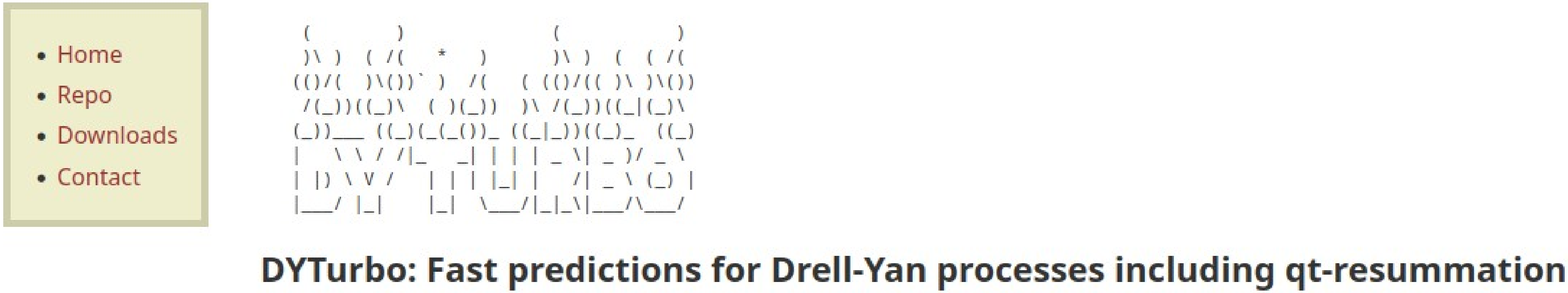
DYTurbo interface for xFitter and its applications

Stefano Camarda

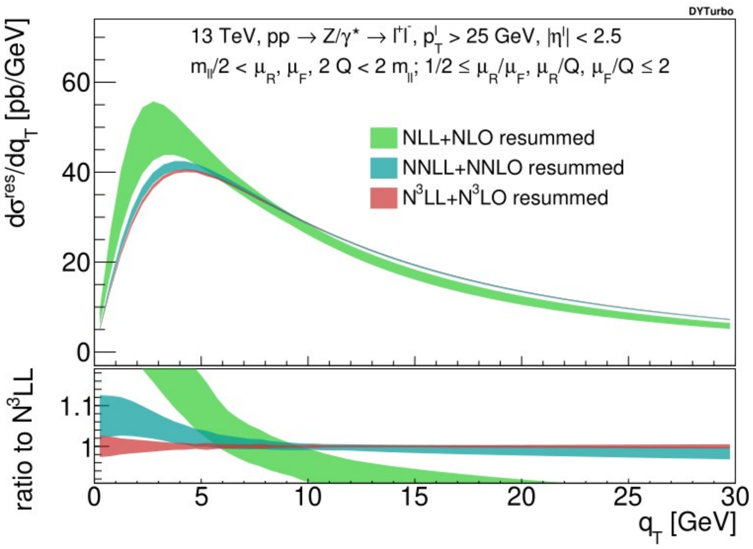
xFitter workshop – 4th May 2023

DYTurbo project

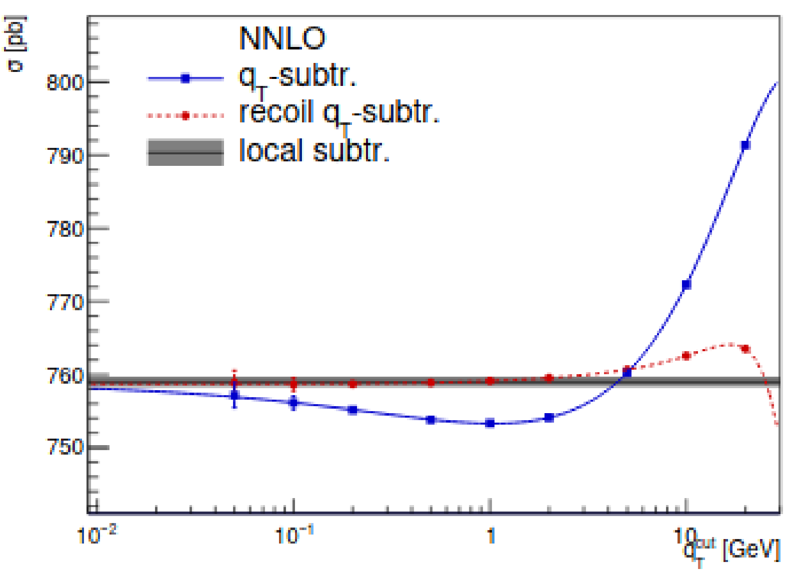


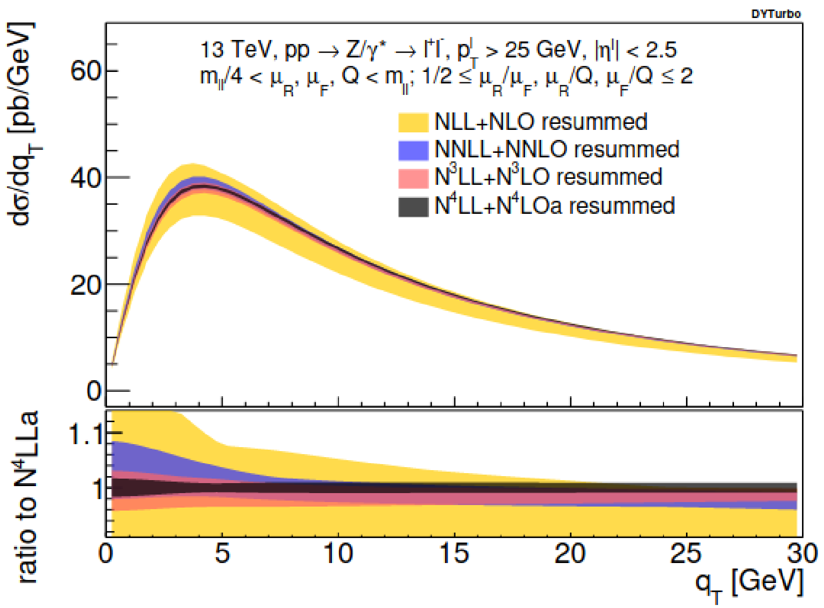
- Public code available on HEPForge (https://dyturbo.hepforge.org/)
- Original reference:
  - DYTurbo: Fast predictions for Drell-Yan processes, arXiv:1910.07049
- Recent developments:
  - Drell–Yan lepton-pair production: q_T resummation at N3LL accuracy and fiducial cross sections at N3LO, arXiv:2103.04974
  - Fiducial perturbative power corrections within the q_T subtraction formalism, arXiv:2111.14509
  - Drell-Yan lepton-pair production: qT resummation at approximate N4LL+N4LO accuracy, arXiv:2303.12781

Recent DYTurbo results



- N3LL q_T resummation (since v1.1)
- Fiducial power corrections (since v1.2)
- N3LO fiducial cross sections (depend on the availability of external V+jet NNLO predictions)
- N4LL q_T resummation (since v1.3)





xFitter interface

- DYTurbo-xFitter interface to version 1.2 is available in master
- Interface to version 1.3 is available in
  - dyturbo-interface-v1.3 → ready to be merged
  - alphas-scan → experimental
- yaml settings currently include order, scales, non-perturbative parameters, debug options

```
# DYTurbo:
order: 2     # Perturbative order in QCD: 0 for LL, 1 for NLL+NLO, 2 for NNLL+NNLO, 3 for NNNLL+NNLO
muR: 1.      # Renormalization scale
muF: 1.      # Factorization scale
muRes: 1.    # Resummation scale
blim: 1.5    # bmax value for the bstar prescription
g1: 1        # Universal Gaussian non-perturbative form factor
g2: 0        # Q-dependent Gaussian non-perturbative form factor
g3: 0        # tau-dependent Gaussian non-perturbative form factor
g1x: 0       # x-dependent Gaussian non-perturbative form factor
g1a: 0       # x-dependent Gaussian non-perturbative form factor
g1b: 0       # x-dependent Gaussian non-perturbative form factor
lambda: 1    # Transition to exponential
q: 0         # Quartic term
debug: 0     # verbosity flag
```

xFitter interface

- Many more options in the experimental branch

```
# DYTurbo:
order: 2    # Perturbative order in QCD: 0 for LL, 1 for NLL+NLO, 2 for NNLL+NNLO, 3 for NNNLL+NNLO
muR: 1.     # Renormalization scale
muF: 1.     # Factorization scale
muRes: 1.   # Resummation scale
blim: 1.5   # bmax value for the bstar prescription
g1: 1       # Universal Gaussian non-perturbative form factor
g2: 0       # Q-dependent Gaussian non-perturbative form factor
g3: 0       # tau-dependent Gaussian non-perturbative form factor
gx: 0       # x-dependent Gaussian non-perturbative form factor
ga: 0       # x-dependent Gaussian non-perturbative form factor
gb: 0       # x-dependent Gaussian non-perturbative form factor
a2: 0.06    # DMW non-perturbative form factor
ap: 0.06    # DMW non-perturbative form factor
Q0: 1       # Reference scale at which the g2 contribution is zero
g1a: 0       # x-dependent Gaussian non-perturbative form factor
g1b: 0       # x-dependent Gaussian non-perturbative form factor
lambda: 1
gA: 0
gB: 0
NA: 0
sigmaA: 1
alphaA: 1
NB: 0
sigmaB: 1
alphaB: 1
q: 0     # quartic term
debug: 0
```

xFitter interface

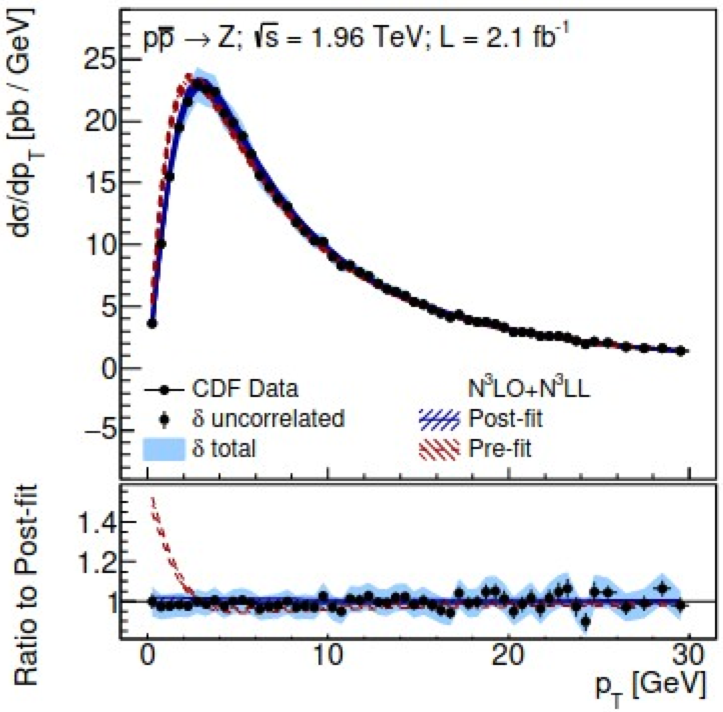
- Operational modes for Z/γ*
  - Full-lepton phase space cross sections in p_T,y,m: DYTurbo can provide the full fixed order or resummed prediction

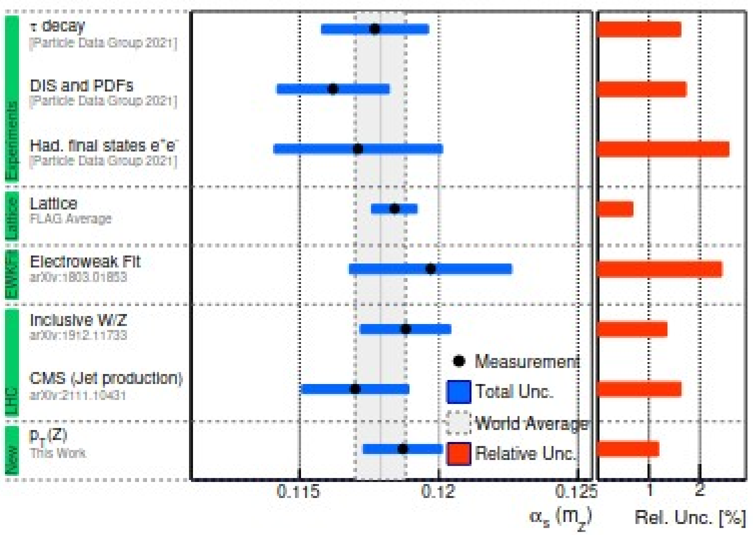
```
&Data
  Name = 'CDF ZPT 1.96 TEV'
  IndexDataset =   101010
  Reaction  = 'NC ppbar'

  TheoryType     = 'expression'
  TermName = 'A','K','C'
  TermSource = 'DYTurbo','KFactor','KFactor'
  TermInfo = 'FileName=datafiles/tevatron/cdf/wzProduction/1207.7138/cdf196-n4ll.in',
             'FileName=datafiles/tevatron/cdf/wzProduction/1207.7138/qedisr.txt:FileColumn=3',
             'FileName=datafiles/tevatron/cdf/wzProduction/1207.7138/n3lo-afactor-r1-f1-q1.txt:FileColumn=3'
  TheorExpr= '(A+C)*K/1000'
```

  - Fiducial phase space cross sections in p_T,y,m: include V+jet APPLgrid, DYTurbo computes the rest

```
&Data
  Name = 'ATLAS Z pT 8 TeV'
  IndexDataset =   20
  Reaction =     'NC pp'

  TheoryType     = 'expression'
  TermName = 'A1','A2'
  TermSource = 'DYTurbo','APPLgrid'
  TermInfo = 'FileName=data/zpt8tev/dyturbo/z-8tev-nnll.in',
             'GridName=data/zpt8tev/applgrid/grid-40-6-15-3-Zjet_41_zpt8tev_peak.root'
  TheorExpr= 'A1+A2'
```

Applications: CDF Z p_T

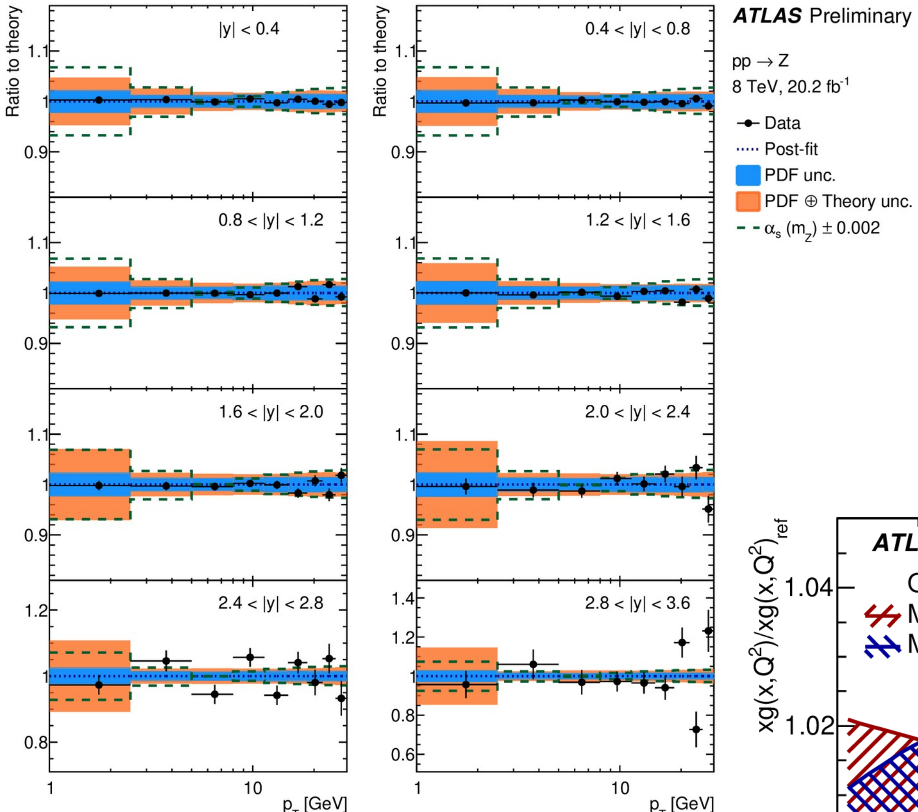


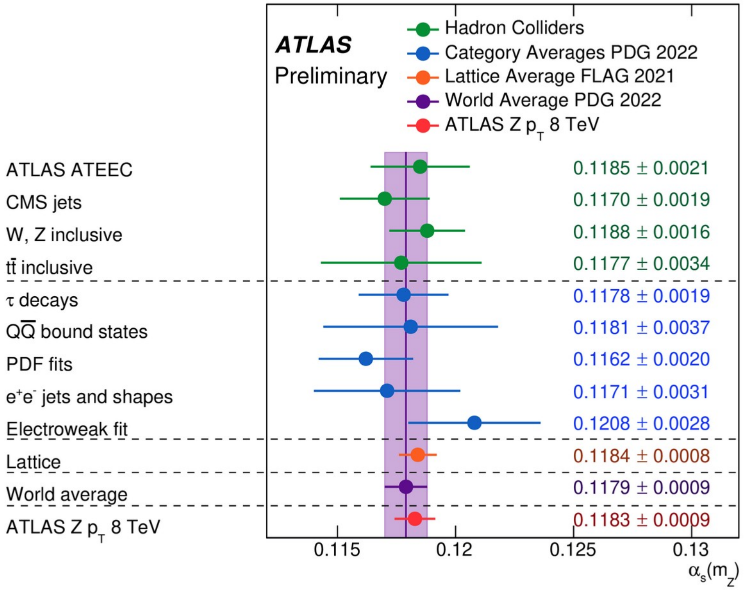


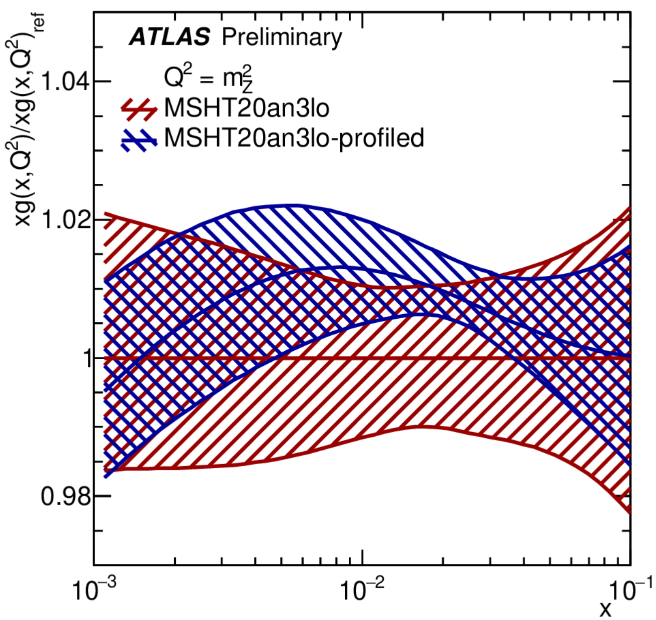
- $\alpha_s(m_Z)$ fit to CDF Z p_T measurement
- Available as example in xfitter

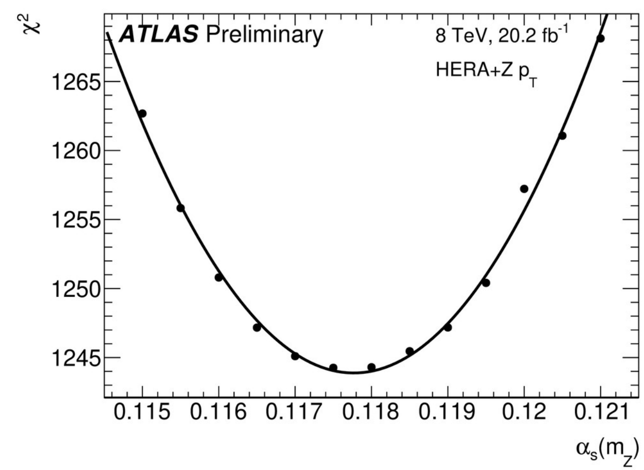
Applications: ATLAS Z p_T

- $\alpha_s(m_Z)$ fit to ATLAS Z p_T 8 TeV measurement
- Profiling and full simultaneous fit



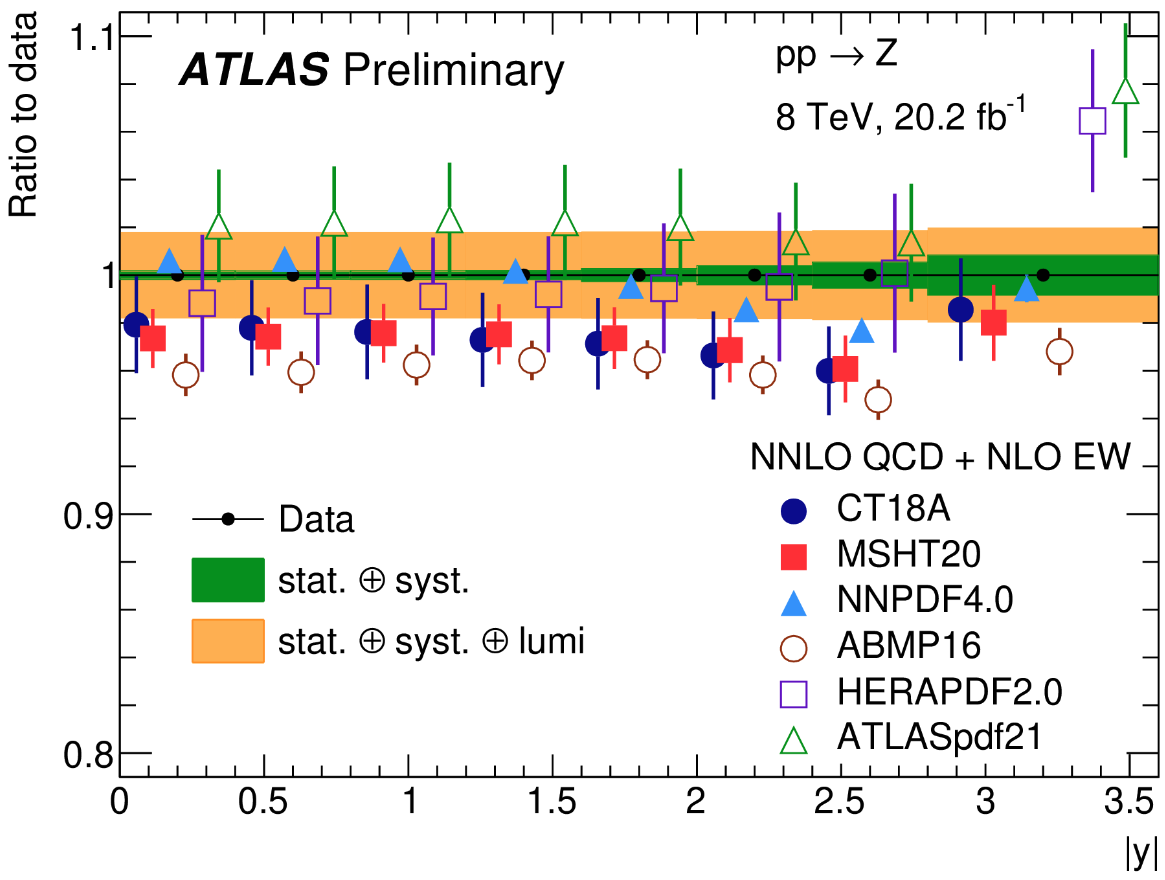






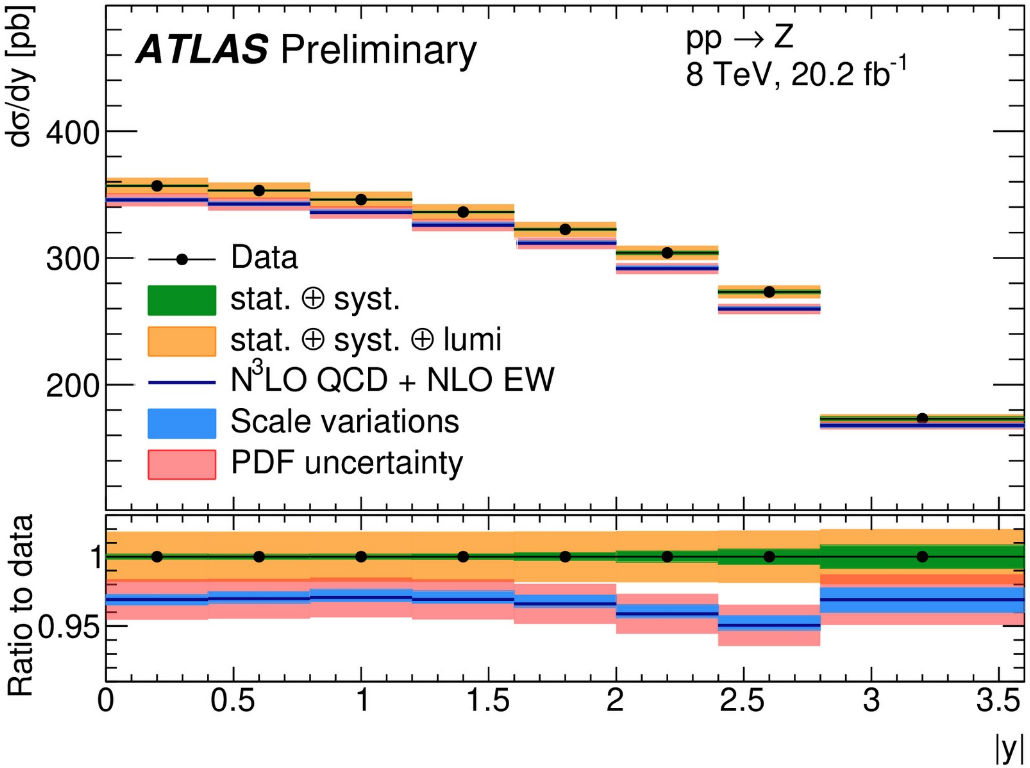
Applications: NNLO Z rapidity

- Z rapidity cross sections in full lepton phase space can be predicted fully with DYTurbo up to NNLO QCD, without grids, without kfactors
- With this setup, EW parameters can be easily included as free parameters in the fit



Applications: N3LO Z rapidity

- Z rapidity cross sections at N3LO with only a small additive correction for the α_s^3 real contribution




```
TheoryType     = 'expression'
TermName = 'A1','C'
TermSource = 'DYTurbo','KFactor'
TermInfo = 'FileName=data/zyfull8tev/dyturbo/zy-8tev.in',
           'FileName=data/zyfull8tev/mcfm/afactor/n3lo-afactor-zy-r1-f1.txt:FileColumn=3'
TheorExpr= '0.996*(2*A1+C)/1000'
```

DYTurbo tools: Anacondas

- ANAlytic CONtinuation of Deeply nested hArmonic Sums → ANACONDAS
  - Software for the analytic continuation of harmonic sums of any weight and depth
  - Based on S. Albino: arXiv:0902.2148
  - Implementation in FORM using some new additional fast recurrence relations

```
id S(R(-1,-1,-1),N?) =  - 2036868563/1280*N^-20*ln2 + 1277801843/3192*N^-19*
      ln2 + 955231/128*N^-18*ln2 - 237097967/16320*N^-17*ln2 - 181561/512*
      N^-16*ln2 + 16397/24*N^-15*ln2 + 1461/64*N^-14*ln2 - 945979/21840*N^-13*
      ln2 - 269/128*N^-12*ln2 + 1033/264*N^-11*ln2 + 19/64*N^-10*ln2 - 263/480
      *N^-9*ln2 - 9/128*N^-8*ln2 + 23/168*N^-7*ln2 + 1/32*N^-6*ln2 - 19/240*
      N^-5*ln2 - 1/32*N^-4*ln2 + 5/24*N^-3*ln2 - 3/8*N^-2*ln2 + 1/2*N^-1*ln2
       - 1/6*ln2^3 - 1/4*z3 - 1/2*z2*ln2 - 75066714618425/1634304*m1(N)*N^-20
       + 221930581/16*m1(N)*N^-20*ln2^2 + 221930581/16*m1(N)*N^-20*z2 +
      25753183106257/65345280*m1(N)*N^-19 + 693398149697/522240*m1(N)*N^-18 -
      3202291/8*m1(N)*N^-18*ln2^2 - 3202291/8*m1(N)*N^-18*z2 - 218817794609/
      15375360*m1(N)*N^-17 - 296712133/6144*m1(N)*N^-16 + 929569/64*m1(N)*
      N^-16*ln2^2 + 929569/64*m1(N)*N^-16*z2 + 425276477/640640*m1(N)*N^-15 +
      795995927/349440*m1(N)*N^-14 - 5461/8*m1(N)*N^-14*ln2^2 - 5461/8*m1(N)*
      N^-14*z2 - 12258493/295680*m1(N)*N^-13 - 4898731/33792*m1(N)*N^-12 + 691/
      16*m1(N)*N^-12*ln2^2 + 691/16*m1(N)*N^-12*z2 + 48763/13440*m1(N)*N^-11
       + 100969/7680*m1(N)*N^-10 - 31/8*m1(N)*N^-10*ln2^2 - 31/8*m1(N)*N^-10*
      z2 - 4213/8960*m1(N)*N^-9 - 9923/5376*m1(N)*N^-8 + 17/32*m1(N)*N^-8*
      ln2^2 + 17/32*m1(N)*N^-8*z2 + 91/960*m1(N)*N^-7 + 901/1920*m1(N)*N^-6 -
      1/8*m1(N)*N^-6*ln2^2 - 1/8*m1(N)*N^-6*z2 - 5/192*m1(N)*N^-5 - 37/96*m1(N
      )*N^-4 + 1/16*m1(N)*N^-4*ln2^2 + 1/16*m1(N)*N^-4*z2 + 7/16*m1(N)*N^-3 -
      1/4*m1(N)*N^-2 - 1/8*m1(N)*N^-2*ln2^2 - 1/8*m1(N)*N^-2*z2 + 1/4*m1(N)*
      N^-1*ln2^2 + 1/4*m1(N)*N^-1*z2;
```

- Could easily run all HS representation up to weight 8, with some moderate computing power should work up to w=10 or 11 (which covers N5LO applications)
- Provide also C++ code for HS up to weight 7 (sufficient for all currently known coefficients and anomalous dimensions in QCD)
- Code ready, but not yet available, let me know if interested in using it

DYTurbo tools: Anacondas

- Used in DYTurbo to compute fast numerical precise analytic continuations of all C3 coefficients and $\gamma 4$ splitting functions
- Functions optimised with Horner scheme

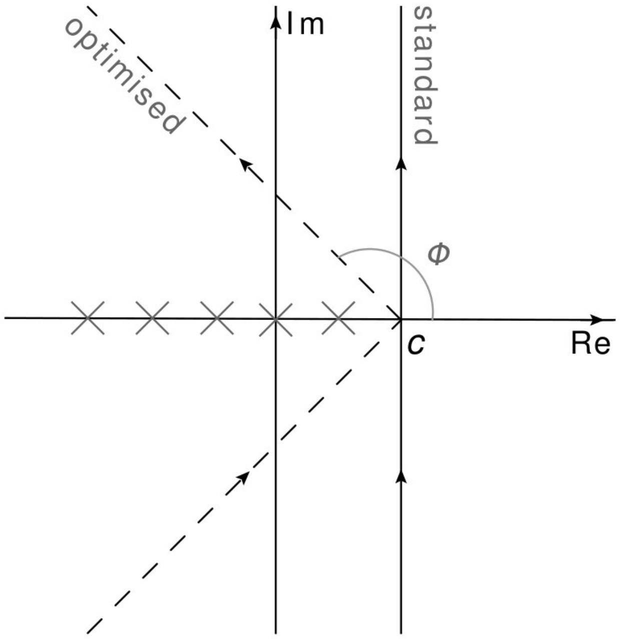
```
complex <double> C3qqN(complex <double> N)
{
  complex <double> Z1_,Z2_,Z3_,Z4_,Z5_,Z6_,Z7_,Z8_,Z9_,Z10_,Z11_,Z12_,Z13_,Z14_

   Z1_=pow(N,-1);
   Z2_=pow(N+1.,-1);
   Z3_=pow(N+2.,-1);
   Z4_=pow(N+3.,-1);
   Z5_=pow(N+4.,-1);
   Z6_=pow(N+5.,-1);
   Z7_=pow(N+6.,-1);
   Z8_=pow(N+7.,-1);
   Z9_=pow(N+8.,-1);
   Z10_=pow(N+9.,-1);
   Z11_=pow(N+10.,-1);
   Z12_=log(N+10.);
  Z13_=3.966666666666666E+1*Z4_;
  Z14_=Z13_ + 1.817859683120132E+2 + 1.2625E+1*Z3_;
  Z14_=Z14_*Z3_;
  Z15_=4.9E+1*Z4_;
  Z16_=Z15_ - 2.812802699573598E+2;
  Z16_=Z16_*Z4_;
  Z17_=2.6E+1*Z1_;
  Z18_=Z17_ + 8.205536466950359E+2;
  Z14_=Z14_ - Z16_ - 3.333333333333333E-1*Z18_;
  Z16_=1.19E+2*Z4_;
```

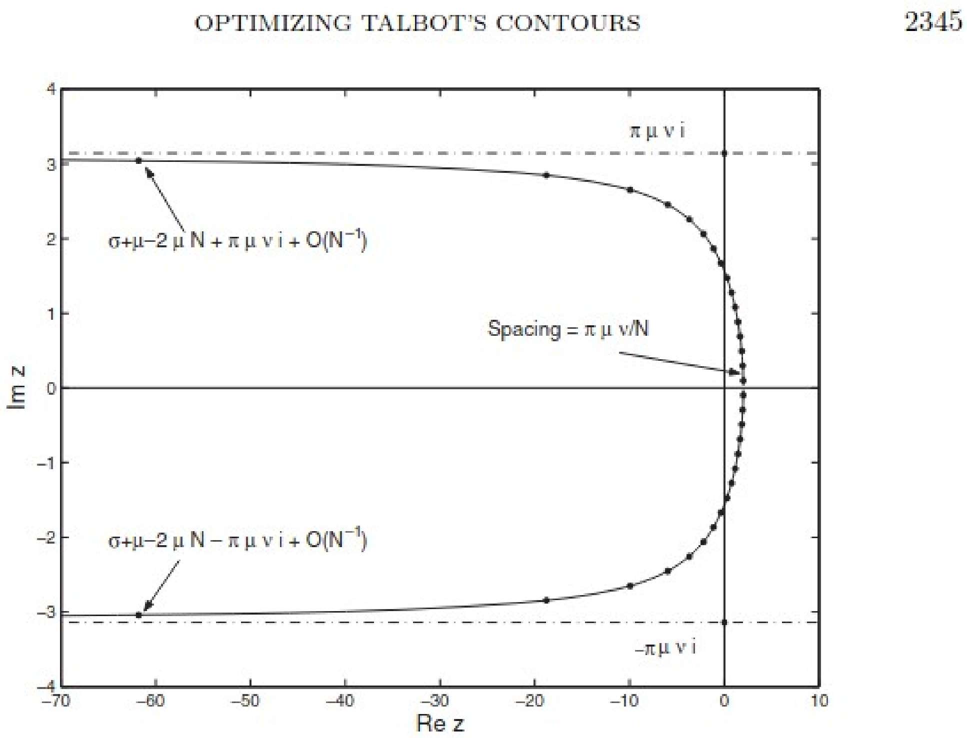
```
complex <double> gamma4nsp_A(complex <double> N)
{
  complex <double> Z1_,Z2_,Z3_,Z4_,Z5_,Z6_,Z7_,Z8_,Z9_,Z10_,Z11_,Z12_,Z13_,Z14_
,Z15_,Z16_,Z17_,Z18_,Z19_,Z20_,Z21_,Z22_,Z23_,Z24_,Z25_,Z26_,Z27_,Z28_,Z29_,Z30_
,Z31_,Z32_,Z33_,Z34_,Z35_,Z36_,Z37_;

   Z1_=pow(N+1.,-1);
   Z2_=pow(N+2.,-1);
   Z3_=pow(N+3.,-1);
   Z4_=pow(N+4.,-1);
   Z5_=pow(N+5.,-1);
   Z6_=pow(N+6.,-1);
   Z7_=pow(N+7.,-1);
   Z8_=pow(N+8.,-1);
   Z9_=pow(N+9.,-1);
   Z10_=pow(N+10.,-1);
   Z11_=log(N+10.);
  Z12_=4.45332212891E+11 + 6.341066100440467E+11*Z10_;
  Z12_=Z12_*Z10_;
  Z12_=Z12_ - 7.306225466856936E+10;
  Z13_=8.354218880534670E-4*Z10_;
  Z12_=Z12_*Z13_;
  Z12_=Z12_ - 4.764618249264705E+7;
  Z12_=Z12_*Z10_;
  Z12_=Z12_ + 8.835125278919895E+6;
  Z12_=Z12_*Z10_;
```

- Long compilation time, but extremely fast run time implementation

DYTurbo tools: Mellin transform

- Various implementation of optimised contours to transform back and forth from x to N space
  - Straight line, Talbot, parabolic, Weideman, Valko', Rizzardi





- Highly optimized inversion by bending on the negative real half plane when the integrand is known analytically
- Allows to deal with arbitrary PDF parametrisation using only numerical integration and positive real half plane contours
- In all cases, contour scaling as 1/log(z) ensures high and homogeneous accuracy

DYTurbo tools: PDF evolution

- Full reimplementation of Pegasus QCD evolution algorithm in C++, for the FFN case
- Extended to N3LO using known high-accuracy approximations of PNSp/m/v, PSGqq, and MSHTan3lo approximation for PSGqg/gq/gg
- Implemented both the truncated and iterative solutions
- Include small-x resummation from HELLx

```
//Compute the Rk matrices with Eq. (2.21) of https://arxiv.org/pdf/hep-ph/0408244.pdf
cmatrix R0 = g1sg/beta0;
cmatrix R1 = g2sg/beta0 - b1*R0;
cmatrix R2 = g3sg/beta0 - b1*R1 - b2*R0;
cmatrix R3 = g4sg/beta0 - b1*R2 - b2*R1 - b3*R0;

//Compute eigenvalues of the R0 matrix (See Eq. (2.27) of https://arxiv.org/pdf/hep-ph/0408244.pdf)
complex<double> htr = R0.trace()/2.;
complex<double> d = R0.det();
rp[i] = htr + sqrt(htr*htr - d);
rm[i] = htr - sqrt(htr*htr - d);

//Compute eigenvector projections (See Eq. (2.28) of https://arxiv.org/pdf/hep-ph/0408244.pdf)
Ep[i] = 1./(rp[i]-rm[i])*(R0-rm[i]*I);
Em[i] = 1./(rm[i]-rp[i])*(R0-rp[i]*I);

//Compute Rktilde and Uk matrices (Eq. (2.25) and (2.31) of https://arxiv.org/pdf/hep-ph/0408244.pdf)
cmatrix RTL1 = R1;
U1[i] = -Em[i]*RTL1*Em[i]/1. - Ep[i]*RTL1*Ep[i]/1. + Ep[i]*RTL1*Em[i]/(rm[i]-rp[i]-1.) + Em[i]*RTL1*Ep[i]/(rp[i]-rm[i]-1.); //Eq. (2.31) of https://arxiv.org/pdf/hep-ph/0408244.pdf
cmatrix RTL2 = R2+R1*U1[i];
U2[i] = -Em[i]*RTL2*Em[i]/2. - Ep[i]*RTL2*Ep[i]/2. + Ep[i]*RTL2*Em[i]/(rm[i]-rp[i]-2.) + Em[i]*RTL2*Ep[i]/(rp[i]-rm[i]-2.); //Eq. (2.31) of https://arxiv.org/pdf/hep-ph/0408244.pdf
cmatrix RTL3 = R3+R2*U1[i]+R1*U2[i];
U3[i] = -Em[i]*RTL3*Em[i]/3. - Ep[i]*RTL3*Ep[i]/3. + Ep[i]*RTL3*Em[i]/(rm[i]-rp[i]-3.) + Em[i]*RTL3*Ep[i]/(rp[i]-rm[i]-3.); //Eq. (2.31) of https://arxiv.org/pdf/hep-ph/0408244.pdf
```

DYTurbo tools: PDF evolution

- Code written with simple or recursive relations, could be easily extended to higher orders of PDF evolution once splitting functions are known
- Speed performance very stable order by order, no significant slow down from LO up to N3LO evolution
- VFN not currently available, but possible to implement it
- With some work (VFN, x-space intepolation, ...), it could be made an evolution engine directly available in xfitter
- Need to test performance of x → N and N → x transform

```
//recursive relation
if (opts.iterative)
  {
    cmatrix R[maxord];
    for (int k = 0; k < maxord; k++)
      R[k] = N;

    if (opts.order_evol >= 1)
      R[0] = R0;
    if (opts.order_evol >= 2)
      R[1] = R1;
    if (opts.order_evol >= 3)
      R[2] = R2;
    if (opts.order_evol >= 4)
      R[3] = R3;

    if (opts.order_evol == 2)
      for (int k = 2; k < maxord; k++)
        R[k] += -b1*R[k-1];
    else if (opts.order_evol == 3)
      for (int k = 3; k < maxord; k++)
        R[k] += -b1*R[k-1] -b2*R[k-2];
    if (opts.order_evol >= 4)
      for (int k = 4; k < maxord; k++)
        R[k] += -b1*R[k-1] -b2*R[k-2] -b3*R[k-3];

    //Recursive relation to define RTL and U at all orders
    cmatrix RTL[maxord];
    for (int k = 1; k < maxord; k++)
      {
        RTL[k] = R[k];
        for (int j = 1; j <= k-1; j++)
          RTL[k] += R[k-j]*U[idx(j,i)];
        U[idx(k,i)] = -Em[i]*RTL[k]*Em[i]/double(k) - Ep[i]*RTL[k]*Ep[i]/double(k) + Ep[i]*RTL[k]*Em[i]/(rm[i]-rp[i]-double(k)) + Em[i]*RTL[k]*Ep[i]/(rp[i]-rm[i]-double(k));
      }
  }
```

Projects: PDF+TMD fit

- First interesting application of the DYTurbo interface could be a simultaneous fit of PDFs and TMDs parameters
- Several TMD functional forms are available in DYTurbo, others can be easily implemented
- Possible also to parametrise TMDs in xfitter and pass them to DYTurbo

```
//Pavia19 fit
if (opts.npff == 4)
  {
    //From Table 5 of https://arxiv.org/pdf/1912.07550.pdf
    double lambda = opts.lambda; //0.580;
    double g2 = opts.g2A; //0.036;   //ln(m)-dependent quadratic (Gaussian) term
    double g2B = opts.g2B; //0.012;  //ln(m)-dependent quartic term
    double N1 = opts.NA; //0.625;
    double sigma = opts.sigmaA; //0.370;
    double alpha = opts.alphaA; //0.205;
    double N1B = opts.NB; //0.044;
    double sigmaB = opts.sigmaB; //0.356;
    double alphaB = opts.alphaB; //0.069;
    double Q0 = 1;
```

```
//SV19 fit
if (opts.npff == 5)
  {
    //https://arxiv.org/pdf/1912.06532.pdf Table 9
    double lambda1 = 0.224;
    double lambda2 = 9.24;
    double lambda3 = 375.;
    double lambda4 = 2.15;
    double lambda5 = -4.97;
```

Projects: PDF+TMD fit

- CDF Z p_T measurement available as example in xfitter

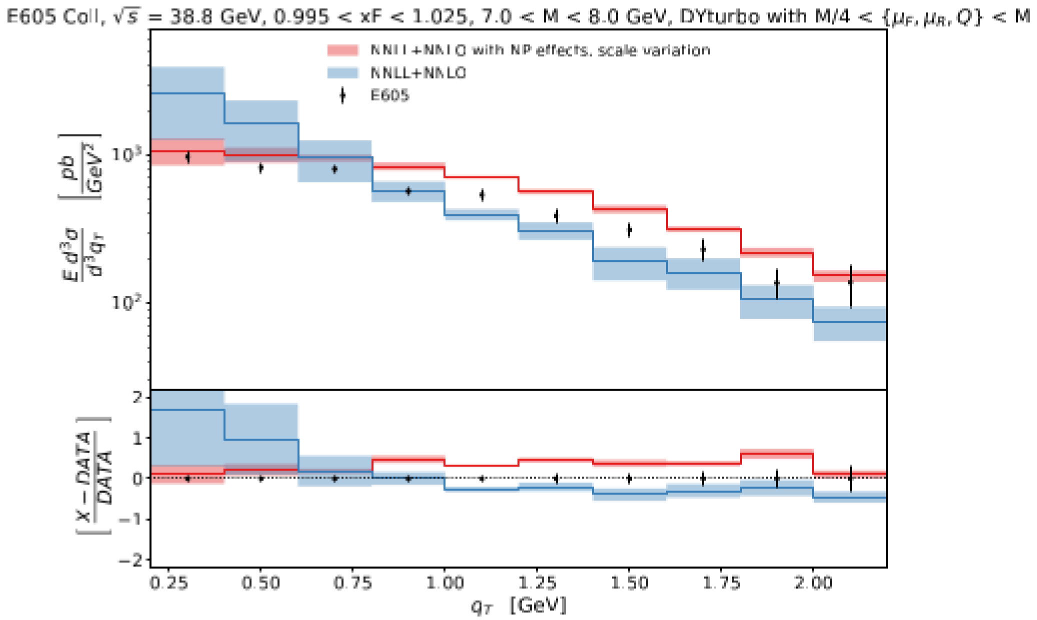
```
5125 $ ./tools/test.sh ZPT
5126 Testing ZPT ... PASS [details in temp/ZPT/test.log]
5127 -> 1 test(s) PASS
```

- The test performs the first few iterations of a very simple simultaneous PDF + TMD fit

NO	NAME	VALUE	STEP	LIM L	LIM H	SHIFT	REDUCT
1	Adbar	0.1613	0.0100				
4	Agp	0.1661	0.0100				
7	Bdbar	-0.1273	0.0040				
8	Bdv	1.0300	0.0600				
9	Bg	-0.0620	0.2700				
10	Bgp	-0.3831	0.0100				
12	Buv	0.8105	0.0160				
13	Cdbar	9.5862	1.2345				
14	Cdv	4.8463	0.3000				
15	Cg	5.5624	0.3200				
17	Cubar	7.0597	0.8000				
18	Cuv	4.8235	0.0600				
19	DYTurbo/g1	0.4889	0.0435				
20	Dubar	1.5481	1.0000				
22	Euv	9.9214	0.8000				

Projects: PDF+TMD fit

- Several low mass DY fixed target data available:
  - R209
  - PHENIX
  - E288
  - NUSEA
  - E605
- Sensitive to TMDs
- Already used in several TMD analyses
- Data could be loaded into xFitter to perform a PDF+TMD analysis



CERES improvements

- Included in master:
  - Synchronised CERES cost function and xFitter FCN so that they return the same value, now CERES and MINUIT optimise FCN to the same minimum (within required EDM)
  - Implemented approximate estimate of parameters errors and covariance from CERES using the Jacobian
- Further improvements:
  - After CERES fit compute errors and covariance estimate with MINUIT Hesse or other methods
  - Expand number of steerable CERES parameters, as needed

```
CERES:
  offset: 2
  tolerance: 1e-5
  strategy: 0
  covariance: 1
```

- PDF+TMD fits are likely to be very computationally heavy, CERES minimization is performing better than MINUIT

Projects: DIS+DY analysis

- Mandy: “Drell-Yan data are the most accurate LHC data and also those for which the most sophisticated predictions exist”
- Several tensions in DY data observed in PDF fits

Drell-Yan data in CT18(Z) global analyses

ID	Expt.	N_{pt}	χ^2	χ^2/N_{pt}	S_E
			CT14HERA2 data		
201	E605DY	119	103.4(102.4)	0.9(0.9)	-1.0(-1.1)
203	E866 $\sigma_{pd}/(2\sigma_{pp})$	15	16.1(17.9)	1.1(1.2)	0.3(0.6)
204	E866 $Q^3 d^2\sigma_{pp}/(dQ dx_F)$	184	244(240)	1.3(1.3)	2.9(2.7)
225	CDF1Z $A(e)$	11	9.0(9.3)	0.8(0.8)	-0.3(-0.2)
227	CDF2W $A(e)$	11	13.5(13.4)	1.2(1.2)	0.6(0.6)
234	D∅2W $A(\mu)$	9	9.1(9.0)	1.0(1.0)	0.2(0.1)
260	D∅2Z $y_{\ell\ell}$	28	16.9(18.7)	0.6(0.7)	-1.7(-1.3)
261	CDF2Z $y_{\ell\ell}$	29	48.7(61.1)	1.7(2.1)	2.2(3.3)
266	CMS7W $A(\mu)$	11	7.9(12.2)	0.7(1.1)	-0.6(0.4)
267	CSM7W $A(e)$	11	4.6(5.5)	0.4(0.5)	-1.6(-1.3)
268	ATL7WZ$_{(2012)}$	41	44.4(50.6)	1.1(1.2)	0.4(1.1)
281	D∅2W $A(e)$	13	22.8(20.5)	1.8(1.6)	1.7(1.4)
			New LHC data		
245	LHCb7WZ(μ)	33	53.8(39.9)	1.6(1.2)	2.2(0.9)
246	LHCb8Z(e)	17	17.7(18.0)	1.0(1.1)	0.2(0.3)
248	ATL7WZ$_{(2016)}$	34	287.3(88.7)	8.4(2.6)	13.7(4.8)
249	CMS8W $A(\mu)$	11	11.4(12.1)	1.0(1.1)	0.2(0.4)
250	LHCb8WZ(μ)	34	73.7(59.4)	2.1(1.7)	3.7(2.6)
253	ATL8ZpT	27	30.2(28.3)	1.1(1.0)	0.5(0.3)

Projects: DIS+DY analysis

- We have expertise to understand and improve the correlation model and the theory predictions for these datasets

Data set	HERA I χ^2 / number of points	HERA I + Tevatron W, Z χ^2 / number of points
NC DIS cross sections H1-ZEUS combined e^-p.	112 / 145	106 / 145
NC DIS cross sections H1-ZEUS combined e^+p.	326 / 337	334 / 337
CC DIS cross sections H1-ZEUS combined e^-p.	20 / 34	19 / 34
CC DIS cross sections H1-ZEUS combined e^+p.	27 / 34	33 / 34
HERA I correlated χ^2	21	23
D0 $d\sigma(Z)/dy$	-	23 / 28
CDF $d\sigma(Z)/dy$	-	33 / 28
D0 muon charge asymmetry in $W \to \mu\nu$	-	12 / 10
CDF W charge asymmetry in $W \to e\nu$	-	15 / 13
D0 W charge asymmetry in $W \to e\nu$	-	16 / 14
Total $\chi^2_{\min}$ / dof	505 / 535	615 / 628

- xFitter PDF fit to DIS and DY data with
  - N3LO QCD predictions for DIS and DY, NLO EW corrections
  - aN3LO PDF evolution
  - qT-resummation corrections where needed
  - χ^2/dof ~ 1, and $\Delta\chi^2$ = 1 for the uncertainties

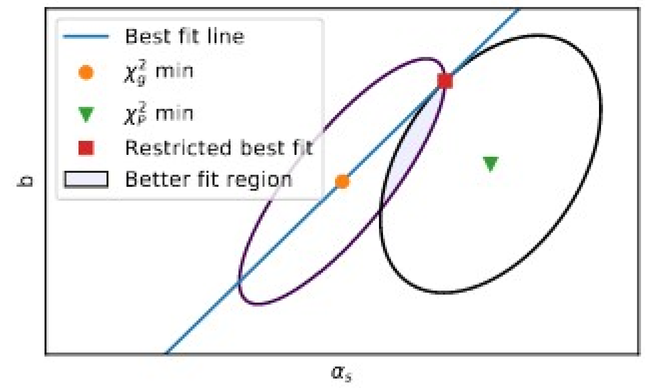
CT14nnlo 68%CL

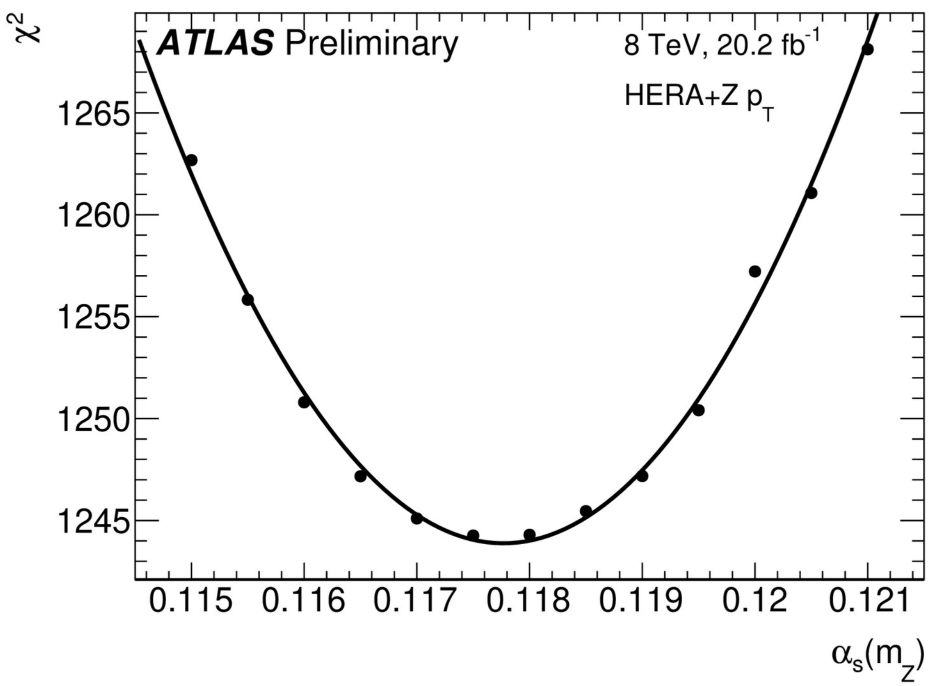
Dataset	NNLO q_T-subtr.	NNLO recoil q_T-subtr.	NNLO+ NNLL
W^+ lepton rapidity	9.4/11	8.8/11	8.8/11
W^- lepton rapidity	8.2/11	8.7/11	8.2/11
Low mass, Z rapidity	11/6	7.2/6	7.5/6
Mass peak, central Z rapidity	15/12	10/12	7.7/12
Mass peak, forward Z rapidity	9.6/9	5.3/9	6.4/9
High mass, central Z rapidity	6.0/6	6.5/6	5.8/6
High mass, forward Z rapidity	5.2/6	5.6/6	5.3/6
Correlated χ^2	40	40	31
Log penalty χ^2	-4.33	-3.39	-4.20
Total χ^2/dof	99/61	88/61	77/61
χ^2 p-value	0.00	0.01	0.08

Projects: α_s profiling or fit equivalence

- Recently argued that α_s should be fitted simultaneously with the PDFs
- But PDF profiling is an approximation of a full PDF fit
- With xFitter it should be possible to study the equivalence of α_s determinations with PDF profiling and full fit

arXiv:2001.04986





Summary

- DYTurbo interface is working, and was already used for a few interesting applications, as α_s from Z p_T
- Implemented the possibility to fit simultaneously TMD and PDF parameters
- Internal PDF evolution engine of DYTurbo could be exported to xFitter
- A comprehensive fit of DIS+DY data at N3LO would be a nice xFitter project