Physics Research Showcase

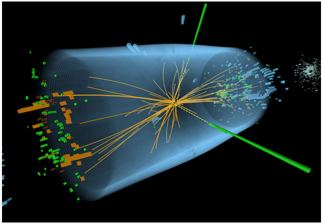
Numerical solutions to the DGLAP equations

Casey Hampson

18 April, 2025

Introduction

- High-energy physicists are tasked with determining the behavior of the universe on the smallest possible scales.
- One of the main places that we do this is at CERN in Geneva, Switzerland, which is where the Large Hadron Collider (LHC) is housed.
- The LHC accelerates protons to almost the speed of light and collides them together inside of a detector, which analyzes the thousands of particles that fly out.



Photos

Photos

- Parton Distribution Functions (PDFs) are experimentally determined distributions that give probabilities of finding partons (i.e. quarks and gluons) with momentum fraction x within the proton at an energy scale Q.

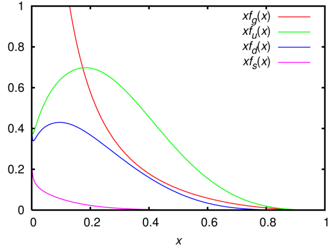


Figure: Parton distributions for the first few light quark flavors and the gluon from the CTEQ group, evaluated at 2 GeV.

DGLAP Evolution Equations

- Unfortunately, PDFs are not able to be calculated from first principles.
- We can, however, determine how they evolve with energy via the DGLAP (Dokshitzer-Gribov-Lipatov-Altarelli-Parisi) equations.
- These are a set of integro-differential equations that go like:

$$\frac{\partial}{\partial \ln Q^2} = P(x, \alpha_s(Q^2)) \otimes f(x, Q^2), \tag{1}$$

where f is the PDF, α_s is the strong coupling constant, and P is the *splitting function*, which roughly corresponds to probabilities for quarks to emit gluons and vice versa. The $\otimes$ corresponds to a convolution, defined like so:

$$[a \otimes b](x) = \int_x^1 \frac{\mathrm{d}y}{y} a\left(\frac{x}{y}\right) b(y) = \int_x^1 \frac{\mathrm{d}y}{y} a(y) b\left(\frac{x}{y}\right). \tag{2}$$

Solutions to the DGLAP Equations

- Solutions are particularly challenging due to the convolution appearing on the R.H.S.
- One main solution is moving to Mellin space:

$$a(N) = \int_0^1 \mathrm{d}x \, x^{N-1} a(x) \tag{3}$$

which turns a convolution into a normal product:

$$[a \otimes b](N) = a(N) \cdot b(N). \tag{4}$$

- This, however, incurs the cost of transforming to and from x-space and Mellin space. It also loses the main mathematical and physical structure of the equations, which is something we want to keep.

The Non-singlet Case

- There are two classes of solutions called “singlet” and “non-singlet.” The singlet case involves matrices, and the non-singlet involves scalars. The non-singlet case is therefore much more simple.
- For this case at Leading Order (LO), one particular ansatz that evolves the PDF f from an initial scale Q_0^2 to a final scale Q^2 is given by:

$$f(x, Q^2) = \sum_{n=0}^{\infty} \frac{A_n(x)}{n!} \ln^n \left(\frac{a_s(Q^2)}{a_s(Q_0^2)} \right), \tag{5}$$

- The $A_n(x)$s satisfy a recursion relation:

$$A_{n+1}(x) = -\frac{2}{\beta_0}[P^{(0)} \otimes A_n](x). \tag{6}$$

The Non-singlet Case

- At *NLO* the ansatz looks roughly the same, but of course with higher order terms

$$f(x, Q^2) = \sum_{n=0}^{\infty} \sum_{s=0}^{n} \frac{B_n^s(x)}{n!(s-n)!} \ln^n\left(\frac{\alpha_s}{\alpha_0}\right) \ln^{s-n}\left(\frac{4\pi\beta_0 + \alpha_s\beta_1}{4\pi\beta_0 + \alpha_0\beta_1}\right). \tag{7}$$

- The $B_s^n(x)$s will follow more advaned recursion relations as well, i.e. all the $n \neq 0$ coefficients first.
- Generalizing this to N^3LO is one of our main goals.

Singlet Ansatz

- For the singlet case, which are matrices, we consider the solution in Mellin space, which is:

$$\mathbf{f}(N, Q^2) = \left[1 + \sum_{k=0}^{\kappa} \alpha_s^k \mathbf{U}_k\right] \mathbf{L} \left[1 + \sum_{k=0}^{\kappa} \alpha_0^k \mathbf{U}_k\right]^{-1} \mathbf{f}(N, Q_0^2). \tag{8}$$

- Here, the $\mathbf{U}_k$ matrices contain information related to the splitting function which we would keep to only N^3LO, but the index κ is called a *truncation index*, which in principle we would take to infinity, but practically must keep finite.
- In x-space, the regular products of these matrices would turn into convolutions. This is currently being tested.

Current Progress

- At the moment, we don't have any concrete results for including the approximate N^3LO splitting function corrections or the new singlet ansatz (it's been a busy semester!).
- The previous CANDIA code was written in C, but has since been ported to C++ with a few minor optimizations. Once the new singlet ansatz and aforementioned splitting function corrections are added we will release the code and call it CANDIA-V2.
- On the next slide are example outputs/plots for the current version of the program.

Example Outputs