begindocument/before

Notes on the algorithms used in artemide ver.3.

Alexey A. Vladimirov
August 17, 2023

I. CONVOLUTION AND GRIDS FOR DISTRIBUTIONS WITH TWIST-2 INPUT

A. X grid

The value of x spans $0 < x < 1$, but never reaches $x = 0$. The grid is characterized by

- $x_{\min}$ the minimal value of x
- N_x number of segments. The number of nodes is $N_x + 1$.
- Function x_i which is $x_0 = x_{\min}$ and $x_{N_x} = 1$.

These functions determine the following extra functions

- $\mathrm{inv}[x]$ which is inverse of x_i. The integer part of $\mathrm{inv}[x]$ gives the number i, such that $x_i < x < x_{i+1}$
- Δ_x typical increment

For any grid one has

$$\mathrm{inv}[x_i] = i. \tag{1.1}$$

There are many functional forms for grid. Here are the most convenient

- **Linear grid**

$$x_i = x_{\min} + i\Delta_x, \qquad \Delta_x = \frac{1 - x_{\min}}{N_x}, \qquad \mathrm{inv}[x] = \frac{x - x_{\min}}{\Delta_x}. \tag{1.2}$$

  Not convenient since PDFs usually grows at small-x

- **Logarithm grid**

$$x_i = x_{\min} 10^{i\Delta_x}, \qquad \Delta_x = \frac{-\log_{10} x_{\min}}{N_x}, \qquad \mathrm{inv}[x] = \frac{\log_{10}(x/x_{\min})}{\Delta_x}. \tag{1.3}$$

  This is almost standard in the PDF community. However, at $x \sim 1$ the points became sparse, and it rise the problem of precision loss. Usual solution is to use several grids for different ranges of x.

- **Arcosh grid**

$$x_i = \frac{1}{\cosh\left(\Delta_x(i - N_x)\right)}, \qquad \Delta_x = \frac{\mathrm{arccosh}(1/x_{\min})}{N_x}, \qquad \mathrm{inv}[x] = N_x - \frac{\mathrm{arccosh}(1/x)}{\Delta_x}. \tag{1.4}$$

  This grid is alike logarithmic at small-x, but has extra points at $x \to 1$. Therefore, one does not need extra griding at large-x.

- **Arcosh grid with control parameter**

$$x_i = \frac{1}{\cosh^p\left(\Delta_x(i - N_x)\right)}, \qquad \Delta_x = \frac{\mathrm{arccosh}(1/x_{\min}^{1/p})}{N_x}, \qquad \mathrm{inv}[x] = N_x - \frac{\mathrm{arccosh}(1/x^{1/p})}{\Delta_x}. \tag{1.5}$$

  This grid is alike logarithmic at small-x, but has extra points at $x \to 1$. Therefore, one does not need extra griding at large-x. The parameter p gives extra flexibility. At $p_x > 1$ the density of points at $x \to 1$ grows.

B. Lagrange interpolation

The Lagrange interpolation is defined by the order and the shift. Basically one defines

- $K = \{s, s+1, ..., s+r+1\}$, where s is a shift, and r is the order.

The standard choices are

- **Qubic backward grid:** $s = -2$, $r = 3$, $K = \{-2, -1, 0, 1\}$. In this case the function is determined in the last segment from three.
- **Qubic central grid:** $s = -1$, $r = 3$, $K = \{-1, 0, 1, 2\}$. In this case the function is determined in the middle segment from three.
- **Qubic forward grid:** $s = 0$, $r = 3$, $K = \{0, 1, 2, 3\}$. In this case the function is determined in the first segment from three.

The interpolation is done by

$$F(x) = \sum_{i=0}^{N_x-1} \theta(x_i < x < x_{i+1}) \sum_{k\in K} P_{i,k}(x) F(x_{i+k}), \tag{1.6}$$

where P is the Lagrange polynomial

$$P_{i,k}(x) = \prod_{\substack{l\in K \\ 0\le l+i\le N_x \\ l\neq r}} \frac{\mathrm{inv}[x] - \mathrm{inv}[x_{i+l}]}{\mathrm{inv}[x_{i+k}] - \mathrm{inv}[x_{i+l}]}. \tag{1.7}$$

Note, that since $\mathrm{inv}[x_i] = i$ it simplifies

$$P_{i,k}(x) = \prod_{\substack{l\in K \\ 0\le l+i\le N_x \\ l\neq r}} \left(\frac{\mathrm{inv}[x]}{k-l} - \frac{i+l}{k-l} \right). \tag{1.8}$$

The restriction $0 \le l + i \le N_x$ drops the contribution of the nodes that are outside of the grid. It reduces the order of interpolation polynomial for these segments. E.g. in the forward grid ($K = [0, 1, 2...]$) the largest segment is interpolated by a linear polynomial.

The interpolation sum can be rearranged as

$$F(x) = \sum_{i=0}^{N_x-1} W_i(x) F(x_i), \tag{1.9}$$

where

$$W_i(x) = \sum_{k\in K} \theta(x_{i-k} < x < x_{i-k+1}) P_{i-k,k}(x). \tag{1.10}$$

This function has the following properties

- $W_i(x) \neq 0$ for $x_{i-K_{\max}} < x < x_{i-K_{\min}+1}$ which is the same as $x_{i-s-r-1} < x < x_{i-s+1}$
- $W_i(x_j) = \delta_{ij}$
- The shape of the function W is independent on i. I.e. being evaluated in the terms of node-variables it is the same, but up to a shift.

One of the problems with griding PDFs is the unknown value of PDF out side of the grid. The possible solutions are 1) reduce the order of interpolation at the borders 2) approximate points outside the grid.

I have checked that with the arcosh-grid [$x_{\min} = 10^{-5}$, $N_x = 200$] the error for forward and backward grids at $x \to 1$ are very similar [test function behaved $(1-x)^{2+\cdot\cdot}$] if $f(x > 1) = 0$. Therefore, the forward grid (in this case) is a good choise.

C. B grid

The value of b spans $0 < b$, but for the large values of b the function does not change much and can be approximated by constant. The grid is characterized by

- $B_{\max}$ the maximum value of b stored in the grid.
- N_b number of segments. The number of nodes is $N_b + 1$.
- Function b_i which is $b_0 = 0$ and $b_{N_b} = B_{\max}$.

These functions determine the following extra functions

- $\mathrm{inv}[b]$ which is inverse of b_i. The integer part of $\mathrm{inv}[b]$ gives the number i, such that $b_i < b < b_{i+1}$
- Δ_b typical increment

For any grid one has

$$\mathrm{inv}[b_i] = i. \tag{1.11}$$

There are many functional forms for grid. Here are the most convenient

- **Linear grid**

$$b_i = i\Delta_b, \qquad \Delta_b = \frac{B_{\max}}{N_b}, \qquad \mathrm{inv}[b] = \frac{b}{\Delta_b}. \tag{1.12}$$

  Not convenient.

- **Exponential grid**

$$b_i = (B_{\max}+1)^{i\Delta_b} - 1, \qquad \Delta_b = \frac{1}{N_b}, \qquad \mathrm{inv}[b] = N_b \frac{\log(b+1)}{\ln(B_{\max}+1)}. \tag{1.13}$$

  This is a good grid because typical OPE is logarithmic in b.

- **Exponential grid with $b_{\min}$**

$$b_i = B_{\min} e^{-i\Delta_b}, \qquad \Delta_b = \frac{1}{N_b} \ln\left(\frac{b_{\min}}{B_{\max}}, \qquad \mathrm{inv}[b] = \frac{1}{\Delta_b} \ln\left(\frac{b_{\min}}{b}\right)\right). \tag{1.14}$$

  This grid has a lot of points at small-b, and for $b < b_{\min}$ one could freeze the value.

For the large values of b (which extend the grid), it is sufficient to extrapolate by the formula

$$f(b) = \left(\frac{b}{b_E}\right)^{\alpha_E}. \tag{1.15}$$

The values b_E and α_E can be found from the last two points of the grid $\{b_1, f_1\}$ and $\{b_2, f_2\}$

$$\alpha_E = \frac{\ln(f_2/f_1)}{\ln(b_2/b_1)}, \qquad b_E = b_1 (f_1)^{-1/\alpha_E} \tag{1.16}$$

D. Convolution on the grid

The convolution has the general form

$$[C \otimes f](x, b, \mu) = \sum_f \int_x^1 \frac{dy}{y} C_{f,f'}\left(\frac{x}{y}, b, \mu\right) f_{f'}(y, \mu). \tag{1.17}$$

Here the μ is internal parameter specified in the model. Furthermore, In the grid I save the function multiplied by x I get

$$\begin{aligned} x[C\otimes f](x,b,\mu) &= \sum_f \int_x^1 dy \frac{x}{y} C_{f,f'}\left(\frac{x}{y},b,\mu\right) f_{f'}(y,\mu) \\ &= \sum_f \int_x^1 dy \frac{x}{y} C_{f,f'}(y,b,\mu) f_{f'}\left(\frac{x}{y},\mu\right) \\ &= \sum_f \int_x^1 dy C_{f,f'}(y,b,)F_{f'}\left(\frac{x}{y},\mu\right), \end{aligned} \tag{1.18}$$

where $F(x) = xf(x)$ is the function stored in LHAPDF grids.

For it I use the algorithm from QCDnum. The coefficient function is presented as

$$C_{f,f'}(x,b,\mu) = \sum_{n=0}\sum_{k=0}^{n} a_s^n \mathbf{L}^k C_{f,f'}^{(n,k)}(x). \tag{1.19}$$

Since the function is the function on the grid one can compute

$$[C\otimes f](x,b,\mu) = \sum_{f'}\sum_{n=0}\sum_{k=0}^{n}\sum_{i=0}^{N_x-1} a_s^n \mathbf{L}^k F_{f'}(x_i,\mu)\mathfrak{T}_{ff'}^{(n,k)}(x,x_i), \tag{1.20}$$

where

$$\mathfrak{T}_{ff'}^{(n,k)}(x,x_i) = \int_x^1 dy C_{ff'}^{(n,k)}(y) W_i\left(\frac{x}{y}\right). \tag{1.21}$$

For the $x = x_j \in$grid one has

$$[C\otimes f](x_i,b,\mu) = \sum_{f'}\sum_{n=0}\sum_{k=0}^{n}\sum_{j=0}^{N_x-1} a_s^n \mathbf{L}^k \mathfrak{T}_{ff';ij}^{(n,k)} f_{f'}(x_j,\mu), \tag{1.22}$$

where

$$\mathfrak{T}_{ff',ij}^{(n,k)} = \int_{x_i}^1 dy\ C_{ff'}^{(n,k)}(y) W_j\left(\frac{x_i}{y}\right). \tag{1.23}$$

Note that since the function W has restrictired support one has

$$\mathfrak{T}_{ff',ij}^{(n,k)} = \int_{\max(x_i/x_{j-K_{\min+1}},x_i)}^{\min(1,x_i/x_{j-K_{\max}})} dy C_{ff'}^{(n,k)}(y) W_j\left(\frac{x_i}{y}\right). \tag{1.24}$$

More over since $x < 1$

$$\mathfrak{T}_{ff',ij}^{(n,k)} = \int_{x_i/x_{j-K_{\min+1}}}^{\min(1,x_i/x_{j-K_{\max}})} dy C_{ff'}^{(n,k)}(y) W_j\left(\frac{x_i}{y}\right). \tag{1.25}$$

The coefficient function has three parts

$$C(x) = C_\delta(x) + C_S(x) + C_R(x), \tag{1.26}$$

where

$$C_\delta(x) = c_\delta \delta(1-x), \qquad C_S(x) = c_S[g(x)]_+, \tag{1.27}$$

and C_R is a regular part. Clearly the contribution of C_δ to $\mathfrak{T}$ is

$$c_\delta \delta_{ij} \to \mathfrak{T}_{ff',ij}^{(n,k)} \tag{1.28}$$

The regular part contributes

$$\int_{x_i/x_{j-K_{\min+1}}}^{\min(1,x_i/x_{j-K_{\max}})} dyC_R(y)W_j\left(\frac{x_i}{y}\right) \to \mathfrak{T}^{(n,k)}_{ff',ij} \tag{1.29}$$

Finally, the plus-part contributes

$$\begin{aligned}
&\int_{x_i/x_{j-K_{\min+1}}}^{\min(1,x_i/x_{j-K_{\max}})} dyC_S(y)W_j\left(\frac{x_i}{y}\right) = \int_{x_i/x_{j-K_{\min+1}}}^{\min(1,x_i/x_{j-K_{\max}})} dyc_S[g(y)]_+W_j\left(\frac{x_i}{y}\right) \qquad (1.30)\\
&= \int_{x_i/x_{j-K_{\min+1}}}^{\min(1,x_i/x_{j-K_{\max}})} dyc_Sg(y)_+(W_j\left(\frac{x_i}{y}\right) - W_j(x_i)) - c_S\int_0^{j-K_{\min}+1} dyg(y)W_j(x_i)\\
&= \int_{x_i/x_{j-K_{\min+1}}}^{\min(1,x_i/x_{j-K_{\max}})} dyc_Sg(y)_+(W_j\left(\frac{x_i}{y}\right) - \delta_{ij}) - c_S\int_0^{j-K_{\min}+1} dyg(y)\delta_{ij} \to \mathfrak{T}^{(n,k)}_{ff',ij},
\end{aligned}$$

where it is used that $W_j(x_i) = \delta_{ij}$.

In the case of forward cubic grid:

$$\int_{x_i/x_{j-K_{\min+1}}}^{\min(1,x_i/x_{j-K_{\max}})} \quad \to \quad \int_{x_i/x_{j+1}}^{\min(1,x_i/x_{j-3})}$$

This method appears to have a fixed precision. I.e. its precision is equal proportional to the grid spacing. For reasonable grid the integral produces 10^{-4} relative precision, which cannot be improved. Precision drops for smaller x (because of the range of integration). This is due to the fact that this method effective incorporate the mistake of interpolation into the integration.
Since I need only a finite number of convolutions. I turn back to the usual computation of the integral.