3  0 -1  0 1  0 3  0 5
5 4  0 -4  0 -2  0 2  0 4

```

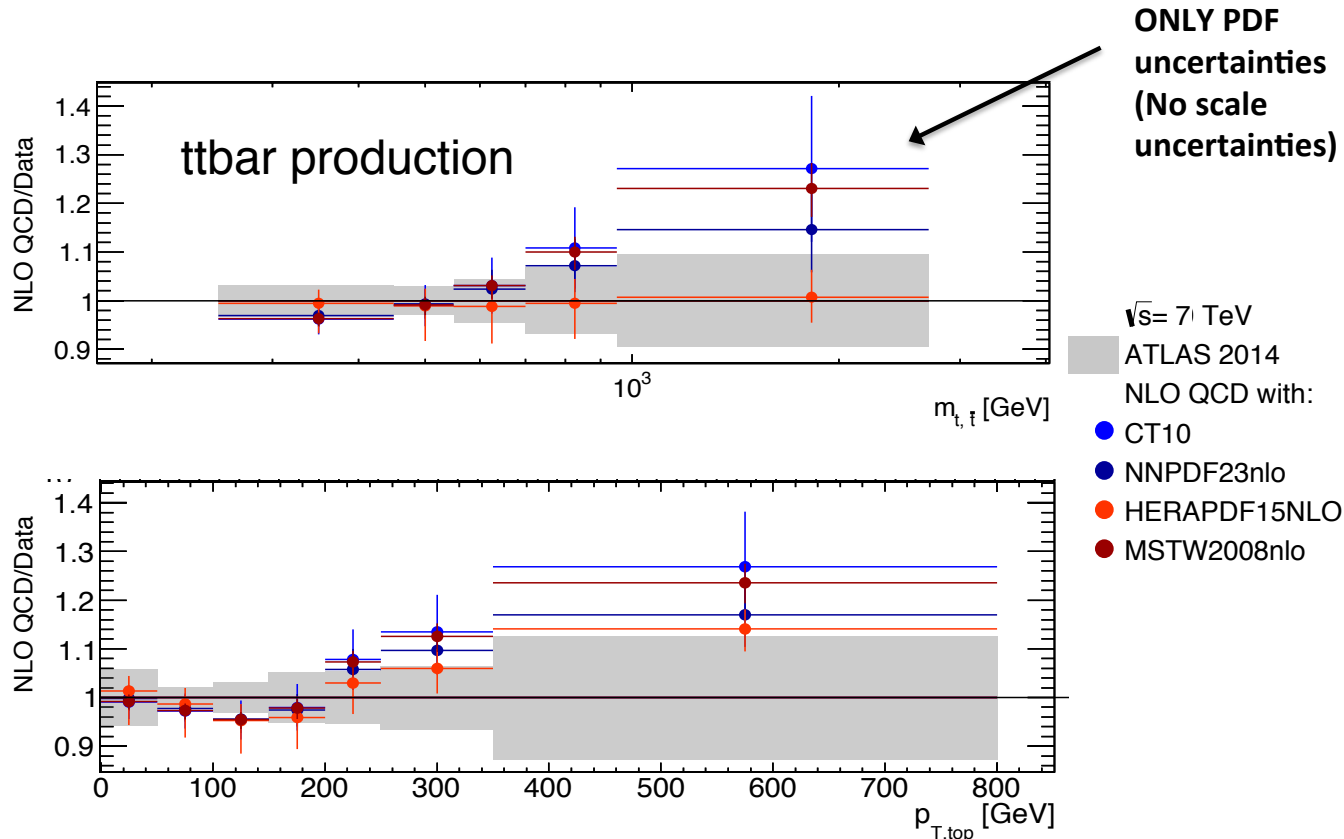
```

0 3  -5 -5  -3 -3  -1 -1
1 6  -5 -4  -5 -2  -3 -4  -3 -2  -1 -4  -1 -2
2 6  -5 -3  -5 -1  -3 -5  -3 -1  -1 -5  -1 -3
3 3  -5 0  -3 0  -1 0
4 6  -5 1  -5 3  -3 1  -3 5  -1 3  -1 5
5 6  -5 2  -5 4  -3 2  -3 4  -1 2  -1 4
6 3  -5 5  -3 3  -1 1
7 6  -4 -5  -4 -3  -4 -1  -2 -5  -2 -3  -2 -1
8 2  -4 -4  -2 -2
9 2  -4 -2  -2 -4
10 2  -4 0  -2 0
11 6  -4 1  -4 3  -4 5  -2 1  -2 3  -2 5
12 2  -4 2  -2 4
13 2  -4 4  -2 2
14 3  0 -5  0 -3  0 -1
15 2  0 -4  0 -2
16 1  0 0
17 3  0 1  0 3  0 5
18 2  0 2  0 4
19 6  1 -5  1 -3  3 -5  3 -1  5 -3  5 -1
20 6  1 -4  1 -2  3 -4  3 -2  5 -4  5 -2
21 3  1 -1  3 -3  5 -5
22 3  1 0  3 0  5 0
23 3  1 1  3 3  5 5
24 6  1 2  1 4  3 2  3 4  5 2  5 4
25 6  1 3  1 5  3 1  3 5  5 1  5 3
26 6  2 -5  2 -3  2 -1  4 -5  4 -3  4 -1
27 2  2 -4  4 -2
28 2  2 -2  4 -4
29 2  2 0  4 0
30 6  2 1  2 3  2 5  4 1  4 3  4 5
31 2  2 2  4 4
32_2 2 4  4 2

```

- NB: This **may not** be the minimal number for which symmetries of the matrix elements could be used, but it is a significant reduction none-the-less

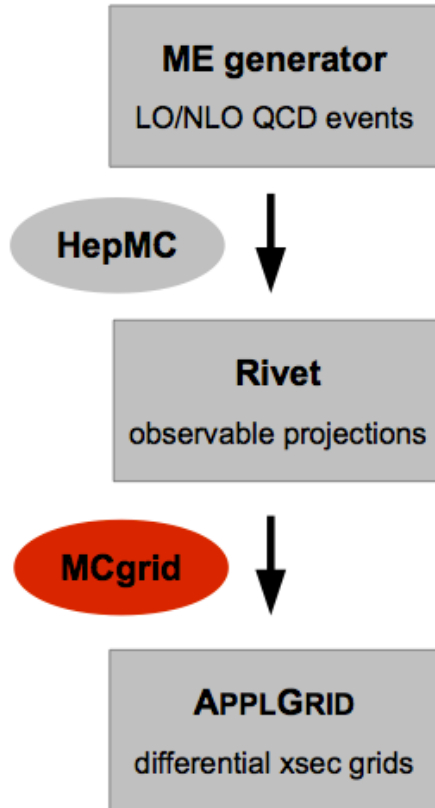
Sherpa - Native APPLgrid - Sherpa interface



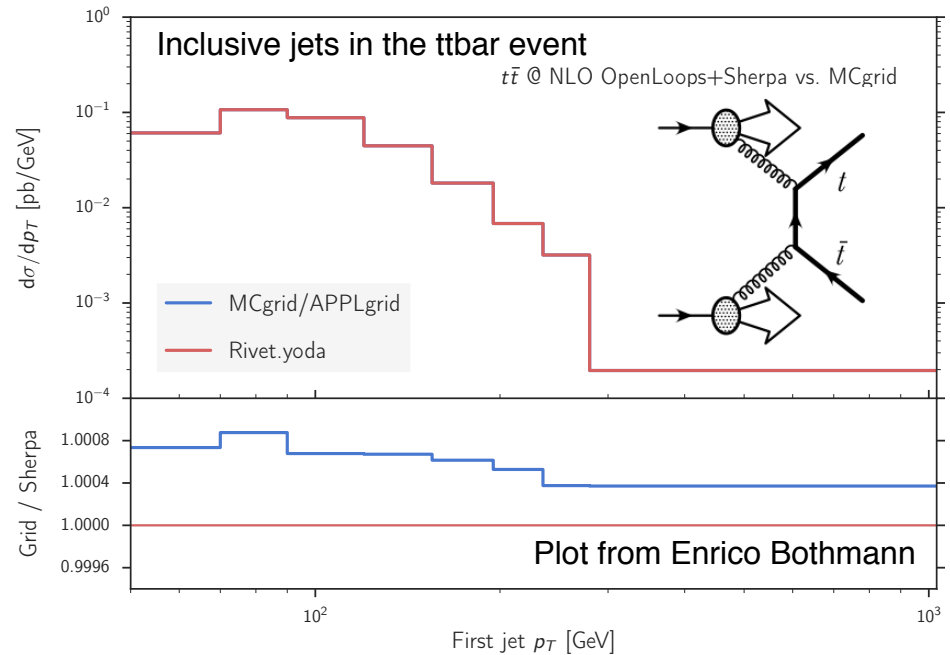
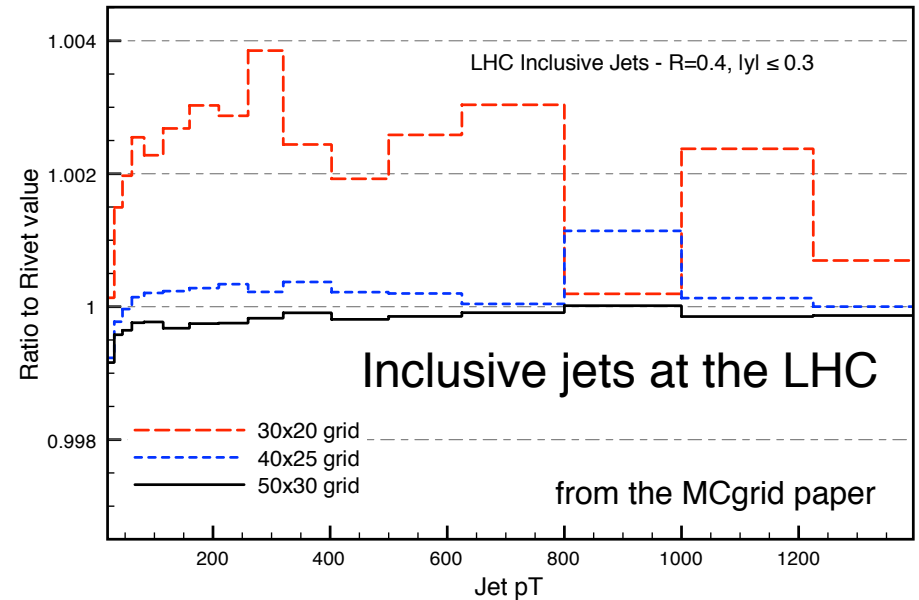
- Uses the native Sherpa output ntuple: Tancredi Carli, Cameron Embree MS
- Uses the generic process decomposition
- Can be used for any fixed order NLO sherpa process
- **Caveat:** when generating weights for individual parton-parton subprocesses, in order to get agreement with reference at low statistics, need to use the full 121 parton-parton luminosities
 - Imposing known symmetries in the matrix element to fill the grid with fewer subprocesses provides more precise grids, but at the cost of better than statistical agreement with the reference histogram at low statistics

MCgrid Sherpa interface - Del Debbio et al

- Written by Del Debbio, Hartland and Schumann
arXiv:1312.4460. <http://mcgrid.hepforge.org>
- Available for all fixed order NLO processes using the generic PDF decomposition



- MCgrid works as a Rivet plugin using the HepMC event record



aMCfast - A fast interface between MadGraph5_aMC@NLO and APPLgrid

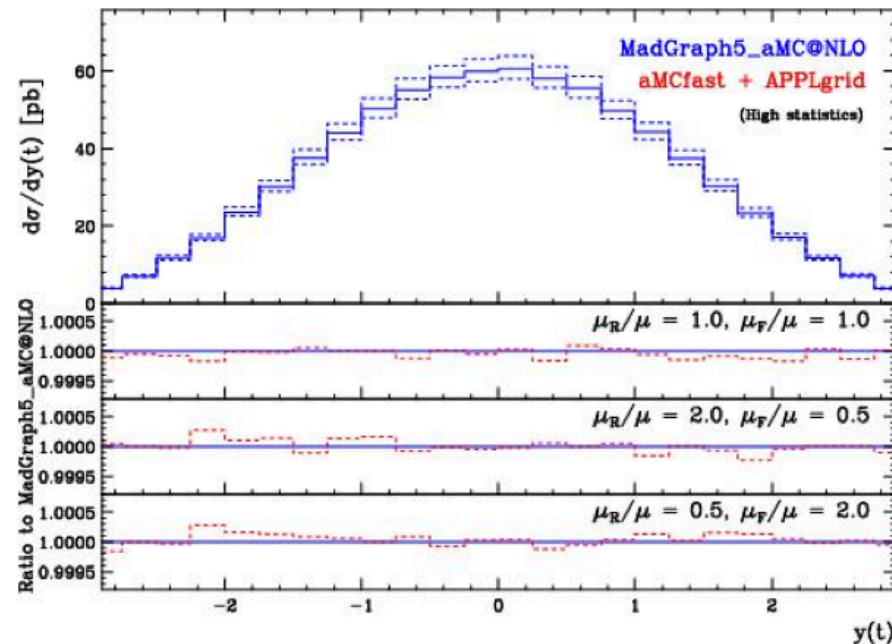
- aMCfast Home
- References
- Download and installation
- Instructions to run the code
- Analysis file examples
- Talks
- Contact

aMCfast [arXiv:1406.7693] is an *automated interface* which bridges the automated cross section calculator **MadGraph5_aMC@NLO** [arXiv:1405.0301] with the fast interpolator **APPLgrid** [arXiv:0911.2985].

The chain **MadGraph5_aMC@NLO** – **aMCfast** – **APPLgrid** will allow one to include, in a straightforward manner, any present or future LHC measurement in an NLO global PDF analysis.

The basic idea behind the use of these three codes is that of computing user-defined observables relevant to arbitrary processes, and to represent them in terms of look-up grids, which can be accessed at later times, and used to obtain predictions for such observables with any PDFs. This a-posteriori computation is both accurate and very fast. Contrary to other **APPLgrid** application, factorisation scale variation can be performed without linking to any third-party code.

The following representative figures show the rapidity of the top quark in top-pair production and the lepton-pair invariant mass in dilepton production in association with one jet at the 14 TeV LHC. For more details, please refer to the original [publication](#).



- Valerio Bertone, Rik Frederix, Stefano Frixione, Juan Rojo, MS

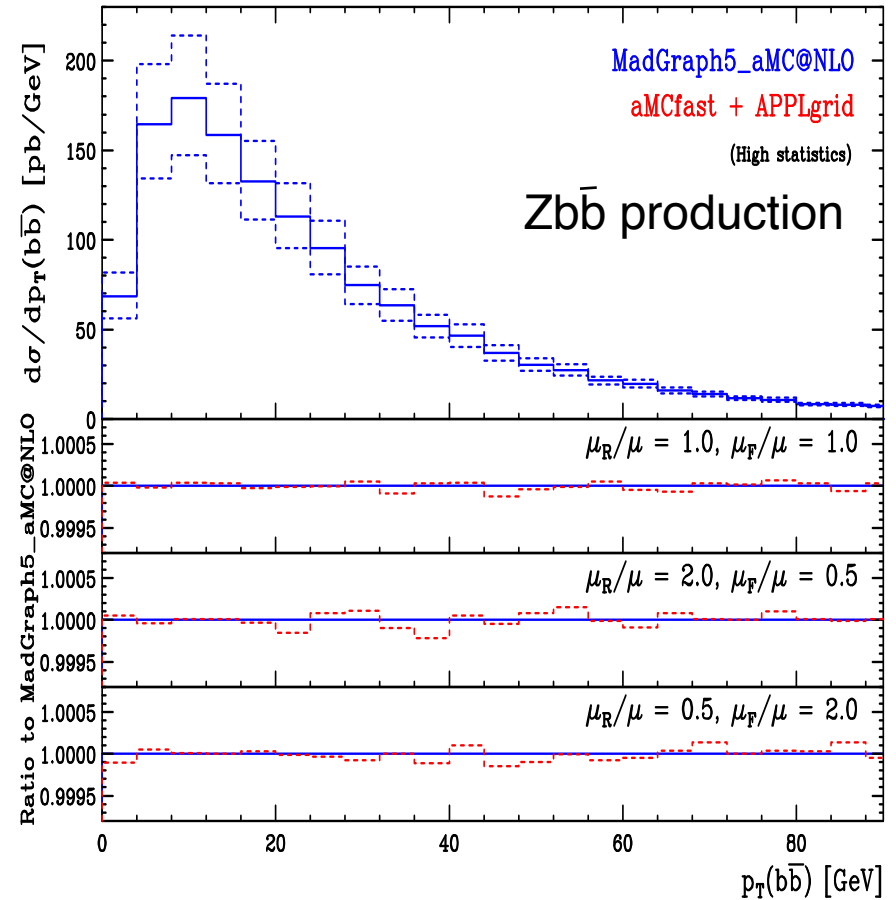
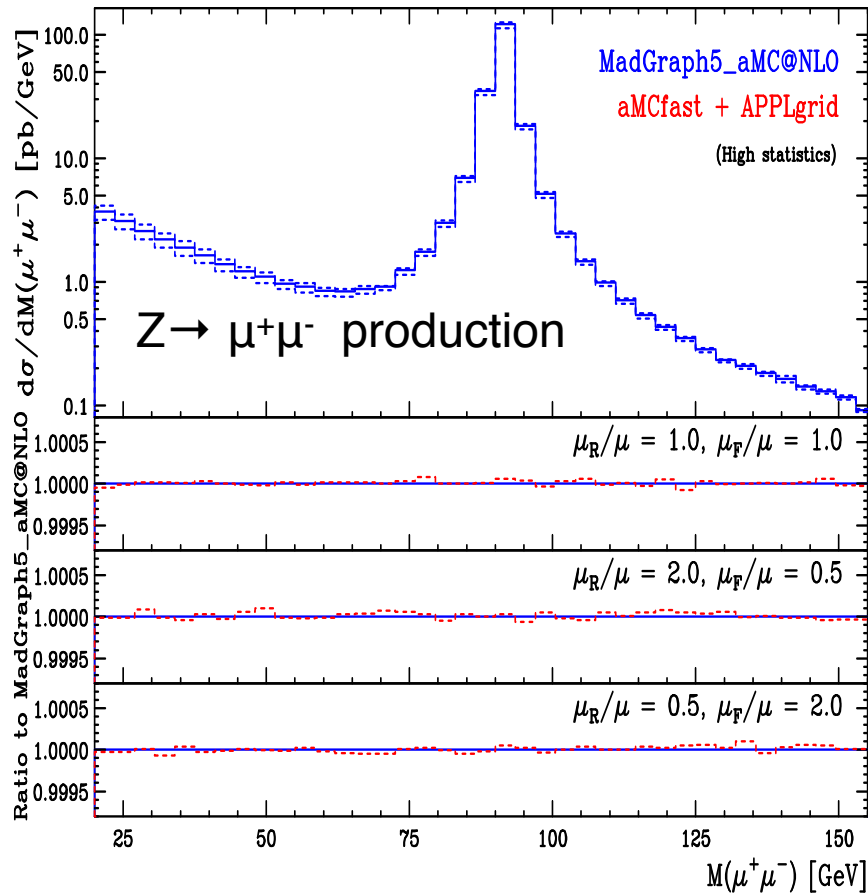
aMCFast interface MC@NLO interface

- Implementation of aMC@NLO standalone calculations uses 4 sets of weights - one at tree level and three for the NLO part

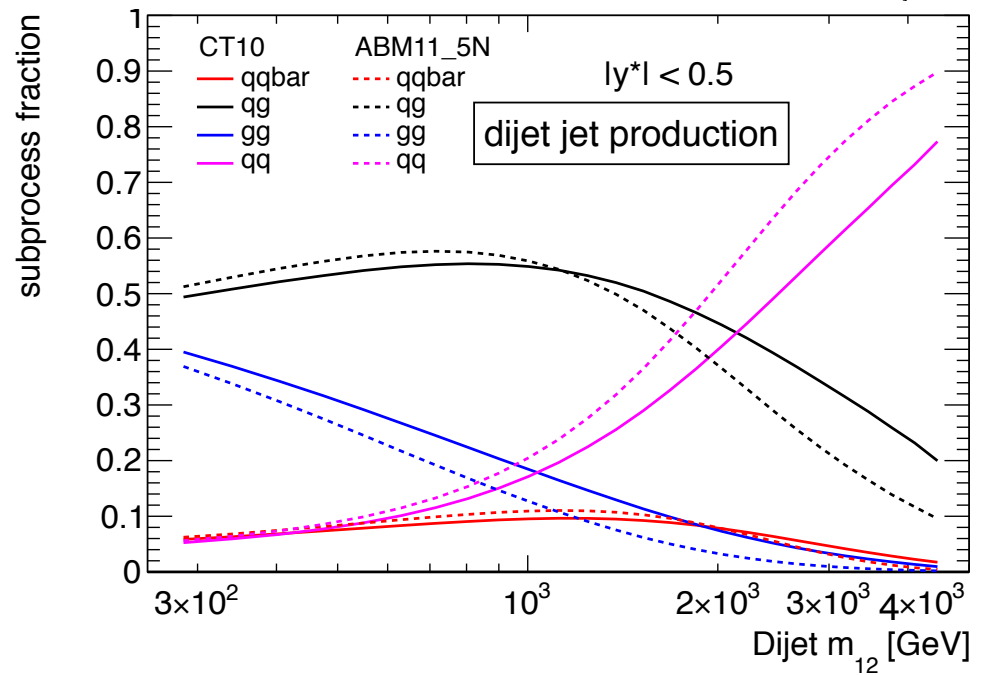
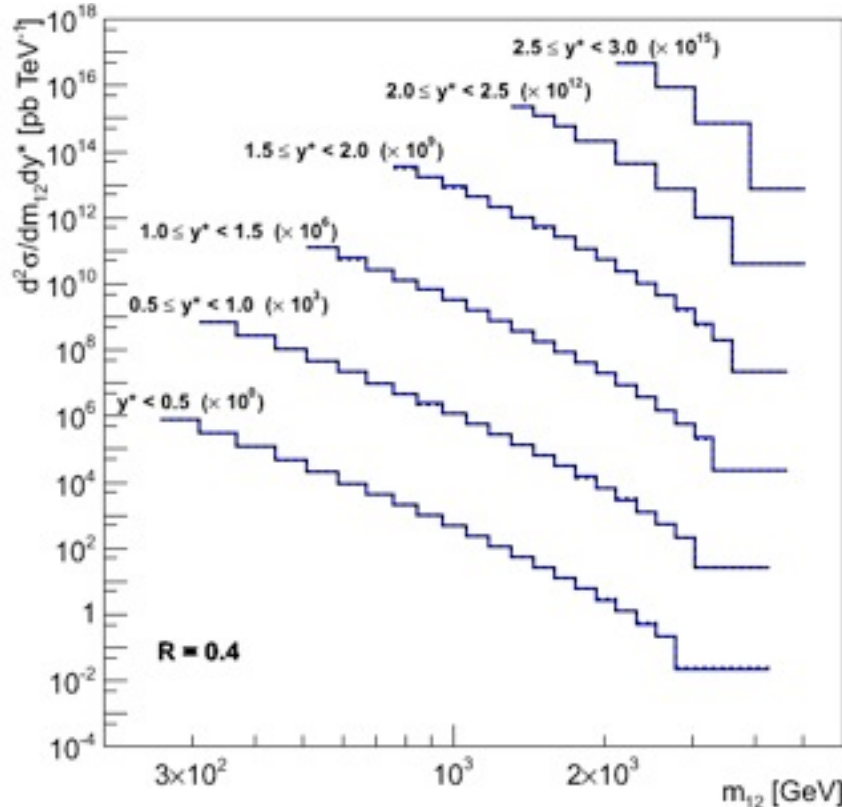
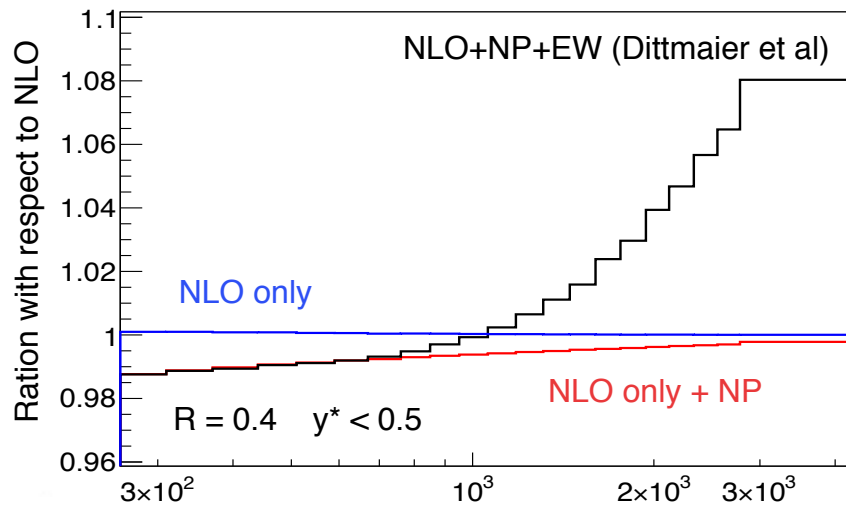
$$d\sigma = \alpha_s(Q)^n \left[W^B + \alpha_s(Q) \left[W^0 + W^R \log \frac{\mu_R}{Q} + W^F \log \frac{\mu_F}{Q} \right] \right]$$

- Minor internal modifications to the grid class to make use of additional weight grids
- Implements all fixed order NLO calculations within aMC@NLO
- Using generic subprocesses automatically configured using the information provided by aMC@NLO and the new "MCFM" interface paradigm discussed earlier
- Makes extensive use of the grid combination utilities to combine grids after generation

aMCfast interface to aMC@NLO



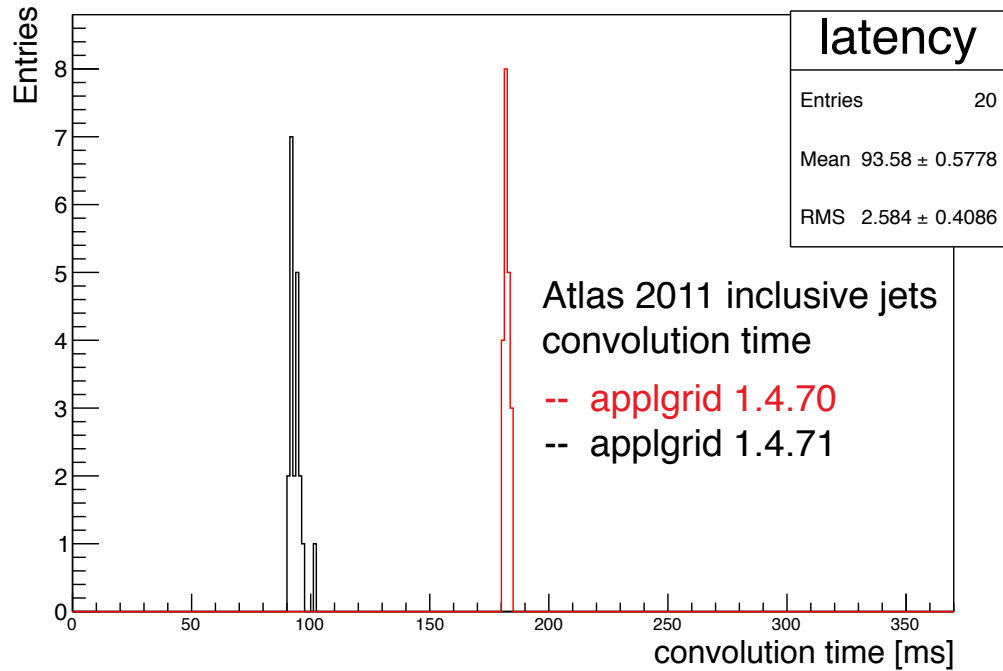
- Agreement to better than 0.05%
- More details in presentation from Valerio



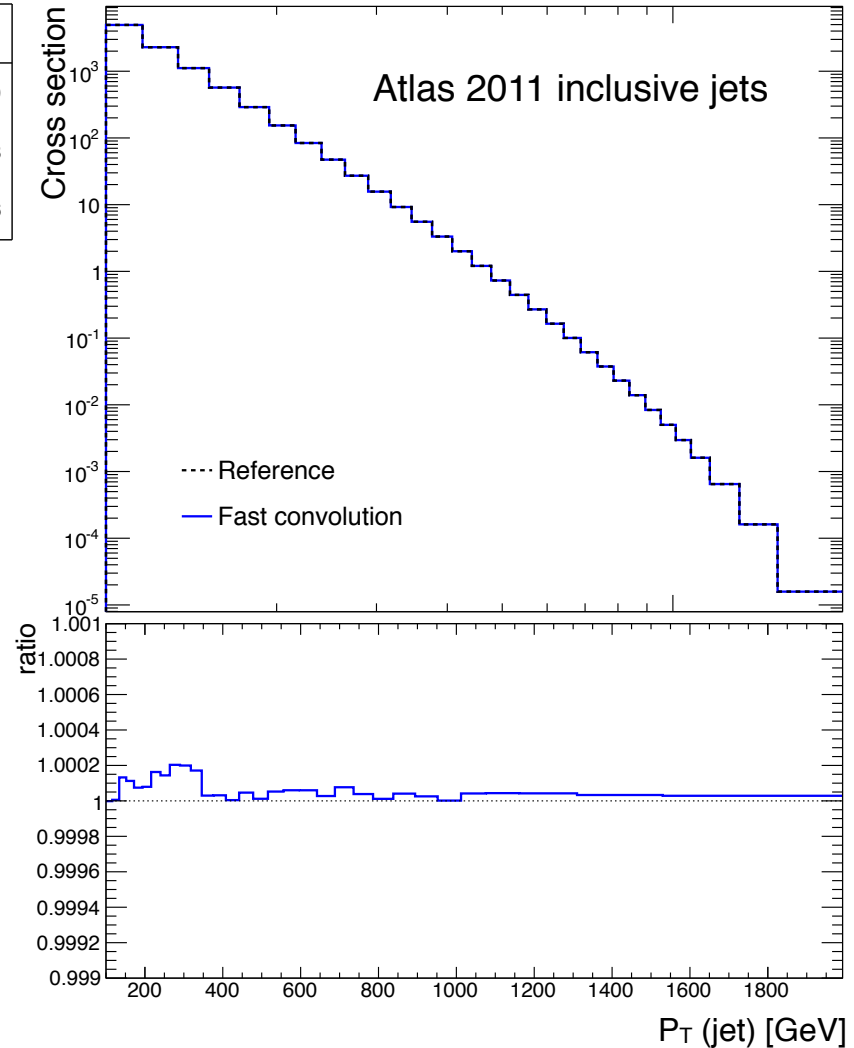
Multiplicative corrections: Example dijet production

- Grids can include eg non-perturbative and EW corrections
- Corrections can be applied independently, or in any combination
- Single multiplicative terms for any give correction is not ideal
 - Move towards subprocess dependent corrections
 - Although not ideal - better dependence than current single k-factor approach

Upcoming developments



- Many new developments are on their way
 - Just a taster here - significantly faster convolution implementation
 - Aspects of internal grid structure being reimplemented, new utilities
- Several new calculations being implemented, watch this space for details



Spectrum

STOP VIOLENCE AGAINST WOMEN

Back Forward | ☆ | Reload | Google | Home | Bookmarks | Convert

HOME PLOT DOWNLOADS

Spectrum

PROTON-PROTON CROSS-SECTION ANALYSIS SOFTWARE
ACCESS TO MULTIPLE PDF AND GRID COMBINATIONS
DATASETS FROM ATLAS, CMS, HERA, AND MORE

[MAKE A PLOT](#)

QUANTUM CHROMODYNAMICS MADE EASY

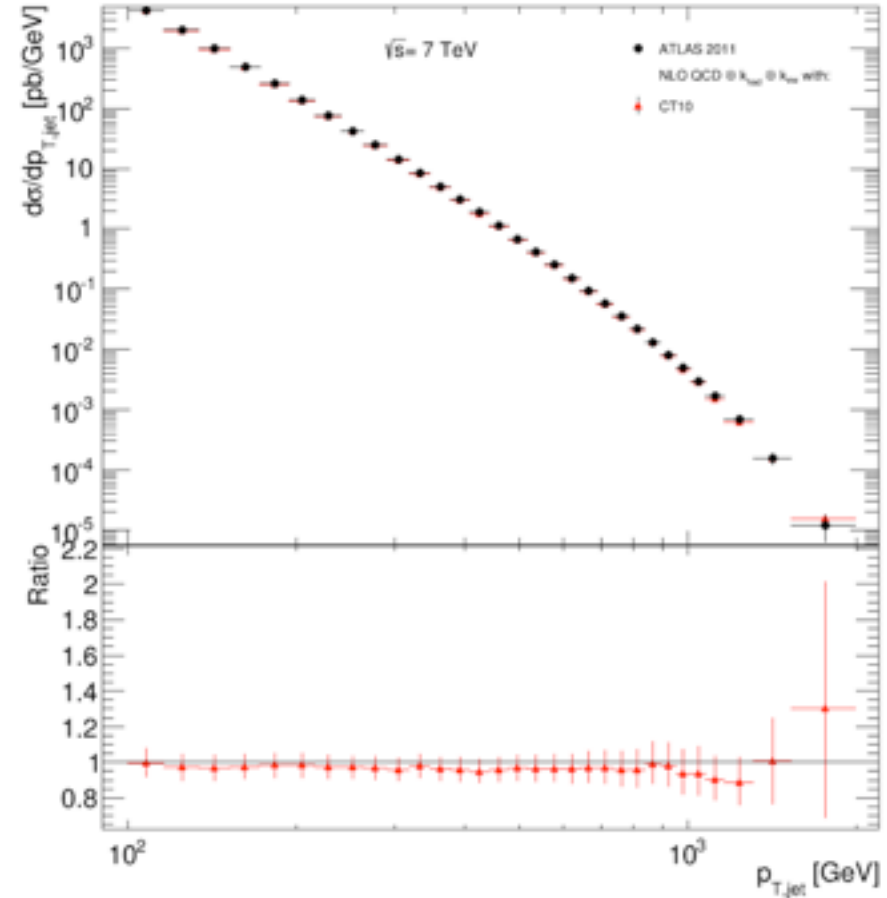
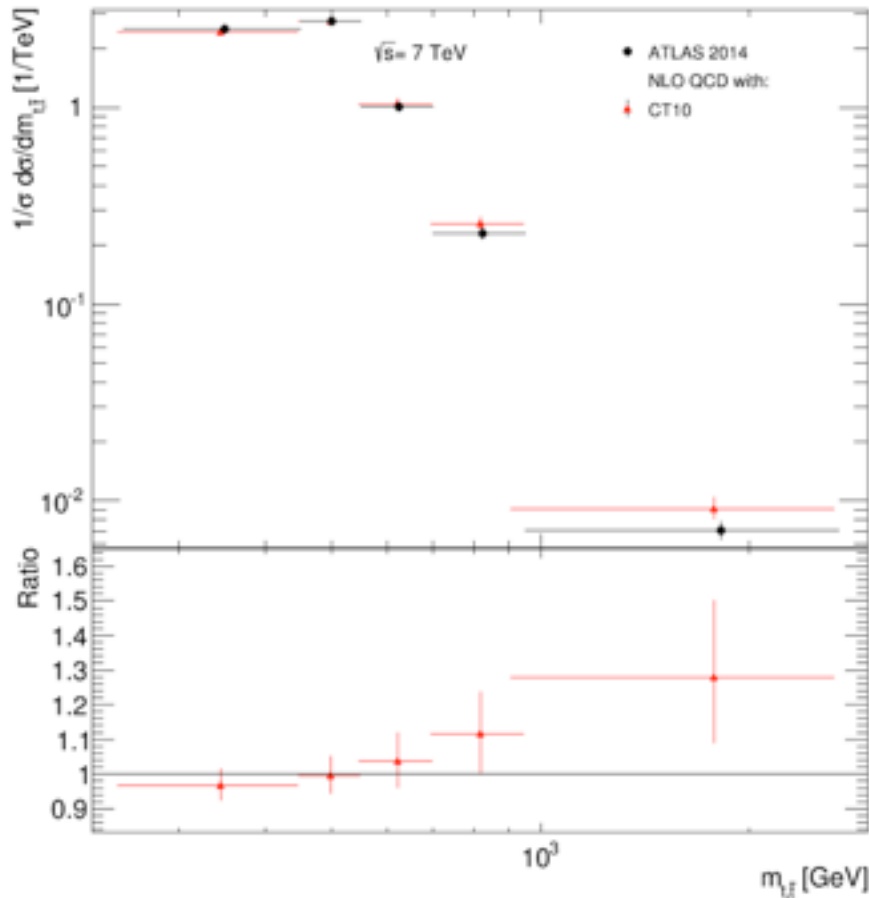
- Start Quickly**
A number of pre-configured plots are available for quick access to the most commonly used plots. This option is presently being worked out....
- Configure With Ease**
Configuring a Spectrum plot is simple due to the dynamic user interface and configuration help dialogues.
- Choose Your Grids**
Select from a large collection of supported Grids and PDFs to perform your convolution.
- Grab Your Plot**
Download your plot in multiple formats including PNG, JPEG, TIFF, GIF, and PDF.
- Expert Mode**
Use the web interface or download the Spectrum source and configure in detail with easy to use Steering Files.

CONFIGURATION MADE SIMPLE

Configuring Spectrum plots couldn't be simpler. Select your plot type, data set, grids, PDFs, and then configure the styles for each graph like the marker style, color, fill style, etc. with the interactive configuration menus.

- New plotting web utility for simpler production of comparisons of data with calculations code
- Still under development but should be available soon
- From Tancredi Carli and Joe Gibson

Spectrum plotter



- Several sets of cross section data and grids are available
 - Compare data and calculation with any choice of PDF, with or without hadronisation, EW corrections etc
 - Aim is to try to collect all available processes and grids - grids will be downloadable, with appropriate authorship attribution etc, so please donate your grids!

Outlook

- The APPLgrid project is now reasonably mature with an increasingly extensive portfolio of cross section processes available
- An increasingly large user base is developing for both the fast convolution code and grid generation
- Developments are underway for interfaces with new calculation code
 - Cross community discussions might be useful for more extensive and rapid progress
- We are always open to new suggestions or requests - particularly if people want to contribute to the project

