DESY 96-266

A fast and precise method to solve the Altarelli-Parisi equations in x space.

C. Pascaud and F. Zomer
Laboratoire de l'Accélérateur Linéaire, IN2P3-CNRS
et Université de Paris-Sud, F-91405 Orsay cedex, France.

November 9, 1996

Abstract

A numerical method to solve linear integro-differential equations is presented. This method has been used to solve the QCD Altarelli-Parisi evolution equations within the H1 Collaboration at DESY-Hamburg. Mathematical aspects and numerical approximations are described. The precision of the method is discussed.

Introduction

Inclusive Deep Inelastic lepton-hadron Scattering (DIS)cross section measurements offer a powerful test of perturbative Quantum Chromo Dynamics (pQCD) [1]. The DIS process $\ell(k)^{\pm}h(P) \to \ell^{\pm}(k')X$ (here we shall only consider the case of charged leptons $\ell^{\pm}$ in order to simplify the discussion) kinematic is described by two Lorentz invariants. One usually chooses the transferred momentum squared $Q^2 = -q^2 \equiv (k'-k)^2$ and the Bjorken variable $x = Q^2/(2P.q)$. In terms of these two kinematic variables, the internal dynamics of the struck hadron enters the cross section via three structure functions : $F_1(x,Q^2)$, $F_2(x,Q^2)$ and $F_3(x,Q^2)$. Because only $F_2(x,Q^2)$ contributes significantly to the cross section for $Q^2 < M_{Z_0}^2 c^4$, we shall concentrate on this structure function in the present paper.

Within the framework of pQCD, $F_2(x,Q^2)$ is given by the convolution in x of the well known Wilson coefficients [2] and of the parton densities inside the hadron (we shall work in the $\overline{MS}$ factorization and renormalization scheme). The densities of partons, consisting of quarks and gluons, are computed from the solution of the Altarelli-Parisi (AP) equations [1]. According to [2], it is useful to define the gluon g, a non-singlet q_{NS} and a singlet Σ quark combination densities. They are the solution of the set of AP integro-differential equations :

$$\frac{\partial q_{NS}}{\partial t} = \int_x^1 \frac{dw}{w} P_{NS}(w,t) q_{NS}(\tfrac{x}{w},t) \tag{1}$$

$$\frac{\partial}{\partial t}\begin{pmatrix}\Sigma(x,t)\\ g(x,t)\end{pmatrix} = \int_x^1 \frac{dw}{w}\begin{pmatrix}P_{qq}(w,t) & n_f P_{qg}(w,t)\\ P_{gq}(w,t) & P_{gg}(w,t)\end{pmatrix}\begin{pmatrix}\Sigma(x/w,t)\\ g(x/w,t)\end{pmatrix} \tag{2}$$

where all kernels P are expanded perturbatively :

$$P = \alpha_s(t) P^{[1]}(w) + \alpha_s^2(t) P^{[2]}(w) + O(\alpha_s^3) \tag{3}$$

Here $t = \log(Q^2/\Lambda^2)$ and $\Lambda \approx 200 MeV$ is the pQCD scale parameter. Expressions of the leading and next leading order splitting functions, $P^{[1]}$ and $P^{[2]}$, can be found in [3]. The expression of the strong coupling constant α_s from ref. [4] will be used. However, eq. (1),(2) hold only for $Q^2 \gg \Lambda^2$ where the perturbative series (3) is convergent. Some non-perturbative input functions are then required to solve the system. In practice, these functions depend on unknown parameters (except in the case of [5]) which are determined from a fit to experimental structure function measurements [6]. Thus, eq. (1),(2) must be solved many times during the usual χ^2 minimization procedure and a fast and precise numerical method is then required.

Two [1]different numerical methods have been used so far to solve eq. (1),(2) (see ref. [8] for a review). The first one uses the fact that the Mellin transform of the system (1,2) leads to a simple set of first order differential equations in the x complex conjugate moment space. The solution is straightforward, after some further necessary simplification of the evolution kernels, but the price to pay is to perform numerically an inverse Mellin transform. The method proposed in the present article belongs to a second kind of approach in which the system is solved in the x space : let us comment with more details the main features of three existent approaches of this kind.

- It is first natural to use methods based on the Taylor expansion of the parton densities in $\log(Q^2)$ [9], [10] (called 'brutale force' method in ref [8]). But it is well known [11] that such techniques lead to rounding errors when the Q^2 evolution covers many order of magnitude. And most of all, reaching a good accuracy is prohibited by a necessary high CPU.

- It was early observed [12] that shifted Jacobi polynomials [13] offer many useful properties to solve AP equations : orthogonality for $x \in [0,1]$, a weight function $x^\beta(1-x)^\alpha$ describing the asymptotic behaviour of parton densities ($\alpha > 0$, $\beta > -1$ are real arbitrary numbers) To understand advantages and disadvantages of that method, let us write the expansion of the non singlet densities in terms of the shifted Jacobi polynomials $\Theta_n^{\alpha,\beta}(x)$:

$$x q_{NS}(x,t) = x^\beta (1-x)^\alpha \sum_{n=0}^{\infty} a_n(t) \times \Theta_n^{\alpha,\beta}(x) \ .$$

  where $\Theta_n^{\alpha,\beta}(x) = \sum_{j=0}^{n} c_{j,n}(\alpha,\beta) x^j$. Using orthogonality relation one immediately relates the coefficients $a_n(t)$ to the non singlet moments. Although the solution can be explained in a very compact form, the sum in the previous equation must be truncated [14] : when taking for example the smallest experimental $x = 10^{-5}$ accessible value [15], one sees that increasing n leads to numerical rounding errors. Note that this method was originally used for analysing data at $x > 10^{-2}$ and, as pointed out in [8], 'a careful study of numerical stability is required to extend this method at lower x values'. Recently, it was claimed [16] that this method could be applied down to $x \approx 10^{-4}$ but no numerical stability studies have been yet provided.

- In ref. [17], the authors made a series expansion of both parton densities and kernels. Changing the integration variable x to $\log(1/x)$ they show that the best polynomial basis to perform this expansion are the Laguerre orthogonal Polynomials [13]. In particular, the convolution product property of these polynomials leads to an explicate mathematical

[1] *Analysis of $F_2(x,Q^2)$ moments will not be considered here. It is well known since a long time [7] that it relies too much on the behaviour of the structure function in the elastic $x \approx 1$ and 'small-x' $x \to 0$ regions.*

form of the solution. Using the generalized Laguerre polynomials, they also include the asymptotic behaviour of the parton densities as depicted in the previous item. The restrictions of this method are twofold : the kernels and the parton densities are approximated leading to series in the solution which must be truncated.

With the forthcoming high precision F_2 measurements by the HERA experiments [15], together with the very accurate fixed target experiment data[18, 19, 20], precise solutions of AP equations are required in a kinematical plane covering five orders of magnitude in x and Q^2. In order to perform QCD analysis in this large kinematical domain, we present a new numerical method to solve the system (1,2). Using the finite element method, together with a scaling property of the convolution integrals appearing in eq. (1),(2), we show that it is possible to obtain an explicit formula for the solution of AP equations without any infinite series. We also show that our formulation leads fortunately to a reduction of the CPU time by one order of magnitude with respect to the 'brutale' method described above.

The rest of this article in organized as follows. The method is described in sect. 1 : notations and formalism of [21] are used. Application to the AP evolution equations is described in sect. 2. The numerical precision, together with the performances are studied in sect. 3.

1 Description of the method

Let us write generically AP integro-differential equations (1),(2) in the form :

$$\frac{\partial F(x,t)}{\partial t} = \int_x^1 \frac{dw}{w} K(\tfrac{x}{w},t) F(w,t), \quad K(\tfrac{x}{w},t) = \sum_{m=1}^{2} \alpha_s^m(t) P^{[m]}(\tfrac{x}{w}) \quad , \tag{4}$$

where F stands for quark or gluon densities in the proton and therefore belongs to a class of smooth functions. In addition, one has $x \in [0,1]$, $t \in [0,\infty]$ and $F(1,t) = 0$ for all t.

We start to solve eq. (4) by defining an ordered sequence of points [2] $|x >= \{x_i\}_1^n$ and the corresponding sequence $|F >= \{F_i\}_1^n$ with $F_i = F(x_i,t)$. It is then possible to define a piecewise polynomial function $\mathfrak{F}_r(x)$ of order r which interpolates $|F >$ in x at a given value of $t : \mathfrak{F}_r(x_i) = F_i$.
$\mathfrak{F}^r$ is described by a set of polynomial functions $|\mathfrak{P} >= \{\mathfrak{P}_i^r\}_1^{n-1}$ of order r :

$$\mathfrak{F}_r(x) = \mathfrak{P}_i^r(x) \text{ if } x_i \leq x \leq x_{i+1}.$$

$\mathfrak{F}_r$ together with $|x >$ define a linear space $\mathbb{P}_{r,x}$. If one supplies a sequence of continuity conditions $|\nu >= \{\nu_i\}_2^{n-1}$, the defined space $\mathbb{P}_{r,x,\nu}$ is still linear and is a subspace of $\mathbb{P}_{r,x}$. Because of the linearity of $\mathbb{P}_{r,x}$, one can define a basis of continuous functions $|\phi >= \{\phi_i(x)\}_1^n$ such that $\phi_i(x_j) = \delta_{ij}$ (δ_{ij} stands for the kroeneker symbol). In this way, one can write

$$\mathfrak{F}_r(x) = \sum_{i=1}^{n} F_i \phi_i(x), \tag{5}$$

with $|\phi_i > \in \mathbb{P}_{r,x,\nu}$. Equation (5) is valid if and only if

$$dim\ \mathbb{P}_{r,x,\nu} \equiv r \times n - \sum_{j=2}^{n-1} \nu_j = n \quad , \tag{6}$$

[2] *We use the following convention :* $\{x_i\}_1^n \equiv (x_1, ..., x_n)$.

and is thus only satisfied for a restrictive class of sequences $|\nu>$. Functions $\mathfrak{F}_r$ which are constructed in such a way are the original spline functions[21].

We turn now to the solution of (4) by making two assumptions :

$$\mathfrak{F}_r(x) = F(x) \text{ and } \int_x^1 \frac{dw}{w} K(x/w)\phi_i(w) \in \mathbb{P}_{r,x,\nu}.$$

Depending on the choice of $|x>$ and r, theorems concerning the precision of these approximations are given in [21]. This topic will be covered in sect. 3 in the case of AP equations.

Using (6), eq. (4) takes a matrix form

$$\frac{\partial}{\partial t}|F>=\sum_{m=1}^{2}\alpha_s^m(t)M_m|F>, \tag{7}$$

with

$$<i|M_m|j>=\int_{x_i}^1 \frac{dw}{w}P^{[m]}(x_i/w)\phi_j(w), \tag{8}$$

The solution of (7) can therefore be written formally

$$|F>=\exp\left(\sum_{m=1}^{2}\int_{t_0}^{t}dt'\alpha_s^m(t')M_m\right)|F_0>, \tag{9}$$

where $|F>=|F_0>$ at $t=t_0$ ($t_0>0$ can be chosen arbitrary).

Next we choose the sequence $|x>$ and the basis functions $|\phi>$ according to the condition (5). We also focus on the numerical computation aspects, namely the CPU time and the precision.

We first remark that if

$$x_{i+1}=\lambda x_i, \text{ with } \lambda>1 \text{ and } \begin{cases}\phi_i(x) & =\phi_{i+1}(\lambda x),\\ \phi_i(x) & =0 \text{ for } x\geq x_{i+1},\end{cases}$$

matrices M_m are triangular and $<i+1|M_m|j+1>=<i|M_m|j>$. This very interesting scaling property relies on the fact that the integrations are convolution products running from x to 1 ; it remains true even in the presence of a '+' (QCD) prescription appearing in the AP kernels[1].

As we are free to choose the order r of the piecewise polynomials, we take the simplest case, *i.e* linear interpolation ($r=2$). It is known to be nearly optimal for large n values. The corresponding basis is unique and consists on the Lagrange (or sometime called 'hat') functions :

$$\phi_i(x)=\begin{cases}(x-x_{i-1})/(x_i-x_{i-1}) & , x_{i-1}\leq x\leq x_i\\ (x_{i+1}-x)/(x_{i+1}-x_i) & , x_i\leq x\leq x_{i+1}\\ 0 & , \text{otherwise.}\end{cases} \tag{10}$$

Note that integrals of eq. (8) are simple to compute using definition (10) ; cancellations of singularities contained in the AP kernels can even be computed analytically.

If one wishes to design a computer program it is necessary to define completely the points where the function F has to be calculated. This is done by defining a sequence in the variable

$t : |t>= \{t_k\}_1^{n_t}$. The solution may then be propagated from the beginning of the grid to its end by using a recursive expression in analogy with eq. (9) :

$$|F_k>= \exp A_k |F_{k-1}> \text{ with } A_k = \sum_{m=1}^{2} \int_{t_{k-1}}^{t_k} dt' \alpha_s^m(t') M_m \ , k = 1, \ldots, n_t \quad . \tag{11}$$

In the case of a fitting procedure, if the kernel $K(w,t)$ is not changed ($i.e$ α_s fixed at a certain scale t) it is sufficient to compute the matrices $\exp(A_k)$ only once. Furthermore, the type of integrals appearing in eq. 8 are computed at the initialization step because they depend on the choice of $|x>$ only. They can be computed with a high numerical precision. If only $|F_0>$ is changed one must only redo the matrix multiplications implied by eq. (11) : this is a consequence of the linearity of (4).

To efficiently exponentiate matrices A_k, it is powerful to introduce the 'band' matrices B defined by :

$$< i|B_l|j >= \delta_{i+l,j} \quad , \quad l \geq 0 \quad .$$

It is easy to show that B matrices fulfil the multiplication rule

$$B_i B_j = B_j B_i = B_{i+j}.$$

These matrices form a basis for the kernels matrices A_k and we can write

$$A_k = \sum_{l=0}^{n-1} {}^k a_l B_l \text{ with } {}^k a_l = \sum_{m=1}^{2} \int_{t_{k-1}}^{t_k} dt' \alpha_s^m(t') < l|M_m|1 > .$$

The product of two matrices such as A_k will be a matrix of the same type, $i.e$ a linear combination of the B's. The use of band matrices leads to a number of operations of order n^2 whereas ordinary matrices would give a CPU time increasing as n^3.

Let us now isolate the diagonal term in A_k and rewrite :

$$A_k = \sum_{i=0}^{n-1} {}^k a_i B_i = {}^k a_0 B_0 + A'_k.$$

Commutativity allows us to split the exponential in two parts : $\exp(A_k) = \exp({}^k a_0) \exp(A'_k)$. In this expression the second exponential can be expanded as a series which will consists in a sum of products of B matrices. As B_0 is not an element of those products and because of the multiplication rule the index of the B's will increase along the sum to reach its maximum $n-1$. So there will be a limited number of terms in the series and one can write

$$\exp(A_k) = \exp({}^k a_0) \sum_{l=0}^{n-1} \frac{(A'_k)^l}{l!} = \sum_{j=0}^{n-1} {}^k T_j B_j. \tag{12}$$

If we turn back to eq. (7), we can finally write the solution in the following form :

$$F(x_i, t_{k+1}) = \sum_{j=0}^{n-i} {}^k T_j F(x_{j+i}, t_k). \tag{13}$$

This equation shows explicitly that $F(x_i, Q^2)$ depends only on the values of the function F for $x > x_i$.

2 Application to the Altarelli-Parisi equations

It is clear that the non-singlet equation (13) is a solution of eq. (1). In the case of singlet and gluon coupled differential equations, we may still retain formally eq. (9) but A is now made of four sub matrices. Let us rewrite eq. (9) with slightly different notations :

$$|\mathcal{F}_{k+1}>= e^{\mathcal{A}_k}|\mathcal{F}_k> \text{ with } \mathcal{A} = \begin{pmatrix} A_{qq} & A_{qg} \\ A_{gq} & A_{gg} \end{pmatrix},$$

where the $\mathcal{A}$ matrix is no longer a band matrix, but A_{qq}, A_{qg}, A_{gq} and A_{gg} are. We can also consider that the vectorial space involved is a direct product of a n-space and of a 2-space and write :

$$\mathcal{A} = A_0\mathcal{E} + A_x\mathcal{X} + A_y\mathcal{Y} + A_z\mathcal{Z} \quad ,$$

where $A_{0,x,y,z}$ are band matrices and

$$\mathcal{E} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \ , \quad \mathcal{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \ , \quad \mathcal{Y} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \ , \quad \mathcal{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \ .$$

The following rules hold :

$$\mathcal{E} = \mathcal{E}^2 = \mathcal{X}^2 = -\mathcal{Y}^2 = \mathcal{Z}^2; \tag{14}$$

$\mathcal{E}$ commutes with $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$ and those anti commute between themselves. Therefore, one can split into pieces the exponential

$$e^{\mathcal{A}} = e^{A_0} e^{A_x\mathcal{X}+A_y\mathcal{Y}+A_z\mathcal{Z}} \ .$$

The second exponential in the r.h.s of the above equation is computed by series expansion. Then, one needs to calculate powers of $A_x\mathcal{X} + A_y\mathcal{Y} + A_z\mathcal{Z}$. Making use of eq (14) one gets :

$$(A_x\mathcal{X} + A_y\mathcal{Y} + A_z\mathcal{Z})^2 = (A_x^2 - A_y^2 + A_z^2)\mathcal{E} = A_e^2\mathcal{E}.$$

This shows that we can split the series in its even and odd parts and sum them independently. The result is the following :

$$e^{\mathcal{A}} = e^{A_0}\left(\frac{e^{A_e}+e^{-A_e}}{2} + A_e^{-1}(A_x\mathcal{X} + A_y\mathcal{Y} + A_z\mathcal{Z})\frac{e^{A_e}-e^{-A_e}}{2}\right).$$

A_e^2, its square root A_e and the inverse A_e^{-1} are easily computed [3]using the B's multiplication rules.

3 Precision and performances

The precision of the method proposed in this paper can be easily estimated by changing the number n of breakpoints in $|x>$. The convergence of the evolved parton densities leads to an

[3] *Depending on the sign of A_e^2 's diagonal, A_e will be real or pure imaginary ; it turns out for AP equations that we are in the first case.*

estimation of the precision. Input functions and Λ ($\Lambda^4 = 225MeV$) value of [22] are used :

$$\begin{aligned} xg(x) &= 1.86x^{-0.22}(1-x)^{7.12} \\ x\Sigma(x) &= 1.15x^{-0.11}(1-x)^{3.10}(1+3.12x) \\ xq_{NS}(x) &= 1.14x^{0.65}(1-x)^{4.66}(1+8.68x) \end{aligned}$$

at $Q_0^2 = 4\ GeV^2$. Here, $\Sigma = u+\bar{u}+d+\bar{d}+s+\bar{s}$ and $q_{NS} = u+\bar{u}-\Sigma/3$ is the chosen non-singlet quark density (for more details concerning the fit we refer to [22]).

We define the two bounded sequences, $|x>$ such that $x_i \in [x_{min} = 10^{-5}, 1], i = 1, \ldots, n$, and $|t>$ such that $t_j \in [0, 10], j = 1, \ldots, n_t$. We start from $n = 100$ and increase up to $n = 900$ by steps of 100. n_t is set to 100 for all n and this does not introduce numerical uncertainties since the expression of eq. (12) is exact. Evolutions of eq. (1),(2) are performed up to $Q^2 = 10^4 GeV^2$. Fig. (1a) shows the ratio $R_{q_{NS}}(x,n)$ defined by :

$$R_{q_{NS}}(x,n) = \frac{q_{NS}(x)|_{n,n_t}}{q_{NS}(x)|_{n=900,n_t}}\ , \text{for } n = 800, 700, 600, 500, 400, 300, 200, 100 \tag{15}$$

for $Q^2 = 10^4 GeV^2$. Note that the calculations done with large values of n are stable because of a choice of a first order spline interpolations (see section 1). With higher order splines, the choice of the breakpoint set $|x>$ is very important and solutions may be unstable [21]. Figs.(1b,c) show $R_\Sigma(x,n)$ and $R_g(x,n)$ respectively. These plots illustrate the convergence of the method. The wiggles appearing in these plots are generated by the Lagrange functions which are also used to interpolate between the breakpoints of the sequence $|x>$. From these plots, one sees that for $n = 300$, the precision is of the order of 0.5% in the 'low-x' region. This is accurate enough comparing to measurement uncertainties [15],[18]. However, at 'high x', huge differences appear. Although these instabilities of the calculations do not modify the results in the 'low-x' region, one cannot use this method to compute the structure functions at 'low-x' and 'high x' simultaneously with a good precision. A straightforward modification of the method is to define a second net in x. This new net starts from $x'_{min} = x_{min}^{1/n'}$. Then n' breakpoints are set equidistantly in $\log(x)$ between two adjacent points of the first net up to $x = 1$. For the calculations, the second net is first used to solve the AP equations. The extension from x'_{min} down to x_{min} is then made by solving AP equations using the first net. For example, taking $x_{min} = 10^{-5}$ and $n' = 5$, the precision for $x \geq x'_{min} = 10^{-1}$ will be equivalent to the use of one net made of $n \times n'$ breakpoints. Note that the CPU time is only multiplied by two instead of $(n')^2$. To show the resulting precision the fit described above is performed with $n = 700$ and $n' = 1, 4, 8$ ($x'_{min} = 10^{-5}, 5.6 \times 10^{-2}, 2.4 \times 10^{-1}$). Then the ratio

$$R'_{q_{NS}}(x,n',n) = \frac{q_{NS}(x)|_{n',n,n_t}}{q_{NS}(x)|_{n'=8,n_{ref}=700,n_t}} \tag{16}$$

is computed. For $n = 700$ and $n' = 8$, one can reasonably assume that the precision is nearly optimum at $x > 2.4 \times 10^{-1}$. We shall take the parton densities, computed with this conditions, as the reference 'high-x' sets. Fig.(2) show $R'_{q_{NS}}$, R'_Σ and R'_g for $n = 700$: the gain in precision at 'high-x' is very significant when going from $n' = 1$ to $n' = 4$. On Fig.(2a,b,c), the ratio R' for $n = 250$ and $n' = 1, 4, 8$ is also shown. As pointed out above, $n = 250$ leads to structure function computations precise enough at 'small-x'. One can then observe that setting $n' = 8$ leads to great improvement of these computation at 'high-x'. We conclude that for $n = 250, n' > 4$, it

is possible to perform a very precise calculation of the structure functions within five orders of magnitude in x and we point out that any global pQCD structure function analysis should pay much attention to the numerical precision (as it was already mentioned in [8]).

One can design a very fast (CPU) and efficient procedure to determine the input functions from a χ^2 minimization : first the minimum is approached rapidly by setting $n = 100$, then the 'true' minimum is reached by increasing n in steps of 100 until $n = 250$. If 'high x' measurements enter the fit, a second net must be considered at the end of this procedure.

According to the AP equations, the momentum sum-rule

$$S(t) = \int_0^1 x \Big(\Sigma(x,t) + g(x,t) \Big) dx \,,$$

should stay constant during the evolution in t. This is a sensitive test of the computational precision. With the parametrizations chosen for the input functions [22], $S(Q_0^2)$ is computable analytically and is equal to a sum of Beta functions. $S(Q_0^2)$ is then set to 1 and $S(Q^2)$ is computed numerically with a linear extrapolation in the interval $x \in [0, 10^{-5}]$. This extrapolation introduces a small bias in the calculation of $S(Q^2)$. Fig. 3 shows $S(Q^2)$ as a function of Q^2 for the nine sequences of breakpointsof eq. (15) (here one x net is considered). In any case, the deviation from 1 for an evolution in t which covers four orders of magnitude is of the order of 2/10000.

Conclusion

We have presented a numerical method to solve the Altarelli-Parisi equations in x space. Unlike conventional methods based on Taylor expansion, the Q^2 evolution is computed 'exactly'. All convolution products involving Altarelli-Parisi Kernels are computed only once with high accuracy. Using the Lagrange functions to construct a basis and taking advantage of scaling properties of AP equations, the number of operations increases with n^2 instead of n^3 like for other simple basis functions. In some senses, our method is equivalent to the polynomial approaches and specially the one of ref. [17]. But, the x space discretization allows the use of first order polynomials to avoid precision computational problems and the freedom in the choice of the breakpoint sequence leads to a formal solution involving only finite series. The only numerical instability comes from the number of breakpoints in the $|x >$ sequence. The resulting numerical precision is found to be very good when a large number of breakpoints are considered or when two $|x >$ sequences are used.

Acknowledgement

We thank Bernard Andrieu, Marco Costa and Albert de Roeck for comments and helpful discussions.

References

[1] G. Altarelli and G. Parisi, Nucl. Phys. B126 (1977) 298.

[2] W. Furmanski and R. Petronzio, Z. Phys. C11 (1982) 293.

[3] G. Curci, W. Furmanski and R. Petronzio, Nucl. Phys. B175 (1980) 27 ; W. Furmanski and R. Petronzio, Phys. Lett. B97 (1980) 437.

[4] W.J. Marciano, Phys. Rev. D29 (1984) 580.

[5] A. Capella et al., Phys. Lett. B337(1994)358.

[6] J.F. Owens and W.K Tung, Ann. Rev. Nucl. Sci. 42(1992)291.

[7] R. Petronzio, Th. 3085-CERN.

[8] C. Chỳla and J. Rameš, Z. Phys. C31(1986)151.

[9] R.T. Harrod, S. Wada and B.R. Weber, Z. Phys. C9(1981)351 ;
A. Devoto, D.W Duke, J.F. Owens and R.G. Roberts, Phys. Rev. D27(1983)508.

[10] M. Virchaux, Thèse, Université Paris-7 (1988) ;
A. Ouraou, Thèse, Université Paris-11 (1988).

[11] J. Chỳla and J. Rameš, Czech. J. Phys. B36(1986)567.

[12] N. Sourlas and G. Parisi, Nucl. Phys. B131(1979)421.

[13] See for example A.M. Mathai, *A Handbook of Generalized Special Functions for Statistical and Physical Sciences*, (1993) Clarendon Press Oxford.

[14] V.G. Krivokhizhin et al., Z. Phys. C36(1987)51.

[15] ZEUS Collaboration, Z. Phys. C69(1996)607; DESY 96-076.
H1 Collaboration, DESY 96-039.

[16] A.K. Mavroidis and W. von Schlippe, Contribution to the International Europhysics Conference on High Energy Physics, HEP 95/ EPS0235.

[17] W. Furmanski and R. Petronzio, Nucl. Phys. B195(1982)237.

[18] M. Arneodo et al., Phys. Lett. B364(1995)107.

[19] A.C. Benvenuti et al., Phys. Lett. B223 (1989) 485 ; CERN preprint CERN-EP/89-06.

[20] L.W. Whitlow, PhD thesis, SLAC-357(1990).

[21] C. de Boor, 'A Practical Guide to Splines', Springer-Verlag, 1978.

[22] S. Aid et al., H1 Collaboration, Phys. Lett. B354(1995)494.

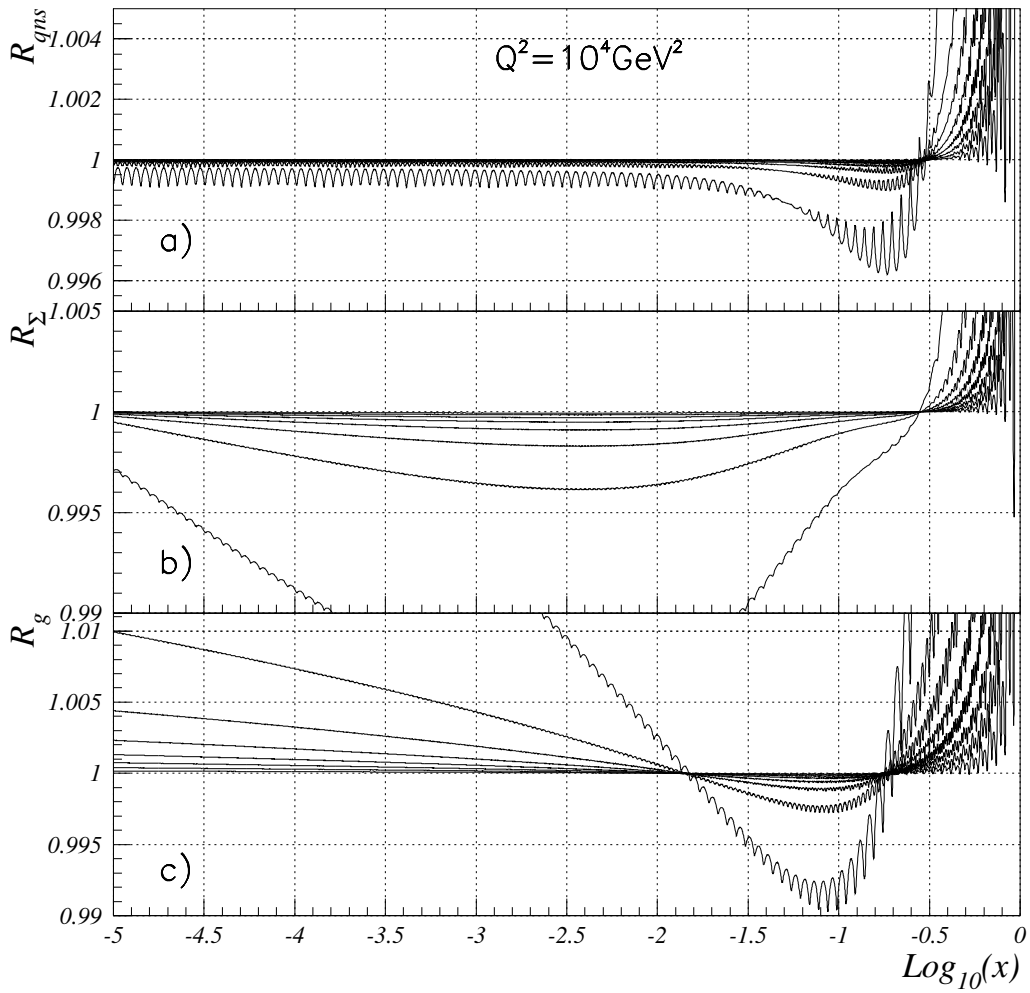

Figure 1: *a) :* $R_{q_{NS}}(x,n)$ *; b)* $R_\Sigma(x,n)$ *; c) :* $R_g(x,n)$*. See eq. (15) for definitions. The less accurate curve (far from the line* $R=1$*) corresponds to* $n=100$*. The next one to* $n=200$ *etc...*

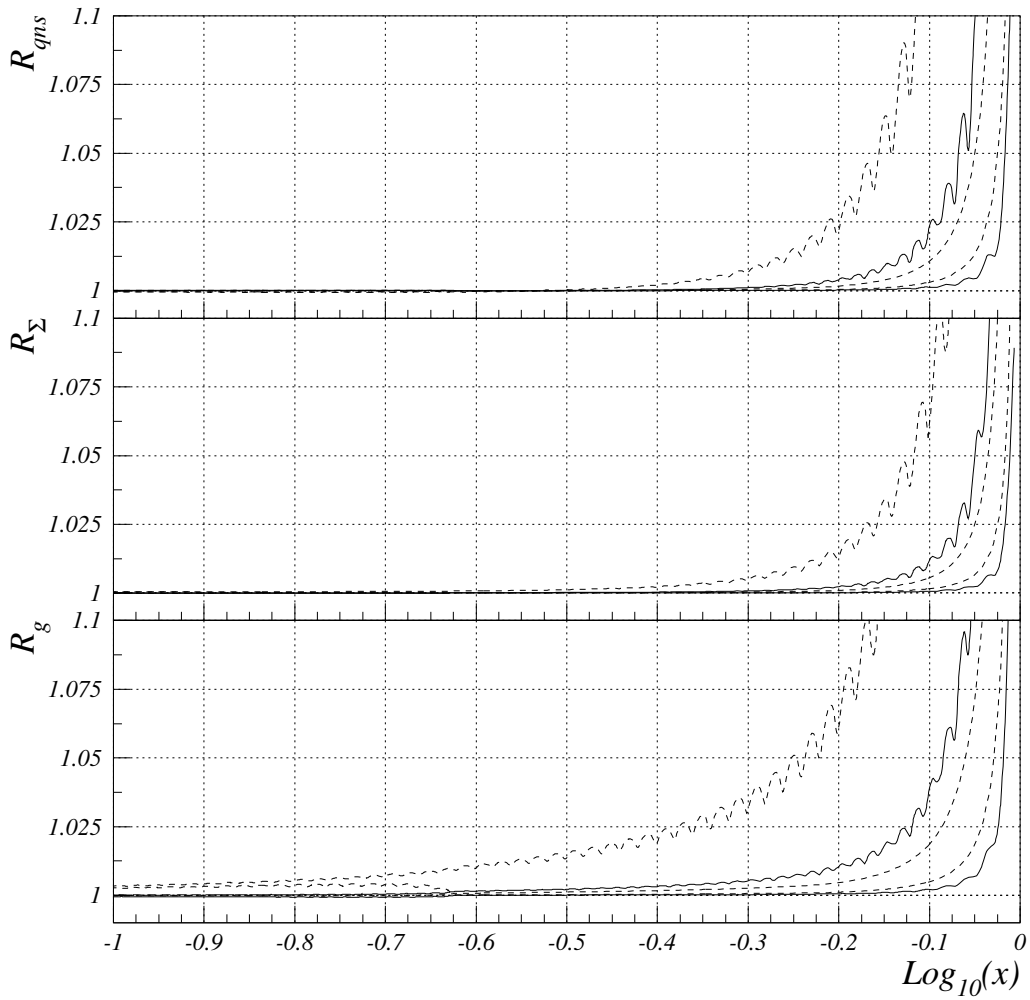

Figure 2: *a) :* $R'_{q_{NS}}(x,n,n')$ *; b)* $R'_\Sigma(x,n,n')$ *; c) :* $R'_g(x,n,n')$*. See eq. (16) for definitions. The full lines correspond to* $n = 700$ *and the dashed lines correspond to* $n = 250$*. The less accurate curve (far from the line* $R = 1$*) corresponds to* $n' = 1$*. The next one to* $n' = 4$*. The last one (defined only in the case* $n = 250$*) to* $n' = 8$*.*

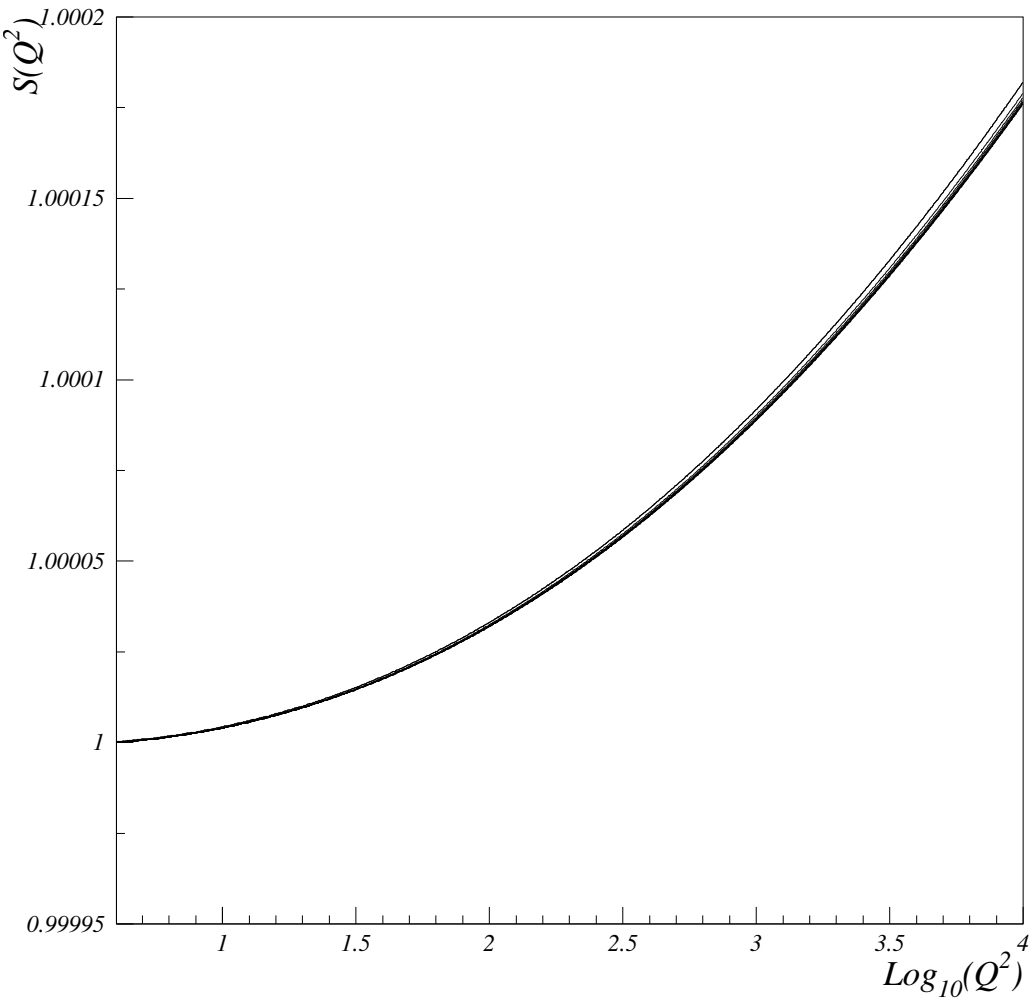

Figure 3: *Momentum sum-rule as function of* Q^2. *The curves described in the text superpose almost exactly.*