Short Report for nPDFs

Patrick Gotzler

University Regensburg, Regensburg
Forschungsgruppe FOR2926

xFitter Developers Meeting, 4 June 2025

Goal and Hurdles

- **Goal:** Incorporate nuclear PDFs into the new xFitter

- **What needs to be modified?**
  1. Datasets
  2. Parameterization
  3. Decomposition
  4. Evolution (QCDNUM)
  5. Storing Output
  6. Draw Tool

Datasets

- TermInfo easily handles more elements

```
&Data
        Name = 'NPB293'
        IndexDataset = 309
        Reaction = 'NC e+-p'

        TermName = 'R'
        TermSource = 'use:hf_scheme_DISNC'
        TermInfo = 'type=F2:flav=incl:echarge=-1.0:epolarity=0.0:A1nucl=2.0:Z1=1.0'
        TheorExpr = 'R'

        NDATA = 66
        NColumn = 8
        ColumnType = 'Bin', 'Bin', 'Bin', 'Sigma', 4*'Error'
        ColumnName = 'y', 'x', 'Q2', 'F2', 'stat', 'ignore', 'uncor', 'ignore'

        Percent = 4*False
```

- Have 'Get functions' similar to how echarge or epolarity are handled.

Parameterisation

- Use nCTEQ or TUJU(NuclearDaiquiri) strategy to have proton parameterisation as boundary value
- Parameterise nPDFs as

$$f_i(A, x, Q^2) = c_0(A)\, x^{c_1(A)}\, (1-x)^{c_2(A)}\, p_i(x, A)$$

  where $p_i(x, A)$ is polynomial in $\sqrt{x}$ with A-dependent coefficients and its exact form depends on the parton i.
- The coefficients have the form

$$c_k(A) = c_{k,0}\, A^{-c_{k,1}}$$

  s.th for $A = 1$, we obtain the proton parameters $c_{k,0}$.
- Implementation based on HERAPDF Parameterisation with only small changes.

Parameterisation

- Setting, e.g. A-Parameter by searching it in the global gParameters map

```
double SetNuclearA1()
{
std::string part = "A1nucl";
    // Declare variable to contain the value of A1 read from gParameters
    double AParameter;

//     Iterate through the map and check if each key contains the part of the string
    for (const auto& pair : XFITTER_PARS::gParameters) {
     // Check if the part of the string exists within the key
     if (pair.first.find(part) != std::string::npos) {
           auto it = XFITTER_PARS::gParameters.find(pair.first);
           if (it != XFITTER_PARS::gParameters.end())
               {
                 AParameter = *it->second;
               }
               }
               }
        return AParameter;
}
```

- Works great, if you only have a single nucleus.
- Works bad for multiple nuclei. Then only last read A value is retained.

Decomposition

- Using UvDvUbarDbarS Decomposition
- Decomposition of structure functions into nuclear PDFs uses isospin symmetry for light quarks.
- nPDF is weighted average of bound proton/neutron PDFs

$$f_i^A = \frac{Z}{A} f_i^{p/A} + \frac{A-Z}{A} f_i^{n/A}$$

- Setting (A, Z) values in the same way as in the parameterisation. Thus same problem.
- Could also implement isospin symmetry at F_2 calculation, but doing so at decomposition is less prone to errors when modifying. Results are the same.

Evolution

- Using QCDNUM and RTDISNC
- Two ideas:
  1. Define multiple evolutions, parameterisations and decompositions and specify for each dataset the evolution by EvolutionCopy
  2. QCDNUM allows up to 24 PDF sets to be handled. Since number of nuclei for which data exists is limited, treat every pdf set in QCDNUM as the pdf for a given nuclei.
- Naive implementation of idea 1 throws an error in GXMAKE(...) as x-Grid is already defined.
- Implementation of idea 2 requires changes such as
  1. EvolutionQCDNUM::atIteration: Modify to the following QCDNUM::evolfg(10 * iset + _itype, funcPDF,..), loop over iset to evolve every PDF set.
  2. EvolutionQCDNUM::xfxQarray(...): Contains QCDNUM::allfxq(iset,...); use iset to select PDF set. Relevant when storing output since this is when all flavor pdfs are gotten.

This part does not work yet. First will have to solve problems in parameterisation and decomposition to decide.

Storing Output

- Added loop over integers associated to A-values.
- Extended created 'pdfs_q2val_XX.txt' to have for every A-value all Q^2-values in their own files, i.e. 'pdfs_q2val_01A1.txt', 'pdfs_q2val_01A2.txt', ...
- Files contain header with basic information (Q^2, Number of PDFs, x-range) and the A-value

Draw Tool

- Smaller modifications to get A-value in the same way as Q^2-value is gotten.
- Modifications to draw a pdf plot for every 'pdfs_q2val_xxAy.txt' output file with reference to A-value in the labelling of PDF plot.