Implementation of the Truncated Solution in APFEL

Richard D. Ball and Valerio Bertone

June 7, 2016

Abstract

In this document we describe how to implement the truncated (or linearized) solution of the DGLAP equation in APFEL.

1 The Algorithm

The DGLAP evolution equations, that govern the evolution with respect to the factorization scale μ, of the vector of parton distribution functions $\mathbf{f}$ has the form:

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

where $\mathbf{P}$ is the matrix of splitting functions and where we have introduced the convolution operator $\otimes$ defined as:

$$f(x) \otimes g(x) = \int_0^1 dy \int_0^1 dz f(y) g(z) \delta(x - yz)\,. \tag{2}$$

Now, considering the RG equation for the strong coupling α_s:

$$\mu_R^2 \frac{\partial}{\partial \mu_R^2} \alpha_s(\mu_R) = \beta(\alpha_s(\mu_R))\,, \tag{3}$$

which governes the evolution of α_s as a function of the renormalization scale μ_R and identifying μ_R with μ, we can combine eqs. (1) and (3) obtaining:

$$\frac{\partial}{\partial \alpha_s} \mathbf{f}(x,\mu) = \mathbf{R}(x,\alpha_s(\mu)) \otimes \mathbf{f}(x,\mu)\,. \tag{4}$$

where we have defined:

$$\mathbf{R}(x,\alpha_s(\mu)) = \frac{\mathbf{P}(x,\alpha_s(\mu))}{\beta(\alpha_s(\mu))}\,. \tag{5}$$

Of course, solving eq. (4) is exactly equivalent to solving eq. (1) and they are referred to as **exact** solutions[1].

At N^{n}LO, the matrix of splitting functions $\mathbf{P}$ and the function β have the following perturbative expansions:

$$\mathbf{P}(x,\alpha_s(\mu)) = \alpha_s(\mu) \sum_{k=0}^{n} \alpha_s^k(\mu) \mathbf{P}^{(k)}(x) \tag{6}$$

and:

$$\beta(\alpha_s(\mu)) = -\alpha_s^2(\mu) \sum_{k=0}^{n} \alpha_s^k(\mu) \beta_k\,. \tag{7}$$

and thus also the matrix $\mathbf{R}$ defined in eq. (5) can be expanded as:

$$\mathbf{R}(x,\alpha_s(\mu)) = -\frac{1}{\alpha_s(\mu)\beta_0} \sum_{k=0}^{n} \alpha_s^k(\mu) \mathbf{R}^{(k)}(x) \tag{8}$$

[1] APFEL implements the solution to both eqs. (4) and (1) and the result are actually identical.

where:

$$\begin{aligned}\mathbf{R}^{(0)}(x) &= \mathbf{P}^{(0)}(x)\\ \mathbf{R}^{(1)}(x) &= \mathbf{P}^{(1)}(x) - b_1\mathbf{P}^{(0)}(x)\\ \mathbf{R}^{(2)}(x) &= \mathbf{P}^{(2)}(x) - b_1\mathbf{P}^{(1)}(x) + (b_1^2 - b_2)\mathbf{P}^{(0)}(x)\end{aligned} \tag{9}$$

where we have defined:

$$b_k = \frac{\beta_k}{\beta_0}\,. \tag{10}$$

We will call the solution of eq. (4) using the expansion of the matrix $\mathbf{R}$ in eq. (11) **expanded** solution.

In order to compute the **truncated** or **linearized** solution, we introduce in the expansion in eq. (11) the small positive parameter ε in such a way that:

$$\mathbf{R}(x,\alpha_s(\mu)) \to \mathbf{R}(x,\alpha_s(\mu),\varepsilon) = -\frac{1}{\alpha_s(\mu)\beta_0}\left[\mathbf{R}^{(0)}(x) + \varepsilon\alpha_s(\mu)\mathbf{R}^{(1)}(x) + \varepsilon^2\alpha_s^2(\mu)\mathbf{R}^{(2)}(x) + \dots\right]. \tag{11}$$

This will clearly introduce a dependence on ε in the evolved PDFs so that the solution of the DGLAP equation will be:

$$\mathbf{f} \equiv \mathbf{f}(x,\mu,\varepsilon)\,. \tag{12}$$

Now, in order to construct the linearized solution at the different orders we need to Taylor expand the solution around $\varepsilon = 0$ and truncated the expansion at the desired order. Provided that ε is small enough, this can easily be done numerically just by taking the following combinations:

$$\begin{aligned}\mathbf{f}^{\rm LO}(x,\mu) &= \mathbf{f}(x,\mu,0)\\ \mathbf{f}^{\rm NLO}(x,\mu) &= \mathbf{f}(x,\mu,0) + \frac{\mathbf{f}(x,\mu,\varepsilon) - \mathbf{f}(x,\mu,-\varepsilon)}{2\varepsilon} + \mathcal{O}(\varepsilon^2\mathbf{f}^{(3)})\\ \mathbf{f}^{\rm NNLO}(x,\mu) &= \mathbf{f}(x,\mu,0) + \frac{\mathbf{f}(x,\mu,\varepsilon) - \mathbf{f}(x,\mu,-\varepsilon)}{2\varepsilon} + \frac{\mathbf{f}(x,\mu,\varepsilon) - 2\mathbf{f}(x,\mu,0) + \mathbf{f}(x,\mu,-\varepsilon)}{2\varepsilon^2} + \mathcal{O}(\varepsilon^2\mathbf{f}^{(3)})\end{aligned} \tag{13}$$

where $\mathbf{f}^{(n)}$ is the n-th-order derivative of $\mathbf{f}$ with respect to ε evaluated in $\varepsilon = 0$. It should be noticed that $\mathbf{f}^{(n)} \propto \alpha_s^{n+1}$, for $n \geq 0$, and thus the real error in our case is $\mathcal{O}(\varepsilon^2\alpha_s^4)$. Finally, it should be pointed out that the order of the perturbative evolution of α_s in each of the solutions above must be consistent with the order to which the solution has been truncated.

Based on numerical tests, we observe that the procedure is quite stable over a wide range of values ε. In particular, the numerical results do not change significantly in the range $\varepsilon \in [10^{-3} : 10^{-9}]$. As a consequence, we have chosen to set our default value of ε to 10^{-5}.

One final remark regarding the implementation of the algorithm described above when evolving PDFs in the VFNS is necessary. As is well known, the evolution in the VFNS is performed by matching the evolutions with different numbers of active flavors at the heavy quark thresholds using a suitable set of perturbative matching conditions. In order to ensure that the matching is consistently done for all contributions, the prescriptions given in eq. (13) are applicable only if no heavy quark threshold is crossed during the evolution. To explain what it practically means, suppose to evolve PDFs between the scales μ_0 and μ and that one heavy quark threshold μ_h is crossed during the evolution. In this case, starting from the initial scale PDFs $\mathbf{f}(x,\mu_0)$, the first step it to compute the contributions $\mathbf{f}(x,\mu_h,0)$, $\mathbf{f}(x,\mu_h,\varepsilon)$ and $\mathbf{f}(x,\mu_h,-\varepsilon)$. These contributions must then be combined using one of the prescriptions given in eq. (13), according to the perturbative order. Finally the result of the combination $\mathbf{f}^{\rm N^kLO}(x,\mu_h)$ can be used as an initial condition to compute the contributions $\mathbf{f}(x,\mu,0)$, $\mathbf{f}(x,\mu,\varepsilon)$ and $\mathbf{f}(x,\mu,-\varepsilon)$ that can finally be combined into the final scale PDFs $\mathbf{f}^{\rm N^kLO}(x,\mu)$, again using eq. (13). The procedure can be easily generalized for a larger number of threshold crossings.

In order to ensure a better accuracy, one might employ a more accurate algorithm for the computation of the derivatives which relies on a larger number of points around the LO solution([2]). We can then use five points rather than three. This clearly ensures a better accuracy, as the error associated to this algorithm is $\mathcal{O}(\varepsilon^4)$, but the price to pay is that we need to compute two more evolution as compared to algorithm presented in eq. (13),

[2]Note that in eq. (13) we use three points, *i.e.* ε, 0 and $-\varepsilon$.

namely for $\pm 2\varepsilon$. The formulas to be implemented are:

$$\begin{aligned}
\mathbf{f}^{\rm LO}(x,\mu) &= \mathbf{f}(x,\mu,0) \\
\mathbf{f}^{\rm NLO}(x,\mu) &= \mathbf{f}(x,\mu,0) + \frac{-\mathbf{f}(x,\mu,2\varepsilon) + 8\mathbf{f}(x,\mu,\varepsilon) - 8\mathbf{f}(x,\mu,-\varepsilon) + \mathbf{f}(x,\mu,-2\varepsilon)}{12\varepsilon} + \mathcal{O}(\varepsilon^4\mathbf{f}^{(5)}) \\
\mathbf{f}^{\rm NNLO}(x,\mu) &= \mathbf{f}(x,\mu,0) + \frac{-\mathbf{f}(x,\mu,2\varepsilon) + 8\mathbf{f}(x,\mu,\varepsilon) - 8\mathbf{f}(x,\mu,-\varepsilon) + \mathbf{f}(x,\mu,-2\varepsilon)}{12\varepsilon} \\
&+ \frac{-\mathbf{f}(x,\mu,2\varepsilon) + 16\mathbf{f}(x,\mu,\varepsilon) - 30\mathbf{f}(x,\mu,0) + 16\mathbf{f}(x,\mu,-\varepsilon) - \mathbf{f}(x,\mu,-2\varepsilon)}{24\varepsilon^2} + \mathcal{O}(\varepsilon^4\mathbf{f}^{(5)})
\end{aligned} \quad . \tag{14}$$

Following the same argument given above, we have that the error associated to the NLO and NNLO solutions is $\mathcal{O}(\varepsilon^4\alpha_s^6)$.