Implementation of the FONLL GM-VFN in HERAfitter

Juan Rojo

HERAfitter workshop

Marseille, 13/02/2012

The FONLL GM-VFN scheme

S. Forte, E. Laenen, P. Nason and J. Rojo, Nucl.Phys. B834 (2010) 116-162

- The FONLL GM-VFN scheme for DIS structure functions matches consistently the **FFN result** (suitable at low Q^2) to the **ZM-VFN result** (suitable at large Q^2)

GM-VFN →

$$F^{\rm FONLL}(x,Q^2) = F^{(d)}(x,Q^2) + F^{(n_l)}(x,Q^2),$$
$$F^{(d)}(x,Q^2) \equiv \left[F^{(n_l+1)}(x,Q^2) - F^{(n_l,0)}(x,Q^2)\right]$$

ZM-VFN →

$$F^{(n_l+1)}(x,Q^2) = x\int_x^1 \frac{dy}{y} \sum_{i=q,\bar{q},h,\bar{h},g} C_i^{(n_l+1)}\left(\frac{x}{y}, \alpha_s^{(n_l+1)}(Q^2)\right) f_i^{(n_l+1)}(y,Q^2)$$

FFN →

$$F^{(n_l)}(x,Q^2) = x\int_x^1 \frac{dy}{y} \sum_{i=q,\bar{q},g} C_i^{(n_l)}\left(\frac{x}{y}, \frac{Q^2}{m^2}, \alpha_s^{(n_l)}(Q^2)\right) f_i^{(n_l)}(y,Q^2).$$

- It is **valid to all perturbative orders**, and can **combine different pert. orders** in the FFN and VFN terms
- FONLL variants: **FONLL-A** (NLO VFN and LO FFN), **FONLL-B** (NLO VFN and NLO FFN), and **FONLL-C** (NNLO VFN + NLO FFN)
- **Unique set of PDFs** for all scales, matched at HQ thresholds

Treatment of HQ thresholds

- In any GM-VFN scheme, there is a freedom to modify the **threshold behavior** by suitably subleading terms
- Such variations are an **inherent theoretical ambiguity** of the GM formulation
- In FONLL two possibilities: a **threshold damping factor** (default) or a **chi-scaling**

Threshold damping factor

$$F^{(d,th)}(x,Q^2) = f_{\rm thr}(x,Q^2) F^{(d)}(x,Q^2),$$

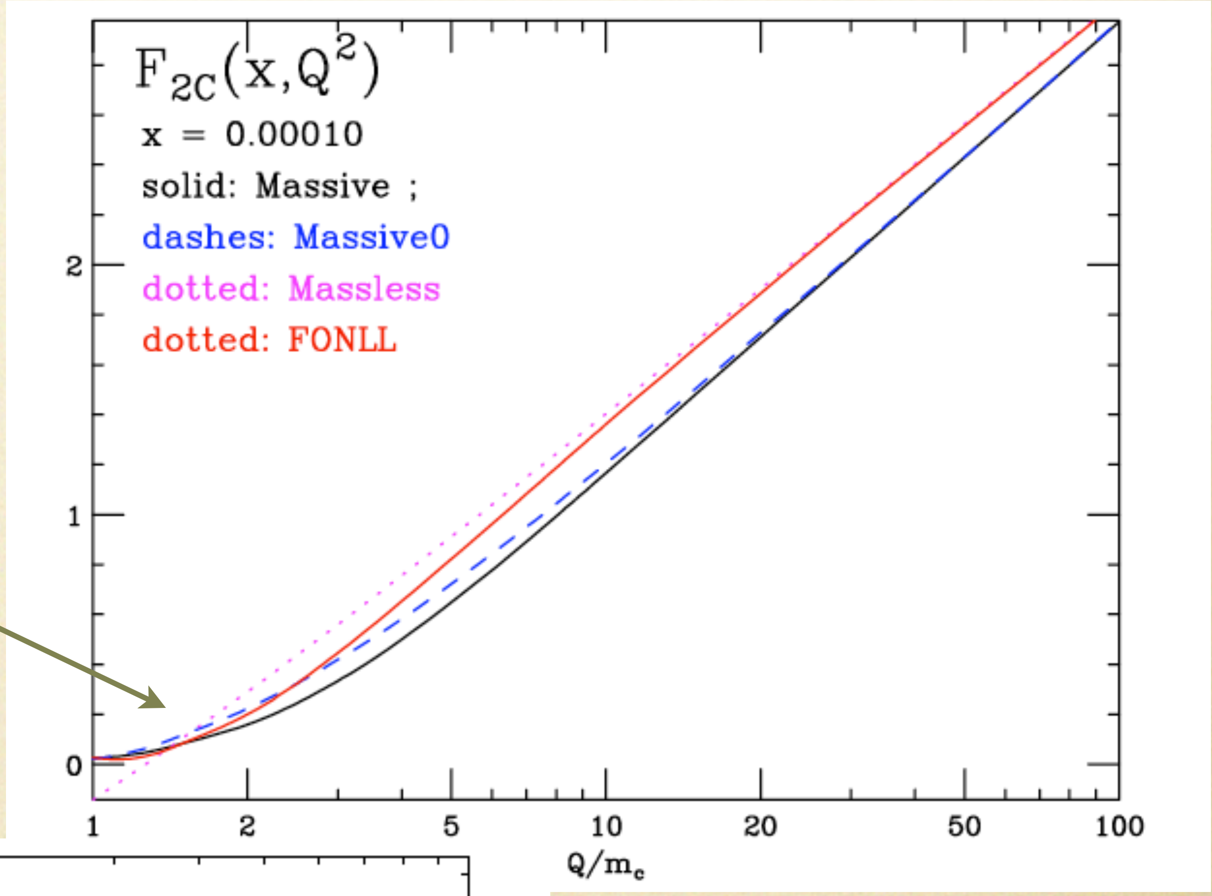
$$f_{\rm thr}(x,Q^2) = \Theta(Q^2 - m^2)\left(1 - \frac{Q^2}{m^2}\right)^2$$

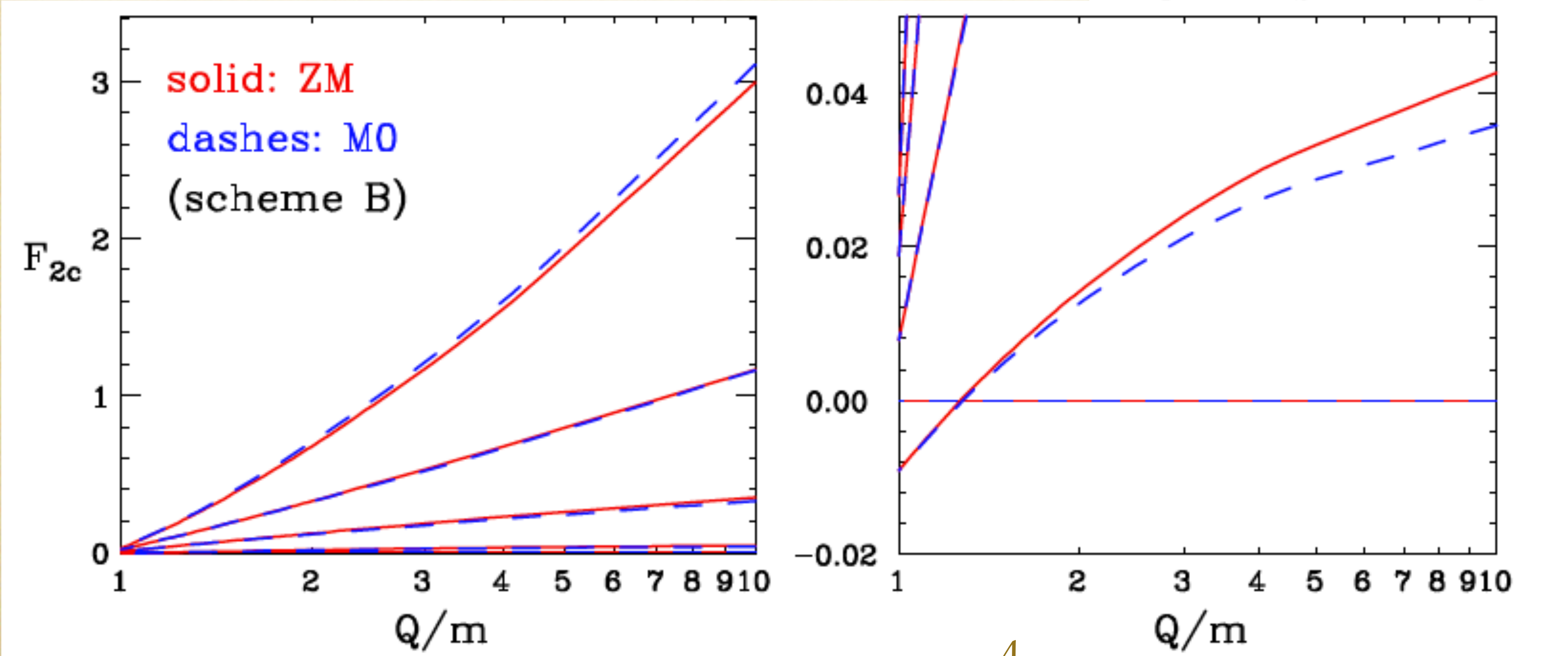
Chi-scaling

$$F^{(\chi)}(x,Q^2) = x\int_{\chi(x,Q^2)}^{1} \frac{dy}{y} C\left(\frac{\chi(x,Q^2)}{y}, \alpha(Q^2)\right) f(y,Q^2),$$

FONLL: Validation

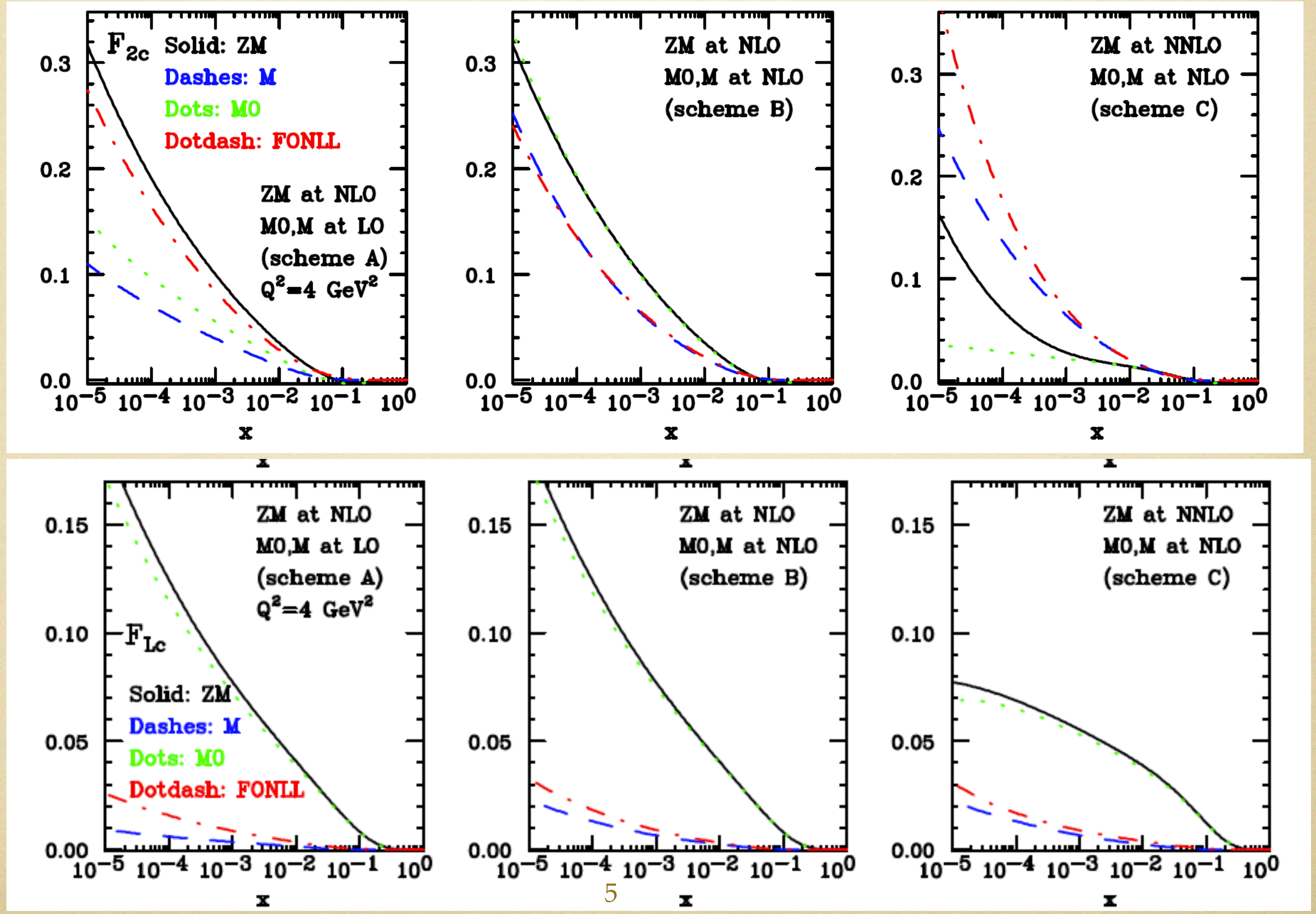
Match with FFN at low scales, with ZM-VFN at high scales





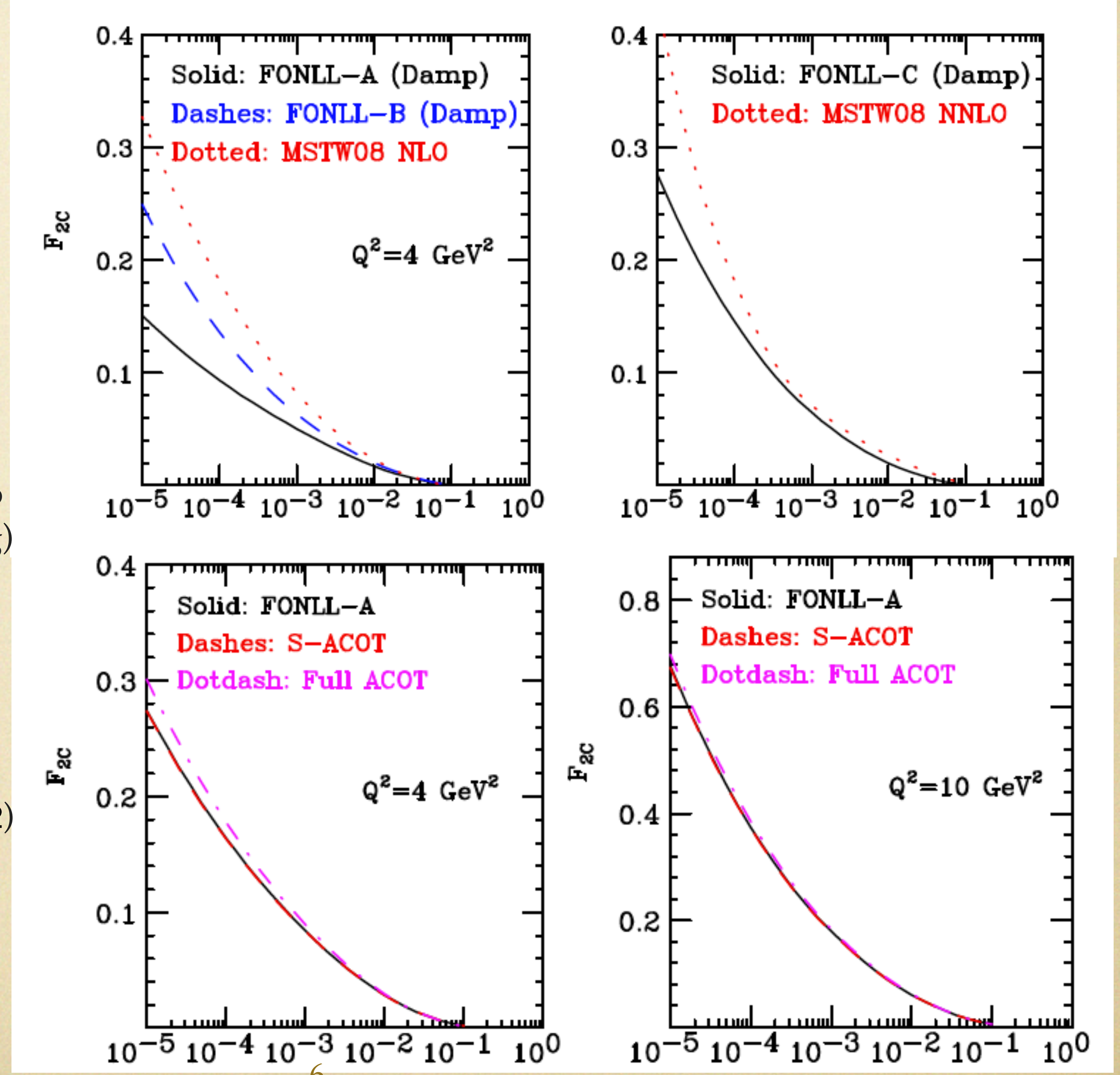
ZM and M0 schemes match smoothly at Q=m in FONLL-B: no need of threshold suppression

FONLL: numerical results



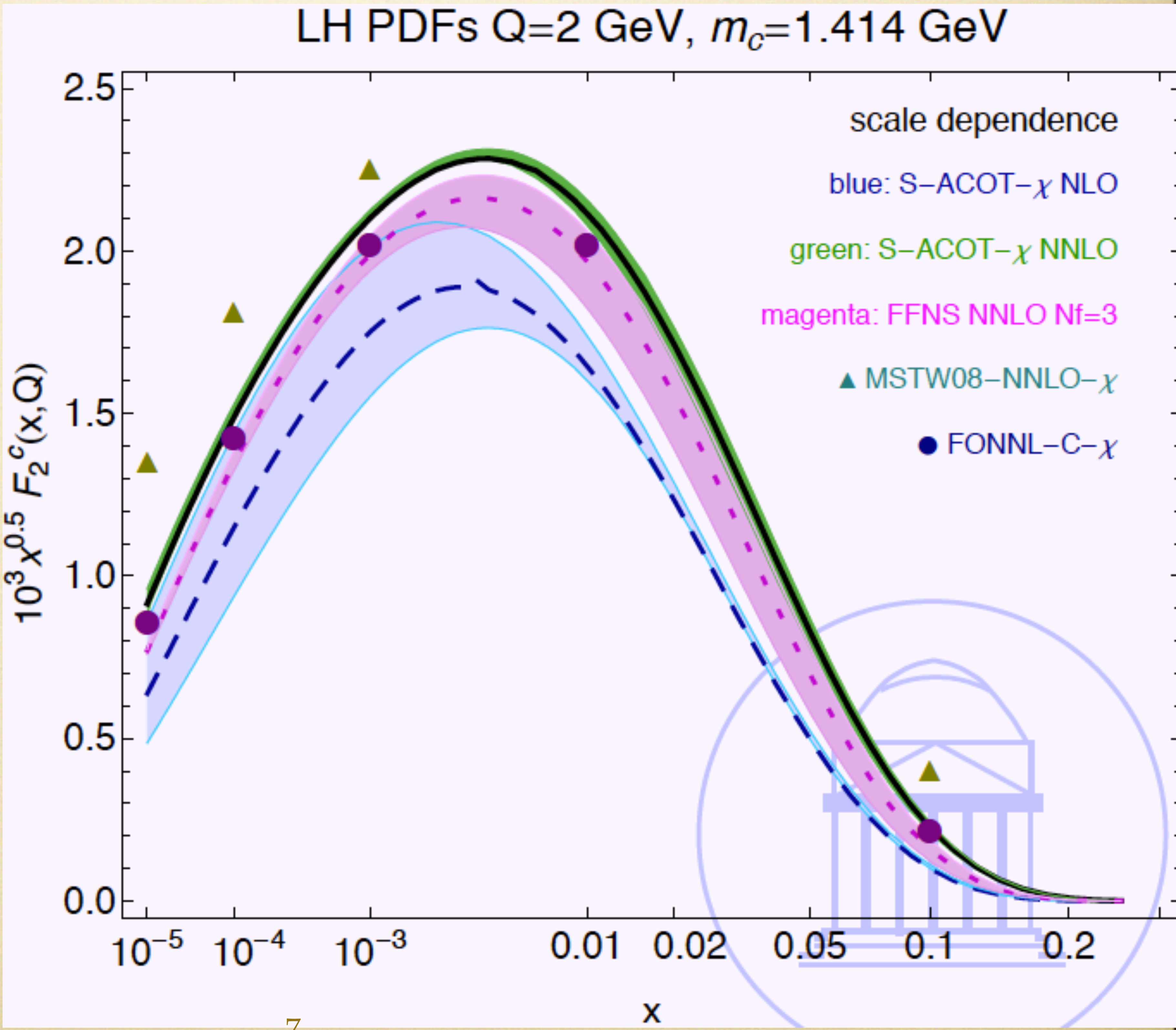
Comparison with other GM-VFN schemes

- Systematic comparison in the **Les Houches 2009 proceedings** (arXiv:1003.1241)
- **FONLL-A** identical to **S-ACOT** if same threshold prescription chosen
- **MSTW08 NLO/NNLO** identical to **FONLL A/C** up to a Q^2 independent (subleading) term
- More recently: **FONLL-C** numerically similar to **S-ACOT NNLO**, to be used in CT NNLO fits (arXiv:1108.5112)
- **Relation between different GM-VFN schemes** well understood



Comparison with other GM-VFN schemes

- Systematic comparison in the **Les Houches 2009 proceedings** (arXiv:1003.1241)
- **FONLL-A** identical to **S-ACOT** if same threshold prescription chosen
- **MSTW08 NLO/NNLO** identical to **FONLL A/C** up to a Q^2 independent (subleading) term
- More recently: **FONLL-C** numerically similar to **S-ACOT NNLO**, to be used in CT NNLO fits (arXiv:1108.5112)
- **Relation between different GM-VFN** schemes well understood



A technical aside

- Original FONLL code uses **exact x-space NNLO massive coefficient functions (**Nucl.Phys. B392 (1993) 162**)**
- Too slow to be used in a QCD analysis: FONLL implemented in the **NNPDF FastKernel** Mellin space framework
- FONLL-A used in **NNPDF2.1 NLO**, FONLL-C in **NNPDF2.1 NNLO**
- NNLO FFN coefficient functions from the **Alekhin-Blumlein parametrization (**Phys.Lett. B594 (2004) 299**)**
- **FastKernel FONLL** is accurate enough, and orders of magnitude faster ...
- .... But requires to **precompute a priori tables** (like APPLgrid) for given **scenarios: set of x,Q^2 values, m_c, $\alpha_S(M_Z)$**, ...

	F_{2c} FONLL-C			F_{Lc} FONLL-C		
x	FONLLdis	FastKernel	Accuracy (%)	FONLLdis	FastKernel	Accuracy (%)
$Q^2 = 4$ GeV^2						
10^{-5}	0.27830	0.28163	1.18	0.02468	0.02500	1.30
10^{-4}	0.14709	0.14858	1.00	0.01423	0.01441	1.23
10^{-3}	0.06556	0.06591	0.52	0.00733	0.00735	0.24
10^{-2}	0.02034	0.02034	0.00	0.00281	0.00283	0.74
$Q^2 = 10$ GeV^2						

Implementation in HERAfitter

- A first beta version of **stand-alone FastKernel FONLL** already available (V. Radescu)

- In the next weeks: **optimization of the interface, implementation in HERAfitter** and **validation**

Source code

```
pb-d-128-141-234-114:fonll-herafitter juanrojochacon$ ls
Makefile                          pdflha2evln.f
f2c-check.data                    perf.data
fonll-herafitter.f                perf.gp
fonll-herafitter.inc              sigma-N2LO_FXD_UNP-TRN-DIS.res-tot
initpdf0.f                        sigmaread.f
ixp-DIS.res-tot                   xgrid.f
kin.data                          xsec.f
obs_DIS_FP_evolx.f
```

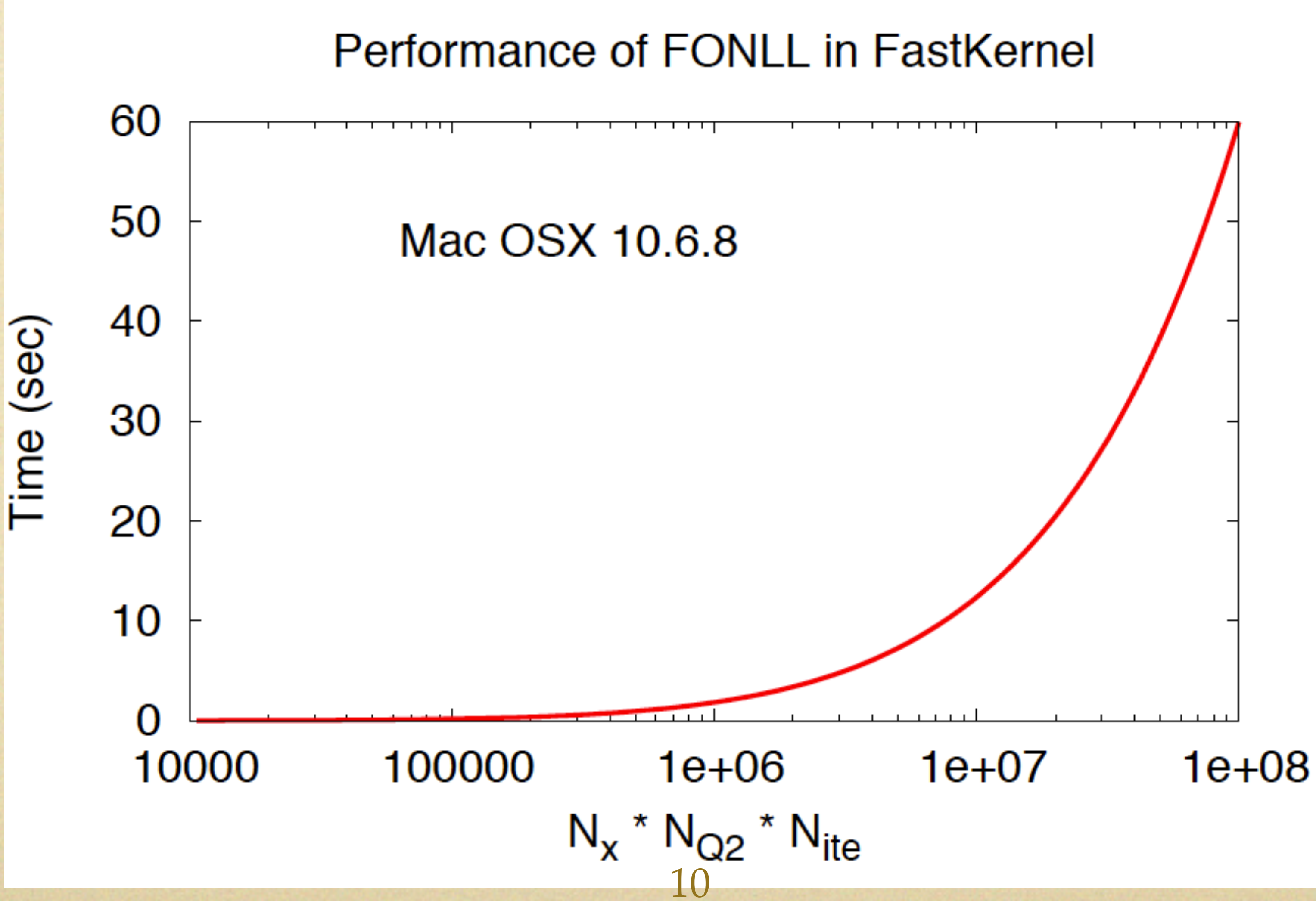
```
write(6,*) " -------------------------
write(6,*) " "
write(6,*) " FONLL structure functions
write(6,*) " computed in the FastKernel framework"
write(6,*) " "
write(6,*) " ---------------------------------------- "

Initialization
call fonll_herafitter_init

Set PDF set
PDFSET="NNPDF21_nnlo_100.LHgrid"
IREP = 0 ! Check on central replica

Initialize PDFs
call InitPDFsetbyname(PDFSET)
call InitPDF(IREP)
CALL initpdf0

Get F2c and save results for cross-check
open(unit=50,status="unknown",file="f2c-check.res")
call wrap_FONLL(F2C)
close(50)
```

Stand-alone FONLL code

- Still need to **define scenarios** for which to precompute FONLL grids

FONLL: Numerical Performance

- The current implementation of FONLL in FastKernel is **extremely fast**
- The speed is the same for all **GM schemes**: A and B for NLO PDFs, C for NNLO PDFs, and any choice of threshold approximations: no need to rely on **approximate K-factors**
- Evaluate $\mathbf{F_{2,C}}$ at a given x and Q^2 takes only **60 ns**



Summary and outlook

- The FONLL GM-VFN scheme provides a **consistent and flexible framework** to include heavy quark mass effects in a QCD analysis
- FONLL-A and FONLL-C used in NNPDF2.1 NLO and NNLO
- The FONLL implementation in FastKernel is extremely fast and accurate. Main caveat is the need to **precompute a priori tables** for given **scenarios: set of x,Q^2 values, m_c, $\alpha_S(M_Z)$**, ...
- **FONLL A,B,C will be implemented and tested** in HERAfitter in the next weeks, then ready for phenomenology!