Testing and improving the numerical accuracy of the NLO predictions

R. Pittau

Departamento de Física Teórica y del Cosmos and CAFPE
Universidad de Granada, E-18971, Granada, Spain

Abstract

I present a new and reliable method to test the numerical accuracy of NLO calculations based on modern OPP/Generalized Unitarity techniques. A convenient solution to *rescue* most of the detected numerically inaccurate points is also proposed.

Keywords:
NLO, QCD, Electroweak Corrections, OPP, Generalized Unitarity.

1. Introduction

With the advent of the modern OPP [1, 2] and Generalized Unitarity [3, 4, 5, 6, 7, 8, 9, 10] based techniques, the *art* of computing NLO corrections received a lot of attention in the last few years and several programs [11, 12, 13, 14, 15, 16] and computations [17, 18, 19, 20, 21, 22, 23] exist, by now, based on this philosophy.

While for the traditional reduction methods [24] a lot of work has been spent already to find ways to control the numerical accuracy of the results [25, 26], in the case of these new techniques the situation is still at an early stage. However, their potential to *self detect* stability problems is known since 2007 [11], the basic observation being that, since a reconstruction of a function $N(q)$ of the *would be* integration momentum q is involved (the coefficients of which are interpreted as the coefficients of the scalar 1-loop functions entering the calculation), one can numerically test the accuracy of it by comparing $N(q)$ and its re-constructed counterpart at a new, arbitrarily chosen value of q. However, the arbitrariness of the point chosen for the test poses serious problems, because it introduces a new, unwanted, parameter upon which the check depends in an unpredictable way [1]. Furthermore, not all the reconstructed coefficients enter into the actual computation, because, for example, some of them may multiply vanishing loop functions, rendering immaterial a possible inaccuracy in their determination. In addition, it is not clear how to test the rational part of the amplitude [2].

Nevertheless, it keeps being very tempting the idea of *self detecting* numerical inaccuracies, avoiding the need of additional analytic work. In this paper, I present a method to achieve this task, based on the construction of a reliable precision estimator working at an event by event basis. Therefore, it becomes possible for the user to safely set a precision threshold above which the inaccurate points are discarded. Furthermore, I prove that, re-fitting the discarded points at higher precision *while keeping the computation of $N(q)$ in double precision* allows to re-include most of them in the original sample. This solution nicely factorizes the problem, in the sense that the codes (of the parts of the code) computing the function $N(q)$ can be kept in double precision and only the fitting procedure to get the coefficients needs to be re-done at higher precision.

The structure of this work is very simple: in Sections 2 and 3, I describe the algorithm and, in Section 4, I report on the tests I performed on the whole procedure.

2. The method

In the OPP technique the numerator $N(q)$ of the integrand of a m-point amplitude is decomposed in terms of denominators $D_i = (q+p_i)^2 - m_i^2$ [2]

$$\begin{aligned} N(q) &= \mathcal{D}^{(m)}(q) + \sum_{i_0<i_1<i_2}^{m-1} c(i_0 i_1 i_2; q) \prod_{i \neq i_0, i_1, i_2}^{m-1} D_i + \sum_{i_0<i_1}^{m-1} b(i_0 i_1; q) \prod_{i \neq i_0, i_1}^{m-1} D_i \\ &\quad + \sum_{i_0}^{m-1} a(i_0; q) \prod_{i \neq i_0}^{m-1} D_i \,, \end{aligned} \tag{1}$$

[1] Improvements on this technique have been recently presented in [16].

[2] In our notation, q in 4-dimensional, $\bar{q}$ n-dimensional and n-dimensional denominators are written as $\bar{D}_i = (\bar{q}+p_i)^2 - m_i^2$.

where, for later convenience, I have grouped all the 4-point contributions into a single term

$$\mathcal{D}^{(m)}(q) \;=\; \sum_{i_0<i_1<i_2<i_3}^{m-1} d(i_0 i_1 i_2 i_3; q) \prod_{i \neq i_0, i_1, i_2, i_3}^{m-1} D_i \,. \tag{2}$$

The functions $d(i_0 i_1 i_2 i_3; q)$, $c(i_0 i_1 i_2; q)$, $b(i_0 i_1; q)$ and $a(i_0; q)$ depend on the integration momentum q and bring information on the coefficients of the scalar 1-loop integrals, that are obtained by fitting $N(q)$ at different values of q that nullify, in turn, the denominators. Finally, performing a global shift of all the masses appearing in the denominators of Eq. 1

$$m_i^2 \to m_i^2 - \tilde{q}^2 \tag{3}$$

and fitting again, allows to reconstruct also a piece of the rational terms, called R_1 [2]. In summary, by knowing the set [3]

$$\begin{array}{lll} d(i_0 i_1 i_2 i_3)\,, & c(i_0 i_1 i_2)\,, & \\ b(i_0 i_1)\,, & a(i_0)\;\;, & R_1\,, \end{array} \tag{4}$$

the amplitude A is reconstructed by simply multiplying by the corresponding scalar 1-loop integrals [12, 27] [4].

$$\begin{aligned} A \;=\; & \sum_{i_0<i_1<i_2<i_3}^{m-1} d(i_0 i_1 i_2 i_3) \int d^n \bar{q} \frac{1}{\bar{D}_{i_0} \bar{D}_{i_1} \bar{D}_{i_2} \bar{D}_{i_3}} \\ & + \sum_{i_0<i_1<i_2}^{m-1} c(i_0 i_1 i_2) \int d^n \bar{q} \frac{1}{\bar{D}_{i_0} \bar{D}_{i_1} \bar{D}_{i_2}} \\ & + \sum_{i_0<i_1}^{m-1} b(i_0 i_1) \int d^n \bar{q} \frac{1}{\bar{D}_{i_0} \bar{D}_{i_1}} \\ & + \sum_{i_0}^{m-1} a(i_0) \int d^n \bar{q} \frac{1}{\bar{D}_{i_0}} + R_1 \,. \end{aligned} \tag{5}$$

[3] I use a notation such that the coefficients of the scalar 1-loop functions have the same name of the functions appearing in Eqs. 1 and 2, but without q dependence.

[4] The remaining piece of the rational terms R_2 can be computed as explained in [2, 28, 29].

The key point of the method I propose in this paper is the observation that, if one would be able to obtain the whole set of Eq. 4 in an independent way, giving, as a result, a new set

$$\begin{array}{lll} d'(i_0 i_1 i_2 i_3)\,, & c'(i_0 i_1 i_2)\,, & \\ b'(i_0 i_1)\,, & a'(i_0)\,, & R'_1\,, \end{array} \tag{6}$$

an independent determination of the 1-loop amplitude would become possible

$$\begin{aligned} A' \;=\; & \sum_{i_0<i_1<i_2<i_3}^{m-1} d'(i_0 i_1 i_2 i_3) \int d^n\bar{q} \frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}\bar{D}_{i_2}\bar{D}_{i_3}} \\ & + \sum_{i_0<i_1<i_2}^{m-1} c'(i_0 i_1 i_2) \int d^n\bar{q} \frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}\bar{D}_{i_2}} \\ & + \sum_{i_0<i_1}^{m-1} b'(i_0 i_1) \int d^n\bar{q} \frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}} \\ & + \sum_{i_0}^{m-1} a'(i_0) \int d^n\bar{q} \frac{1}{\bar{D}_{i_0}} + R'_1\ , \end{aligned} \tag{7}$$

that could then be used to define a reliable estimator of the accuracy as follows [5]

$$E^A \equiv \frac{|A - A'|}{|A|}\,. \tag{8}$$

The advantage of Eq. 8, with respect to a test performed at the level of the function $N(q)$, is that *only* the coefficients contributing to the amplitude enter into the game. Furthermore, a test on R_1 becomes possible. As it will become clear shortly, it is convenient to differentiate the two cases where the coefficients of the sets in Eqs. 4 and 6 (and, *a fortiori*, the amplitudes in Eqs. 5 and 7) are computed in double or multi-precision. Then, I denote the double precision estimator by

$$E^A_d \equiv \frac{|A_d - A'_d|}{|A_d|}\,, \tag{9}$$

[5] In an actual, numerical implementation, a small quantity ϵ has to be included in the denominator of Eq. 8 to deal with the case of vanishing amplitude.

and its multi-precision version by

$$E_m^A \equiv \frac{|A_m - A'_m|}{|A_m|}\,. \tag{10}$$

In the rest of this section, I illustrate how to obtain the new set of Eq. 6.

The technique is similar to the procedure adopted to compute R_1. Under the shift in Eq. 3, Eq. 1 becomes

$$\begin{aligned} N(q) &= \bar{\mathcal{D}}^{(m)}(q) + \sum_{i_0<i_1<i_2}^{m-1} \bar{c}(i_0 i_1 i_2;q) \prod_{i\neq i_0,i_1,i_2}^{m-1} (D_i + \tilde{q}^2) \\ &\quad + \sum_{i_0<i_1}^{m-1} \bar{b}(i_0 i_1;q) \prod_{i\neq i_0,i_1}^{m-1} (D_i + \tilde{q}^2) \\ &\quad + \sum_{i_0}^{m-1} \bar{a}(i_0;q) \prod_{i\neq i_0}^{m-1} (D_i + \tilde{q}^2)\,, \end{aligned} \tag{11}$$

where

$$\bar{\mathcal{D}}^{(m)}(q) = \sum_{j=2}^{m} \tilde{q}^{\,(2j-4)} d^{\,(2j-4)}(q)\,, \tag{12}$$

with the last coefficient of $\bar{\mathcal{D}}^{(m)}(q)$ independent on q

$$d^{\,(2m-4)}(q) = d^{\,(2m-4)}\,, \tag{13}$$

and where

$$\begin{aligned} \bar{c}(i_0 i_1 i_2;q) &= c(i_0 i_1 i_2;q) + \tilde{q}^2 c^{\,(2)}(i_0 i_1 i_2;q) \\ \bar{b}(i_0 i_1;q) &= b(i_0 i_1;q) + \tilde{q}^2 b^{\,(2)}(i_0 i_1;q) \\ \bar{a}(i_0;q) &= a(i_0;q)\,. \end{aligned} \tag{14}$$

This last equation implies, for the 1-,2 and 3-point coefficients

$$\begin{aligned} \bar{c}(i_0 i_1 i_2) &= c(i_0 i_1 i_2) + \tilde{q}^2 c^{\,(2)}(i_0 i_1 i_2) \\ \bar{b}(i_0 i_1) &= b(i_0 i_1) + \tilde{q}^2 b^{\,(2)}(i_0 i_1) \\ \bar{a}(i_0) &= a(i_0)\,. \end{aligned} \tag{15}$$

The constants $b^{(2)}(i_0 i_1)$, $c^{(2)}(i_0 i_1 i_2)$ and $d^{(2m-4)}$ enter into the computation of R_1 [2]

$$\begin{aligned} R_1 &= -\frac{i}{96\pi^2} d^{(2m-4)} - \frac{i}{32\pi^2} \sum_{i_0<i_1<i_2}^{m-1} c^{(2)}(i_0 i_1 i_2) \\ &- \frac{i}{32\pi^2} \sum_{i_0<i_1}^{m-1} b^{(2)}(i_0 i_1) \left(m_{i_0}^2 + m_{i_1}^2 - \frac{(p_{i_0} - p_{i_1})^2}{3} \right) , \end{aligned} \tag{16}$$

and can be determined with the help of Eqs. 15 and 12.

With the knowledge of $a(i_0)$, $b(i_0 i_1)$, $c(i_0 i_1 i_2)$, $b^{(2)}(i_0 i_1)$ and $c^{(2)}(i_0 i_1 i_2)$, a', b' and c' in Eq. 6 can be immediately obtained with a new mass shift

$$m_i^2 \to m_i^2 - \tilde{q}_1^2 , \tag{17}$$

giving

$$\begin{aligned} \bar{c}_1(i_0 i_1 i_2) &= c(i_0 i_1 i_2) + \tilde{q}_1^2 c^{(2)}(i_0 i_1 i_2) \\ \bar{b}_1(i_0 i_1) &= b(i_0 i_1) + \tilde{q}_1^2 b^{(2)}(i_0 i_1) \\ \bar{a}_1(i_0) &= a(i_0) , \end{aligned} \tag{18}$$

where I attached the subscript $_1$ to the coefficients obtained with the new shift. Combining Eqs. 15 and 18 gives

$$\begin{aligned} a'(i_0) &= \bar{a}_1(i_0) \\ b'(i_0 i_1) &= \frac{\bar{b}(i_0 i_1) + \bar{b}_1(i_0 i_1)}{2} - \frac{\tilde{q}^2 + \tilde{q}_1^2}{2} \, b^{(2)}(i_0 i_1) , \\ c'(i_0 i_1 i_2) &= \frac{\bar{c}(i_0 i_1 i_2) + \bar{c}_1(i_0 i_1 i_2)}{2} - \frac{\tilde{q}^2 + \tilde{q}_1^2}{2} \, c^{(2)}(i_0 i_1 i_2) . \end{aligned} \tag{19}$$

As for R_1, an independent determination of $b^{(2)}(i_0 i_1)$, $c^{(2)}(i_0 i_1 i_2)$ in Eq. 16 also follows from the new shift

$$\begin{aligned} c'^{(2)}(i_0 i_1 i_2) &= \frac{\bar{c}(i_0 i_1 i_2) - \bar{c}_1(i_0 i_1 i_2)}{\tilde{q}^2 - \tilde{q}_1^2} \\ b'^{(2)}(i_0 i_1) &= \frac{\bar{b}(i_0 i_1) - \bar{b}_1(i_0 i_1)}{\tilde{q}^2 - \tilde{q}_1^2} . \end{aligned} \tag{20}$$

With the help of Eq. 18 one can now completely reconstruct the 1-, 2- and 3-point parts of the numerator function with masses shifted according to

Eq. 17, namely

$$\sum_{i_0<i_1<i_2}^{m-1} \bar{c}_1(i_0 i_1 i_2; q) \prod_{i\neq i_0,i_1,i_2}^{m-1} (D_i + \tilde{q}_1^2) + \sum_{i_0<i_1}^{m-1} \bar{b}_1(i_0 i_1; q) \prod_{i\neq i_0,i_1}^{m-1} (D_i + \tilde{q}_1^2)$$
$$+ \sum_{i_0}^{m-1} \bar{a}_1(i_0; q) \prod_{i\neq i_0}^{m-1} (D_i + \tilde{q}_1^2)\,. \tag{21}$$

By subtracting Eq. 21 from $N(q)$ one determines $\bar{\mathcal{D}}_1^{(m)}(q)$ obeying the following polynomial (in $\tilde{q}_1^2$) representation (see Eq. 12)

$$\bar{\mathcal{D}}_1^{(m)}(q) \;=\; \sum_{j=2}^{m} \tilde{q}_1^{(2j-4)} d_1^{(2j-4)}(q)\,, \tag{22}$$

the first coefficient of which, computed at values $q = q_{i_0,i_1,i_2,i_3}$ nullifying, in turns, all possible combinations of 4 denominators

$$D_{i_0}(q_{i_0,i_1,i_2,i_3}) = D_{i_1}(q_{i_0,i_1,i_2,i_3}) = D_{i_2}(q_{i_0,i_1,i_2,i_3}) = D_{i_3}(q_{i_0,i_1,i_2,i_3}) = 0\,, \tag{23}$$

gives the desired independent determination of the box coefficients

$$d'(i_0 i_1 i_2 i_3) = d_1^{(0)}(q_{i_0,i_1,i_2,i_3})\,. \tag{24}$$

From the last term in Eq. 22 one obtains, instead

$$d'^{\,(2m-4)} = d_1^{(2m-4)}\,, \tag{25}$$

that, together with the coefficients in Eq. 20, gives a complete alternative determination of R_1

$$R_1' \;=\; -\frac{i}{96\pi^2} d'^{\,(2m-4)} - \frac{i}{32\pi^2} \sum_{i_0<i_1<i_2}^{m-1} c'^{\,(2)}(i_0 i_1 i_2)$$
$$-\; \frac{i}{32\pi^2} \sum_{i_0<i_1}^{m-1} b'^{\,(2)}(i_0 i_1) \left(m_{i_0}^2 + m_{i_1}^2 - \frac{(p_{i_0} - p_{i_1})^2}{3} \right)\,, \tag{26}$$

I close the Section by summarizing the procedure. One fits the numerator function $N(q)$ three times; the first time with $\tilde{q}^2 = 0$ (Eq. 1) to determine the cut-constructible part of the amplitude, namely all of the coefficients in

Eq. 4; the second time with $\tilde{q}^2 \neq 0$, to compute R_1 by means of Eq. 16; the third time with a new value of the mass shift $(\tilde{q}_1^2)$ to calculate the alternative set of coefficients in Eqs. 19, 20, 24 and 25, that allow to build the precision estimator E_d^A given in Eq. 9.

It is clear that performing the three fits by using directly the numerator function $N(q)$ appearing in the l.h.s. of Eq. 1 could be computationally very expensive because, in practical cases, the calculation of $N(q)$ is rather time consuming [6]. Fortunately, after the first fit, one is allowed to use the *reconstructed* numerator function (namely the r.h.s. of Eq. 1) to determine both R_1 and E_d^A. The additional CPU time is then very moderate. To further decrease it, one can also observe that, instead of determining $d^{\,(2m-4)}$ through the expansion in Eq. 12, that requires the knowledge of $\bar{\mathcal{D}}^{(m)}(q)$ at $(m-2)$ different values of $\tilde{q}^2$, one can get it by means of the following relation among the OPP coefficients

$$\begin{aligned} d^{\,(2m-4)} = & - \sum_{i_0<i_1<i_2}^{m-1} c^{(2)}(i_0 i_1 i_2; q) - \sum_{i_0<i_1}^{m-1} b(i_0 i_1; q) - \sum_{i_0<i_1}^{m-1} b^{(2)}(i_0 i_1; q) \sum_{i \neq i_0, i_1}^{m-1} D_i \\ & - \sum_{i_0}^{m-1} a(i_0; q) \sum_{i \neq i_0}^{m-1} D_i \,. \end{aligned} \tag{27}$$

Eq. 27 is proved in Appendix A.

3. Rescuing the inaccurate points

Under the assumption that E_d^A in Eq. 9 is a good precision estimator, points can be rejected when $E_d^A > E_{\rm lim}$, where $E_{\rm lim}$ is a threshold value chosen by the user. In this Section, I propose, as a simple recipe to *rescue* the rejected points, to re-perform the three fits described in Section 2 at higher precision *while keeping the computation of $N(q)$ in double precision.* The advantage of this recipe is that multi-precision routines need to be implemented *just in the fitting program*, while the code providing $N(q)$ can be left untouched [7]. A new test in multi-precision can then be performed on the

[6] I assume here an unoptimized computation of $N(q)$, performed without cashing the information that does not depend on q.

[7] This strategy is especially relevant in the case of programs that already implement, internally, multi-precision routines [30], such as `CutTools` [11].

rescued points using the multi-precision estimator of Eq. 10 and only if, even in this case, $E_m^A > E_{\rm lim}$ the event is discarded for good (or re-computed, if possible, with $N(q)$ also evaluated in multi-precision). The hope is that the percentage of points rejected by this second test is very limited, so that they can be safely eliminated from the sample. The effectiveness of this strategy is studied in the next Section.

4. Testing the method

To test the procedures described in the previous two Sections I implemented in `CutTools`, as a numerator function $N(q)$ mimicking the full amplitude A, one of the 120 diagrams contributing to the $\gamma\gamma \to 4\gamma$ scattering in massless QED [8]. To enhance the problematic region I did not apply any cut on the final state particles, so that numerically unstable Phase Space configurations with zero Gram determinant can be freely approached. From a practical point of view, I constructed the alternative amplitude of Eq. 7 by keeping the 4-point coefficients and re-computing only c', b', a' and R'_1 with the help of Eqs. 19, 20 and 25. The reason is that, in practice, the derivation of the 4-point part of an amplitude is numerically quite stable, the bulk of the numerical instabilities coming from the lower-point sectors.

Before describing in details the tests, I introduce, besides the estimators given in Eqs. 9 and Eqs. 10, a few more variables. I define two *true precision variables* as follow

$$P_d = |A_d - A_e|/|A_e| \quad \text{and} \quad P_m = |A_m - A_e|/|A_e|\,, \tag{28}$$

where A_e is the *exact* reference amplitude computed with both fits *and* $N(q)$ in multi-precision. P_d and P_m will be used, in the following, to test the actual precision in the computation of A_d and A_m. Furthermore, for the sake of comparison, I define two additional precision estimators, based on the so called $N = N$ test of [11]

$$E_d^N = |N_d - N_{d,rec}|/|N_d| \quad \text{and} \quad E_m^N = |N_d - N_{m,rec}|/|N_d|\,, \tag{29}$$

where N_d is the numerator function $N(q)$ computed, in double precision, at a random value of q [9], $N_{d,rec}$ the same numerator, but reconstructed, in double

[8] This diagram contains up to rank six 6-point functions, so it fairly represents the complexity of the real situations.

[9] I picked up the point $q = \sqrt{s}\,(1/2, -1/3, 1/4, -1/5)$.

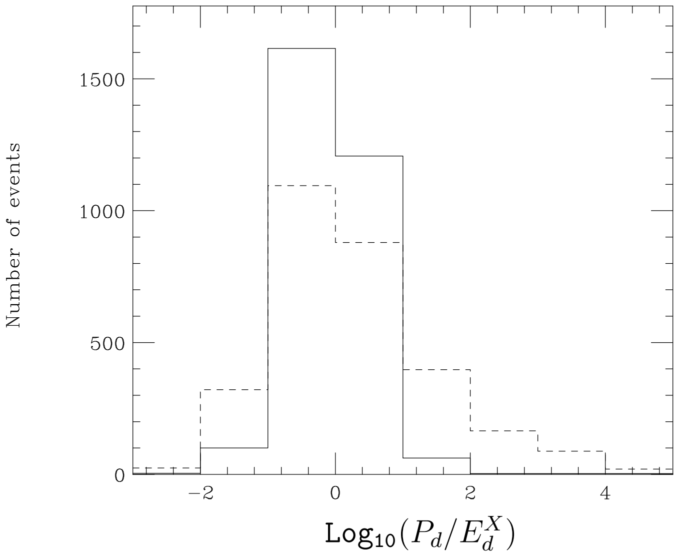

Figure 1: Distribution of the ratio between the true precision $P_d = |A_d - A_e|/|A_e|$ and two different precision estimators. The dashed histogram refers to the estimator at the numerator level $E_d^X = E_d^N = |N_d - N_{d,rec}|/|N_d|$, the solid one to the estimator at the amplitude level $E_d^X = E_d^A = |A_d - A'_d|/|A_d|$.

precision, via the r.h.s. of Eq. 1, and, finally, $N_{m,rec}$ is the numerator function reconstructed by means of a multi-precision fitting procedure.

In Figs. 1-3, I collect the results obtained by using 3000 random, uniformly distributed Phase Space points. In Fig. 1 I plot the distributions of the ratios P_d/E_d^N and P_d/E_d^A. In the latter case (solid histogram), most of points fall within 2 orders of magnitude, indicating that E_d^A is expected to accurately estimate the *true* numerical precision. This is not the case for the estimator based on the $N = N$ test. The long right tail in the P_d/E_d^N distribution (dashed histogram) shows that there are points for which the accuracy is badly overestimated by E_d^N. In Fig. 2, I plot the distributions of P_d (dashed histogram) and P_m (solid histogram). It can be seen that keeping the computation of $N(q)$ in double precision, while performing the fit in multi-precision improves the accuracy. Nevertheless, points exist for which A_m is still not accurate enough, even if the solid plot stops order of magnitudes before the dashed one. In Fig. 3, I show the tails of the P_d distribution, imposing four different cuts in the value of the estimators E_d^A and E_m^A. In the solid histograms, when $E_d^A > E_{\rm lim}$, a *rescue* of the point is

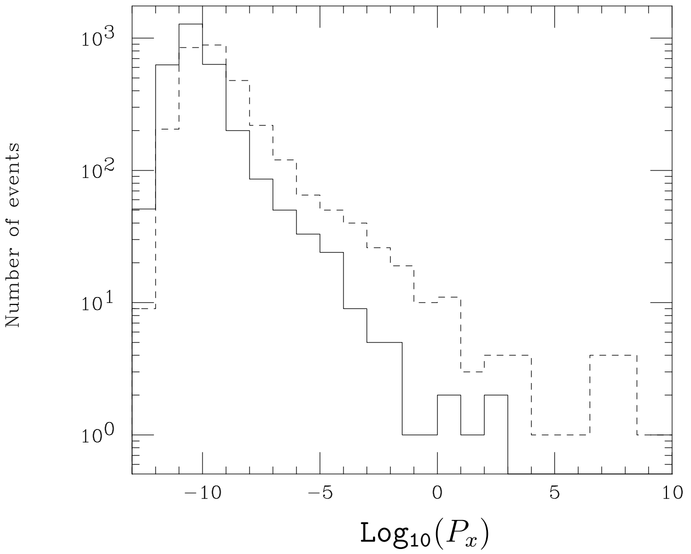

Figure 2: Distribution of the true precision variables $P_x = |A_x - A_e|/|A_e|$ (see text). The dashed histograms refers to the double precision result ($P_x = P_d$), the solid histogram to the case with fitting procedure carried out in multi-precision, but numerator function computed in double precision ($P_x = P_m$).

performed by re-fitting the 1-loop coefficients in multi-precision, while keeping the computation of the numerator function in double precision, and, if also $E^A_m > E_{\rm lim}$, the event is discarded. In the dashed histograms, the same procedure is applied, but using, as estimators, E^N_d and E^N_m. Again, the right tails of the dashed histograms show that E^N_d and E^N_m are not good estimators of the numerical accuracy, while the absence of points above 10^{-1} in the case of all the solid plots, indicates that E^A_d and E^A_m are able to select the *bad* inaccurate points quite efficiently. For reader's reference I summarize, in table 1, a statistics of the number of points computed in multi-precision and discarded in each of the 4 cases. As a conclusion, the rescue procedure is able to recover most of them.

As a final check on the goodness of the estimator at the amplitude level, I present, in table 2, the quantity $\max[\mathtt{Log_{10}}(P_d)] - \mathtt{Log_{10}}(E_{lim})$ as a function of E_{lim}, when using the precision estimator E^A_d. This variable measures the difference between the worst detected point, in an analysis like that one represented by the solid histograms of Fig. 3, and the chosen threshold value E_{lim} for E^A_d. It can be seen that, for values of E_{lim} between 10^{-2} and 10^{-6},

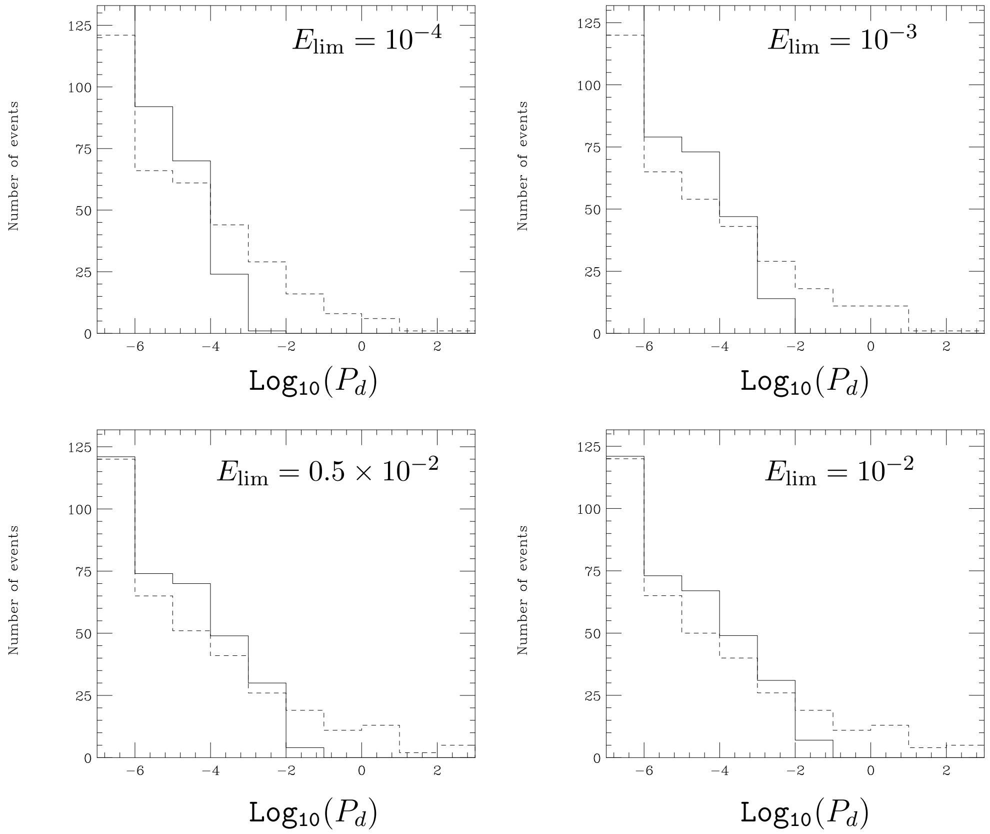

Figure 3: The tails of the distributions of the true precision variable P_d, with an additional constraint on the value of the precision estimators E_d^A, E_m^A, E_d^N and $E_m^N = |N_d - N_{m,rec}|/|N_d|$ (see text). In the solid histograms, when $E_d^A > E_{\rm lim}$, a *rescue* of the point is performed by re-fitting the 1-loop coefficients in multi-precision (with numerator functions kept in double precision) and, if also $E_m^A > E_{\rm lim}$, the event is discarded. In the dashed histograms, the same procedure is applied, but using the estimators E_d^N and E_m^N instead.

E_{lim}	N_{mp}	N_{dis}
10^{-4}	90	14
10^{-3}	62	8
$.5 \times 10^{-2}$	44	6
10^{-2}	40	6

Table 1: The number of points computed in multi-precision (N_{mp}) thanks to the *rescue* procedure and the number of points discarded (N_{dis}) as a function of the threshold value E_{lim}. The numbers refers to the solid histograms of Fig. 3, over a total number of 3000 events.

E_d^A overestimates the accuracy at most by 1.1 decimals and that the points where the overestimate is by almost 2 decimals lie in the safe region of very small values of E_{lim}, from which one argues that E_d^A is able to detect badly instable Phase Space points in a reliable way.

5. Conclusions

I introduced a novel method to test the numerical accuracy of the NLO results produced by modern OPP/Generalized Unitarity techniques. The key ingredient is a re-computation of the 1-loop coefficients based on the properties of the OPP equation under a global shift of all the masses. This re-computation can be performed by using the function previously reconstructed during the determination of the cut-constructible part of the amplitude, therefore at a moderate CPU time cost. As a by-product, I also introduced a faster determination of one of the coefficients contributing to the rational piece of the amplitude. I proved, with numerical tests, the reliability of the procedure and I proposed a convenient solution to *rescue* most of the detected numerically inaccurate points in a way that allows the computation of the integrand to remain in double precision.

6. Acknowledgments

Work supported by the European Community under contract MRTN-CT-2006-035505 and by the Spanish MEC under project FPA2008-02984.

E_{lim}	$\max[\mathtt{Log_{10}}(P_d)] - \mathtt{Log_{10}}(E_{lim})$
10^{-2}	0.80
5×10^{-3}	1.1
10^{-3}	0.74
5×10^{-4}	0.57
10^{-4}	1.1
5×10^{-5}	0.98
10^{-5}	1.0
5×10^{-6}	1.1
10^{-6}	1.0
10^{-8}	1.7
10^{-10}	1.8

Table 2: The variable $\max[\mathtt{Log_{10}}(P_d)] - \mathtt{Log_{10}}(E_{lim})$ as a function of E_{lim}, when using the precision estimator E_d^A.

Appendix A. An alternative determination of $d^{(2m-4)}$

The last coefficient $d^{(2m-4)}$ in the expansion of Eq. 12 contributes to R_1 through Eq. 16. In this appendix, I present a novel technique to determine it from the other, known, coefficients of the OPP expansion.

The starting points are Eqs. 11-14. The l.h.s. of Eq. 11 does not depend on $\tilde{q}^2$, so that one can equate to zero the coefficients of all the powers of $\tilde{q}^2$ appearing in the r.h.s. From the highest power, $\tilde{q}^{(2m-2)}$, one obtains

$$\sum_{i_0<i_1}^{m-1} b^{(2)}(i_0 i_1; q) + \sum_{i_0}^{m-1} a(i_0; q) = 0\,, \tag{A.1}$$

while the next to highest power, $\tilde{q}^{(2m-4)}$, gives Eq. 27. Notice that the r.h.s. of Eq. 27 can be computed at arbitrary values of q, allowing extra numerical checks.

References

[1] G. Ossola, C. G. Papadopoulos and R. Pittau, Nucl. Phys. B **763** (2007) 147 [arXiv:hep-ph/0609007].

[2] G. Ossola, C. G. Papadopoulos and R. Pittau, JHEP **0805** (2008) 004 [arXiv:0802.1876 [hep-ph]].

[3] Z. Bern, L. J. Dixon, D. C. Dunbar and D. A. Kosower, Nucl. Phys. B **425** (1994) 217 [arXiv:hep-ph/9403226].

[4] Z. Bern, L. J. Dixon, D. C. Dunbar and D. A. Kosower, Nucl. Phys. B **435**, 59 (1995) [arXiv:hep-ph/9409265].

[5] Z. Bern and A. G. Morgan, Nucl. Phys. B **467**, 479 (1996) [arXiv:hep-ph/9511336].

[6] Z. Bern, L. J. Dixon and D. A. Kosower, Nucl. Phys. B **513**, 3 (1998) [arXiv:hep-ph/9708239].

[7] R. Britto, F. Cachazo and B. Feng, Nucl. Phys. B **725**, 275 (2005) [arXiv:hep-th/0412103].

[8] D. Forde, Phys. Rev. D **75**, 125019 (2007) [arXiv:0704.1835 [hep-ph]].

[9] R. K. Ellis, W. T. Giele and Z. Kunszt, JHEP **0803** (2008) 003 [arXiv:0708.2398 [hep-ph]].

[10] R. K. Ellis, W. T. Giele, Z. Kunszt and K. Melnikov, Nucl. Phys. B **822** (2009) 270 [arXiv:0806.3467 [hep-ph]].

[11] G. Ossola, C. G. Papadopoulos and R. Pittau, JHEP **0803** (2008) 042 [arXiv:0711.3596 [hep-ph]].

[12] A. van Hameren, C. G. Papadopoulos and R. Pittau, JHEP **0909** (2009) 106 [arXiv:0903.4665 [hep-ph]].

[13] W. T. Giele and G. Zanderighi, JHEP **0806** (2008) 038 [arXiv:0805.2152 [hep-ph]].

[14] C. F. Berger *et al.*, Phys. Rev. D **78** (2008) 036003 [arXiv:0803.4180 [hep-ph]].

[15] A. Lazopoulos, arXiv:0812.2998 [hep-ph].

[16] P. Mastrolia, G. Ossola, T. Reiter and F. Tramontano, arXiv:1006.0710 [Unknown].

[17] G. Bevilacqua, M. Czakon, C. G. Papadopoulos, R. Pittau and M. Worek, JHEP **0909** (2009) 109 [arXiv:0907.4723 [hep-ph]].

[18] G. Bevilacqua, M. Czakon, C. G. Papadopoulos and M. Worek, Phys. Rev. Lett. **104** (2010) 162002 [arXiv:1002.4009 [Unknown]].

[19] R. K. Ellis, W. T. Giele, Z. Kunszt, K. Melnikov and G. Zanderighi, JHEP **0901** (2009) 012 [arXiv:0810.2762 [hep-ph]].

[20] R. K. Ellis, K. Melnikov and G. Zanderighi, *JHEP* **0904** (2009) 077.

[21] R. Keith Ellis, K. Melnikov and G. Zanderighi, Phys. Rev. D **80** (2009) 094002 [arXiv:0906.1445 [hep-ph]].

[22] C. F. Berger *et al.*, Phys. Rev. Lett. **102** (2009) 222001 [arXiv:0902.2760 [hep-ph]].

[23] C. F. Berger *et al.*, Phys. Rev. D **80** (2009) 074036 [arXiv:0907.1984 [hep-ph]].

[24] G. Passarino and M. J. G. Veltman, Nucl. Phys. B **160** (1979) 151.

[25] A. Denner and S. Dittmaier, Nucl. Phys. B **734** (2006) 62 [arXiv:hep-ph/0509141].

[26] R. K. Ellis, W. T. Giele and G. Zanderighi, Phys. Rev. D **73** (2006) 014027 [arXiv:hep-ph/0508308].

[27] R. K. Ellis and G. Zanderighi, JHEP **0802** (2008) 002 [arXiv:0712.1851 [hep-ph]].

[28] P. Draggiotis, M. V. Garzelli, C. G. Papadopoulos and R. Pittau, JHEP **0904** (2009) 072 [arXiv:0903.0356 [hep-ph]].

[29] M. V. Garzelli, I. Malamos and R. Pittau, JHEP **1001** (2010) 040 [arXiv:0910.3130 [Unknown]].

[30] D. H. Bailey; ARPREC (C++/Fortran-90 arbitrary precision package) http://crd.lbl.gov/∼dhbailey/mpdist/.
See also D. H. Bailey, "A Fortran-90 Based Multiprecision System," ACM Transactions on Mathematical Software, vol. 21, no. 4 (Dec 1995), pg. 379-387.