Numerical calculation of high-order QED contributions to the electron anomalous magnetic moment

Sergey Volkov*

Skobeltsyn Institute of Nuclear Physics of Lomonosov Moscow State University (SINP MSU), 1(2), Leninskie gory, GSP-1, Moscow 119991, Russia and Dzhelepov Laboratory of Nuclear Problems, JINR, Joliot-Curie 6, 141980 Dubna, Moscow region, Russia

(Received 16 August 2018; published 31 October 2018)

This paper describes a method of numerically evaluating high-order QED contributions to the electron anomalous magnetic moment. The method is based on the subtraction of infrared and ultraviolet divergences in Feynman parametric space before integration and on nonadaptive Monte Carlo integration that is founded on Hepp sectors. A realization of the method on the graphics accelerator NVidia Tesla K80 is described. A method of removing round-off errors that emerge due to numerical subtraction of divergences without losing calculation speed is presented. The results of applying the method to all 2-loop, 3-loop, and 4-loop QED Feynman graphs without lepton loops are presented. A detailed comparison of the 2-loop and 3-loop results with known analytical ones is given in the paper. A comparison of the contributions of six gauge-invariant 4-loop graph classes with known analytical values is presented. Moreover, the contributions of 78 sets of 4-loop graphs for comparison with the direct subtraction on the mass shell are presented. Also, the contributions of the 5-loop and 6-loop ladder graphs are given, as well as a comparison of these results with known analytical ones. The behavior of the generated Monte Carlo samples is described in detail, and a method of the error estimation is presented. Detailed information about the graphics processor performance on these computations and about the Monte Carlo convergence is given in the paper.

DOI: 10.1103/PhysRevD.98.076018

I. INTRODUCTION

The electron anomalous magnetic moment (AMM) is known with a very high accuracy. In Ref. [1], the value

$$a_e = 0.00115965218073(28)$$

was obtained. So, an extremely high precision is required also from theoretical predictions.

The most precise prediction of the electron's AMM at the present time uses the following representation:

$$a_e = a_e(\mathrm{QED}) + a_e(\mathrm{hadronic}) + a_e(\mathrm{electroweak}),$$

$$a_e(\mathrm{QED}) = \sum_{n\geq 1}\left(\frac{\alpha}{\pi}\right)^n a_e^{2n},$$

$$a_e^{2n} = A_1^{(2n)} + A_2^{(2n)}(m_e/m_\mu) + A_2^{(2n)}(m_e/m_\tau) + A_3^{(2n)}(m_e/m_\mu, m_e/m_\tau),$$

where m_e, m_μ, and m_τ are the masses of the electron, muon, and tau lepton, respectively. Different terms of this expression were calculated by different groups of researchers. Some of them have independent calculations, but others were calculated only by one scientific group. The best theoretical value [2],

$$a_e = 0.001159652182032(13)(12)(720), \qquad (1)$$

was obtained by using the fine-structure constant $\alpha^{-1} = 137.035998995(85)$ that had been obtained by using methods independent from a_e (see Ref. [2]). Here, the first, second, and third uncertainties come from $A_1^{(10)}$, $a_e(\mathrm{hadronic}) + a_e(\mathrm{electroweak})$, and the fine-structure constant,[1] respectively. The values

$$A_1^{(2)} = 0.5,$$
$$A_1^{(4)} = -0.328478965579193\ldots,$$
$$A_1^{(6)} = 1.181241456\ldots,$$
$$A_1^{(8)} = -1.9122457649\ldots$$

*volkoff_sergey@mail.ru, sergey.volkov.1811@gmail.com

[1] So, the calculated coefficients are used for improving the accuracy of α.

are known from the analytical and semianalytical results in Refs. [3,4], Refs. [5,6], Ref. [7], and Ref. [8], respectively.[2] The value

$$A_1^{(10)} = 6.675(192)$$

was presented in Ref. [2]. At the present time, there are no independent calculations of $A_1^{(10)}$. However, $A_1^{(8)}$ was evaluated independently[3] in Refs. [25,27,28] (and for the graphs without lepton loops in Ref. [29]). We must take into account the fact that the contributions of some individual graphs turn out to be several times greater than the total contribution in absolute value.[4] Therefore, an error in one graph evaluation can cause the final result to be entirely wrong. So, the problem of evaluating $A_1^{(2n)}$ is still relevant.

The QED contributions to a_e that are the most uncertain and difficult to evaluate correspond to Feynman graphs without lepton loops. We consider an evaluation of these contributions in this paper and denote the n-loop part of it by $A_1^{(2n)}$[no lepton loops].

This paper is the continuation of a series of papers [29,30] with increasing precision, number of independent loops in graphs, and refinement of the consideration.

We use the subtraction procedure that was introduced in Ref. [30] for removing both infrared and ultraviolet divergences. It is briefly described in Sec. II of this paper. This procedure eliminates IR and UV divergences in each AMM Feynman graph point by point, before integration, in the spirit of Refs. [2,31–41], etc. This property is substantial for many-loop calculations when reducing the computer time is of critical importance. Let us note that $A_1^{(2n)}$ is free from infrared divergences, since they are removed by the on-shell renormalization, as well as the ultraviolet ones (see a more detailed explanation in Ref. [30]). However, the subtractive on-shell renormalization cannot eliminate IR divergences in Feynman parametric space before integration as well as it does for UV divergences.[5] The structure of IR and UV divergences in individual Feynman graphs is quite complicated.[6] Therefore, a special procedure is required for removing both UV and IR divergences. Let us recapitulate the advantages of the developed subtraction procedure:

(1) It is fully automated for any n.
(2) It is comparatively easy for realization on computers.
(3) It can be represented as a forestlike formula. This formula differs from the classical forest formula [38,39,42] only in the choice of linear operators and in the way of combining them.
(4) The contribution of each Feynman graph to $A_1^{(2n)}$ can be represented as a single Feynman parametric integral. The value of $A_1^{(2n)}$ is the sum of these contributions.
(5) Feynman parameters can be used directly, without any additional tricks.

See a detailed description in Ref. [30]. The subtraction procedure was checked independently by F. Rappl using Monte Carlo integration based on Markov chains [27]. An additional advantage of the procedure is described below and in Sec. IV H.

After the subtraction is applied, the problem is reduced to numerical integration of functions of many variables. The number of variables can be quite large[7]; this fact compels us to use Monte Carlo methods. In most cases the precision of Monte Carlo integration behaves asymptotically as $C/\sqrt{N}$, where N is the number of samples. Thus, for reaching a high precision in practical time, it is very important to decrease the constant C as much as possible. Unfortunately, the behavior of Feynman parametric integrands that appear in $A_1^{(2n)}$ computation often leads to slow Monte Carlo convergence. An integration method with a relatively good constant C was introduced in Ref. [29]. The method is based on importance sampling with probability density functions that are constructed for each Feynman graph individually. The construction is based on Hepp sectors [37] and uses functions of the form that was first used by E. Speer [43] with some modifications. The modification is based on the concept of I-closure that was introduced in Ref. [29]. The method from Ref. [29] demonstrated better convergence than the universal Monte Carlo routines. A refined version of the construction is described in Sec. III of this paper. This refinement reduces the uncertainty of $A_1^{(8)}$[no lepton loops] by about 15% when the number of samples is fixed.

When we have to deal with unbounded functions or with functions having sharp peaks, the standard Monte Carlo error estimation approach has a tendency to underestimate the inaccuracy. A method of preventing underestimation was described in Ref. [29]. However, some tests show that in many cases a more accurate consideration of peaks is required. An improved method of error estimation that uses a specificity of the considered integrands is presented in Sec. IV F. Detailed information about sample behavior for the 5-loop and 6-loop ladder graphs is provided. Also, information about the dependence of the results on the

[2]The value for $A_1^{(6)}$ was a product of the efforts of many scientists. See, e.g., Refs. [9–24].

[3]However, by 2016, most parts of $A_1^{(8)}$ had been calculated by only one scientific group [25]. The first numerical estimations for $A_1^{(8)}$ were presented in Ref. [26].

[4]This turns out to be the case regardless of the divergence subtraction method used.

[5]Moreover, it can generate additional IR divergences; see a more detailed explanation in Ref. [30].

[6]See notes in Ref. [29].

[7]For example, for five loops we have 13 variables; see Ref. [29] and Sec. IV A.

number of samples is given for $A_1^{(8)}$[no lepton loops] and for the 5-loop and 6-loop ladder graph contributions.

The numerical subtraction of divergences leads to a situation in which small numbers (in absolute value) are obtained as the difference of astronomically big numbers. This generates round-off errors that significantly affect the result.[8] To control these errors, we need to use additional techniques that substantially slow down the computation speed. In Ref. [30], all integrand evaluations were first performed with two different precisions,[9] and when a difference in the results was noticeable, the calculation was repeated with increased precision. This approach requires twice as much computer time than the direct calculation. Also, an emergence of bias is possible in this case. All calculations that are described in Ref. [29] use interval arithmetic.[10] Interval arithmetic is reliable, but it slows down the computation many times: e.g., the multiplication of two intervals requires eight number multiplications with correct rounding, three minimums, and three maximums. To eliminate this slowdown, a special modification of interval arithmetic was developed. This technique gave a significant improvement in computation speed without the loss of reliability. In many cases, this method works faster than the approach with two precisions.[11] A specificity of the construction of the integrands is used for reaching such performance. The description of this technique is contained in Sec. IV C.

The rapid development of specialized computing devices that solve some tasks many times faster than ordinary computers makes it possible to use them for scientific calculations. All Monte Carlo integrations that are described in this paper were performed on one[12] graphics processor of the NVidia Tesla K80. Graphics processors (GPUs) are very useful for Monte Carlo integration. However, specific programming is required to use these devices effectively. Sections IV A, IV D, and IV I contain some information about the realization of the described integration method on GPUs.

The developed method and realization were applied for computing $A_1^{(2n)}$[no lepton loops], $n = 2, 3, 4$. Also, the contributions of the 5-loop and 6-loop ladders were evaluated for testing purposes. The results are presented in Sec. IV A. A comparison with known analytical results is provided in Table XVII.

High-order calculations in quantum field theory require performing some operations with enormous amounts of information. For example, the total integrand code size[13] for $A_1^{(8)}$[no lepton loops] is 2.5 GB. There are too many places where a mistake can emerge. However, the total independent check requires a lot of resources. So, it is very important to have the possibility of checking the results by parts using another method. Section IV H demonstrates that the developed method provides such a possibility. The total number of 269 Feynman graphs for $A_1^{(8)}$[no lepton loops] is divided into 78 sets, and the contribution of each set must coincide with the contribution that is obtained by direct subtraction on the mass shell in Feynman gauge. The contribution of each set is provided in Sec. IV H. Also, analogous information is given for the 2-loop and 3-loop cases; the comparison in this paper is as good as the one in Ref. [30]. The contributions of six gauge-invariant classes of 4-loop graphs without lepton loops are presented in Sec. IV H and compared with the semianalytical ones from Ref. [8]. Knowing the values of the contributions of gauge-invariant classes gives us the ability to check some hypotheses from quantum field theory.[14] Section IV G contains detailed information about the contributions of individual Feynman graphs, including the influence of round-off errors and information about Monte Carlo error estimation. A summary of the results and technical information about GPU performance and code sizes is presented in Sec. IV I.

II. SUBTRACTION OF DIVERGENCES

We will work in the system of units in which $\hbar = c = 1$, the factors of 4π appear in the fine-structure constant $\alpha = e^2/(4\pi)$, the tensor $g_{\mu\nu}$ is defined by

$$g_{\mu\nu} = g^{\mu\nu} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix},$$

and the Dirac γ matrices satisfy the condition $\gamma^\mu\gamma^\nu + \gamma^\nu\gamma^\mu = 2g^{\mu\nu}$.

We will use Feynman graphs with the propagators

$$\frac{i(\hat{p} + m)}{p^2 - m^2 + i\varepsilon} \tag{2}$$

for electron lines and

$$\frac{-g_{\mu\nu}}{p^2 + i\varepsilon} \tag{3}$$

for photon lines. We restrict our attention to graphs without lepton loops. However, the developed subtraction procedure works for graphs with lepton loops as well [30].

[8] Moreover, these errors can convert a finite result to an infinite one.

[9] In 64-bit and 80-bit precisions that are supported on processors that are compatible with the Intel x86 family.

[10] See Sec. IV B.

[11] See Table XVII.

[12] The NVidia Tesla K80 has two GPUs.

[13] See Table XVII.

[14] See Sec. V.

The number $\omega(G) = 4 - N_\gamma - \frac{3}{2}N_e$ is called the *ultraviolet degree of divergence* of the graph G. Here, N_γ is the number of external photon lines of G, and N_e is the number of external electron lines of G.

A subgraph[15] G' of the graph G is called UV divergent if $\omega(G') \geq 0$. There are the following types of UV-divergent subgraphs in QED Feynman graphs without lepton loops: *electron self-energy subgraphs* ($N_e = 2,\ N_\gamma = 0$) and *vertexlike* subgraphs ($N_e = 2,\ N_\gamma = 1$).

Two subgraphs are said to overlap if they are not contained one inside the other and their sets of lines have a nonempty intersection.

A set of subgraphs of a graph is called a *forest* if any two elements of the set do not overlap.

For a vertexlike graph G, by $\mathfrak{F}[G]$ we denote the set of all forests F consisting of UV-divergent subgraphs of G and satisfying the condition $G \in F$. By $\mathfrak{T}[G]$, we denote the set of all vertexlike subgraphs G' of G such that G' contains the vertex that is incident[16] to the external photon line of G.[17]

We will use the following linear operators that are applied to the Feynman amplitudes of UV-divergent subgraphs:

(1) A is the projector of AMM. This operator is applied to the Feynman amplitudes of vertexlike subgraphs. See the definition in Refs. [29,30].

(2) The definition of the operator U depends on the type of UV-divergent subgraph to which the operator is applied:

a. If $\Sigma(p)$ is the Feynman amplitude that corresponds to an electron self-energy subgraph

$$\Sigma(p) = u(p^2) + v(p^2)\hat{p}, \tag{4}$$

then, by definition,[18]

$$U\Sigma(p) = u(m^2) + v(m^2)\hat{p}. \tag{5}$$

b. If $\Gamma_\mu(p,q)$ is the Feynman amplitude[19] corresponding to a vertexlike subgraph

$$\Gamma_\mu(p,0) = a(p^2)\gamma_\mu + b(p^2)p_\mu + c(p^2)\hat{p}p_\mu + d(p^2)(\hat{p}\gamma_\mu - \gamma_\mu\hat{p}), \tag{6}$$

then, by definition,

$$U\Gamma_\mu = a(m^2)\gamma_\mu. \tag{7}$$

The operator U can be used for extracting the UV-divergent part of the amplitude without touching the IR-divergent part. For example, for the 1-loop amplitude in Eq. (6), all UV divergences are contained in $a(p^2)\gamma_\mu$, but all IR divergences are in $b(p^2)p_\mu + c(p^2)\hat{p}p_\mu$. For the 1-loop amplitude in Eq. (4), IR divergences appear after on-shell differentiating that is needed in the standard renormalization, but not for defining U. See a detailed description in terms of Feynman parameters in Ref. [30]. It is important that U preserve the Ward identity. This fact is used for proving that the subtraction procedure is equivalent to the on-shell renormalization and for calculating the contributions of graph classes; see Ref. [30] and Sec. IV H. It is also important for removing IR divergences that Eq. (5) extract the self-mass completely; see Discussion in Ref. [30].

(3) L is the operator that is used in the standard subtractive on-shell renormalization of vertexlike subgraphs. If $\Gamma_\mu(p,q)$ is the Feynman amplitude that corresponds to a vertexlike subgraph, Eq. (6) is satisfied, and then, by definition,

$$L\Gamma_\mu = [a(m^2) + mb(m^2) + m^2c(m^2)]\gamma_\mu. \tag{8}$$

Let f_G be the unrenormalized Feynman amplitude that corresponds to a vertexlike graph G. Let us write the symbolic definition

$$\tilde{f}_G = \mathcal{R}_G^{\text{new}} f_G, \tag{9}$$

where

$$\mathcal{R}_G^{\text{new}} = \sum_{\substack{F=\{G_1,\ldots,G_n\}\in\mathfrak{F}[G] \\ G'\in\mathfrak{T}[G]\cap F}} (-1)^{n-1} M_{G_1}^{G'} M_{G_2}^{G'} \ldots M_{G_n}^{G'}, \tag{10}$$

$$M_{G''}^{G'} = \begin{cases} A_{G'}, & \text{if } G' = G'', \\ U_{G''}, & \text{if } G'' \notin \mathfrak{F}[G], \text{or } G'' \subsetneq G', \\ L_{G''}, & \text{if } G'' \in \mathfrak{F}[G], G' \subsetneq G'', G'' \neq G, \\ (L_{G''} - U_{G''}), & \text{if } G'' = G, G' \neq G. \end{cases} \tag{11}$$

In this notation, the subscript of an operator symbol denotes the subgraph to which this operator is applied.

The coefficient before γ_μ in $\tilde{f}_G$ is the contribution of G to a_e. See the examples of applying the procedure in Refs. [29,30]. The operators $L_{G''}$ and $(L_{G''} - U_{G''})$ are used for removing the IR divergences that are connected with subgraphs in the sense of Ref. [44] and the corresponding UV ones. Note that the operator $(L_{G''} - U_{G''})$ is required in Eq. (11) for removing UV divergences,[20] and in

[15] In this paper, we take into account only subgraphs that are strongly connected and contain all lines that join the vertices of the given subgraph.

[16] We say that a line l and a vertex v are *incident* if v is one of the end points of l.

[17] In particular, $G \in \mathfrak{T}[G]$.

[18] Note that it differs from the standard on-shell renormalization.

[19] These rules are applied for individual Feynman graphs and even for fixed values of Feynman parameters. So we cannot neglect $\ldots(\hat{p}\gamma_\mu - \gamma_\mu\hat{p})$ terms, and we cannot use the Ward-Takahashi identity or other simplifications.

[20] See Ref. [30], Appendix C.

order to make this subtraction equivalent to the on-shell renormalization,[21] it cannot be replaced by $L_{G''}$.

III. PROBABILITY DENSITY FUNCTIONS FOR MONTE CARLO INTEGRATION

We use Feynman parameters for calculations. Thus, to obtain the contribution of a graph G, we need to calculate the integral

$$\int_{z_1,\ldots,z_n>0} I(z_1,\ldots,z_n)\delta(z_1+\cdots+z_n-1)dz_1\ldots dz_n,$$

where the function I is constructed by using the known rules [30].

We use the Monte Carlo approach based on importance sampling: we generate randomly N samples $\underline{z}_1,\ldots,\underline{z}_N$, where $\underline{z}_j=(z_{j,1},\ldots,z_{j,n})$, using some probability density function $g(\underline{z})$ and approximating the integral value by

$$\frac{1}{N}\sum_{j=1}^{N}\frac{I(\underline{z})}{g(\underline{z})}. \tag{12}$$

The density g is fixed for a fixed graph G. The speed of Monte Carlo convergence depends on the selection of g. A construction of G that gives a good convergence is described below.

We will use Hepp sectors [37] and functions of the form that was first used by E. Speer [43] with some modifications. All the space $\mathbb{R}^n$ is split[22] into *sectors*. Each sector corresponds to a permutation $(j_1,\ldots,j_n)$ of $\{1,2,\ldots,n\}$ and is defined by

$$S_{j_1,\ldots,j_n}=\{(z_1,\ldots,z_n)\in\mathbb{R}:z_{j_1}\geq z_{j_2}\geq\ldots\geq z_{j_n}\}.$$

We define the function $g_0(z_1,\ldots,z_n)$ on $S_{j_1,\ldots,j_n}$ by the following relation:

$$g_0(z_1,\ldots,z_n)=\frac{\prod_{l=2}^{n}(z_{j_l}/z_{j_{l-1}})^{\mathrm{Deg}(\{j_l,j_{l+1},\ldots,j_n\})}}{z_1z_2\ldots z_n}, \tag{13}$$

where $\mathrm{Deg}(s)>0$ is defined for each set s of internal lines[23] of G except the empty set and the set of all internal lines of G. The probability density function is defined by

$$g(z_1,\ldots,z_n)=\frac{g_0(z_1,\ldots,z_n)}{\int_{z_1,\ldots,z_n>0}g_0(z_1,\ldots,z_n)\delta(z_1+\cdots+z_n-1)dz_1\ldots dz_n}. \tag{14}$$

A fast random samples generation algorithm for a given $\mathrm{Deg}(s)$ is described in Ref. [29].

Let us describe the procedure of obtaining $\mathrm{Deg}(s)$. The following auxiliary definitions repeat the ones from Ref. [29]. By definition, we set

$$\omega(s)=2N_L(s)+|e(s)|/2-|s|,$$

where $|x|$ is the cardinality of a set x, $e(s)$ is the set of all electron lines in s, and $N_L(s)$ is the number of independent loops in s. If s is the set of all internal lines of a subgraph of G, then $\omega(s)$ coincides with the ultraviolet degree of divergence of this subgraph that is defined above.

The problem of constructing a good $g(\underline{z})$ is very close to the problem of obtaining a simple and close enough upper bound for $|I(\underline{z})|$ and proving the integral finiteness; see Ref. [29]. Feynman parametric expressions for the integrands (without subtraction terms) can be represented as fractions with denominators that vanish on the boundary of the integration area, if we are on the mass shell [30]. If we consider the numerators only, we can use the ultraviolet degrees of divergence themselves; see Ref. [43]. If we take into account the denominators too, the degrees must be increased, which is performed by I-closures that are defined below. In addition to vanishing denominators, the divergence subtraction complicates the problem. The construction described below is based on both theoretical considerations[24] and numerical experiments.

By $\mathrm{IClos}(s)$ we denote the set $s\cup s'$, where s' is the set of all internal photon lines l in G such that s contains the electron path in G connecting the ends of l. The set $\mathrm{IClos}(s)$ is called the *I-closure* of the set s.

By definition, we set

$$\omega'(s)=\omega(\mathrm{IClos}(s)).$$

A graph G'' belonging to a forest $F\in\mathfrak{F}[G]$ is called a *child* of a graph $G'\in F$ in F if $G''\subsetneq G'$ and there is no $G'''\in F$ such that $G'''\subsetneq G'$, $G''\subsetneq G'''$.

If $F\in\mathfrak{F}[G]$ and $G'\in F$, then by G'/F we denote the graph that is obtained from G' by shrinking all children of G' in F to points.

We will also use the symbols ω, ω' for graphs G' that are constructed from G by some operations like those described above,[25] and for sets s that are subsets of the set of internal lines of the whole graph G. We will denote these by $\omega_{G'}(s)$ and $\omega'_{G'}(s)$, respectively. This means that we apply the operations ω and ω' in the graph G' to the set s' that is the intersection of s and the set of all internal lines of G'.

Electron self-energy subgraphs and lines joining them form chains $l_1G_1l_2G_2\ldots l_rG_rl_{r+1}$, where l_j's are electron

[21] See Sec. IV H and Ref. [30], Appendix B.

[22] Let us remark that the components have intersections on their boundaries. However, this is inessential for integration.

[23] Note that the sets can be not connected.

[24] Some of the ideas underlying the concept of I-closure and this procedure of obtaining $\mathrm{Deg}(s)$ will be described in future papers. (These ideas are quite complicated and are not completely substantiated mathematically at this moment.)

[25] See the corresponding examples in Ref. [29].

lines of G, and G_j's are electron self-energy subgraphs of G. Maximal (with respect to inclusion) subsets $\{l_1, l_2, \ldots, l_{r+1}\}$ corresponding to such chains are called *SE chains*. The set of all SE chains of G is denoted by $\mathrm{SE}[G]$.

Suppose a graph G' is constructed from G by operations like those described above; by definition, we set

$$\omega^*_{G'}(s) = \omega'_{G'}(s) + \frac{1}{2} \sum_{\substack{s' \in \mathrm{SE}[G] \\ s' \subseteq s,\ s' \text{ in } G'}} (|s'| - 1).$$

(It is important that here we consider the SE chains of the whole graph G.)

By $\mathfrak{F}_{\max}[G]$, we denote the set of all maximal forests belonging to $\mathfrak{F}[G]$ (with respect to inclusion).

Let $C_{\mathrm{bigF}} > 0$, $C_{\mathrm{bigZ}} > 0$, C_{add}, C_{subI}, C_{subSE}, and C_{subO} be constants. By definition, we set

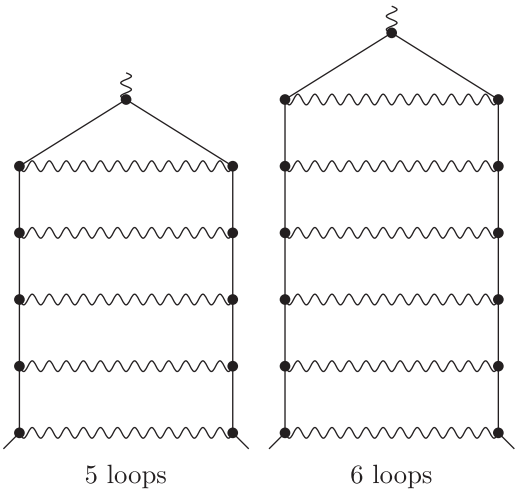


FIG. 1. 5-loop and 6-loop ladder graphs.

$$\mathrm{Deg}(s) = \begin{cases} C_{\mathrm{bigZ}} + \frac{(C_{\mathrm{bigF}} - C_{\mathrm{bigZ}}) N_L(s)}{N_L(G)}, & \text{if } s \text{ contains all electron lines of } G, \\ C_{\mathrm{add}} + \min_{F \in \mathfrak{F}_{\max}[G]} \sum_{G' \in F} \max(0, -\omega^*_{G'/F}(s) - \mathrm{Sub}[G']), & \\ \text{otherwise}, & \end{cases}$$

where

$$\mathrm{Sub}[G'] = \begin{cases} C_{\mathrm{subI}}, & \text{if } G' \in \mathfrak{F}[G], \\ C_{\mathrm{subSE}}, & \text{if } G' \text{is a self-energy subgraph}, \\ C_{\mathrm{subO}} & \text{in the other cases}. \end{cases}$$

This formula for $\mathrm{Deg}(s)$ differs from the one that was defined in Ref. [29] and gives better Monte Carlo convergence, if appropriate values for constants are taken. For good Monte Carlo convergence, we can use the values

$$\begin{aligned} C_{\mathrm{bigZ}} &= 0.256, & C_{\mathrm{bigF}} &= 0.839, \\ C_{\mathrm{add}} &= 0.786, & C_{\mathrm{subI}} &= 0.2, \\ C_{\mathrm{subSE}} &= 0, & C_{\mathrm{subO}} &= 0.2. \end{aligned} \tag{15}$$

These values were obtained by a series of numerical experiments on 4-loop Feynman graphs. See the examples for the considered combinatorial constructions in Ref. [29].

IV. REALIZATION AND NUMERICAL RESULTS

A. Overview

The computation on one GPU of an NVidia Tesla K80 that was leased from Google Cloud[26] showed the following results (1σ limits[27]):

$$\begin{aligned} A_1^{(4)}[\text{no lepton loops}] &= -0.3441651(34), \\ A_1^{(6)}[\text{no lepton loops}] &= 0.90485(10), \\ A_1^{(8)}[\text{no lepton loops}] &= -2.181(10), \end{aligned}$$

where the corresponding computation times are 21 h 37 min, 5 d 8 h, and 7 d. The obtained contributions of the 5-loop and 6-loop ladder graphs from Fig. 1 are 11.6530(58) and 34.31(20), respectively. The corresponding computation times are 4 h 38 min and 8 h 24 min. All obtained results are in good agreement with the known analytical and semianalytical ones; see Table XVII. See also the detailed results in Secs. IV G, IV H, and IV I.

We reduce the number of integration variables by one using the fact that each integrand $I(z_1, \ldots, z_n)$ depends linearly on z_a when $z_a + z_b$ is fixed, where a and b are the electron lines that are incident to the vertex that is incident to the external photon line; see Ref. [29].[28] In contrast to Refs. [29,30], we use a nonadaptive[29] Monte Carlo algorithm. The absence of adaptivity simplifies a realization on the GPU and allows us to undertake an analysis of the Monte Carlo samples behavior; see Sec. IV F.

The D programming language [46] was used for the generator of the integrands code. The integrands and the

[26] Using the free trial.

[27] See Sec. IV F.

[28] And also Ref. [45].

[29] Except for the selection of the parameters (15) and an intergraph adaptivity: the numbers of Monte Carlo samples for each Feynman graph are selected to make the convergence maximally fast.

Monte Carlo integrator were written in C++ with CUDA [47]. The integrand code sizes are presented in Table XVII. The pseudorandom generator MRG32k3a from the CURAND library [48] was used for the Monte Carlo integration.

The integrand values are evaluated first using double-precision[30] floating-point operations that are fully supported on the GPU. If the double-precision operations do not give enough accuracy, the calculations are repeated using arbitrary-precision floating-point operations with increasing precision; see the details in Sec. IV D.

All the integrand code is divided into shared libraries that are linked dynamically with the integrator. Each Feynman graph and type of arithmetic corresponds to one or several shared libraries. Each of these shared libraries contains CUDA kernels[31] and functions for calling them. To reduce the compilation time[32] without losing the computation performance, the size of the integrand CUDA kernels is set at approximately 5000 operations. Also, to reduce the compilation time, each arbitrary-precision shared library contains no more than 10 CUDA kernels.

The memory speed is a weak spot of GPU computing. So, the integrand GPU code is organized in such a way that the most of the operations are performed with the GPU register memory: we are trying to minimize the number of the used variables, often to the detriment of the arithmetic optimization.

To use the GPU parallel computing effectively, we divide the Monte Carlo samples for one Feynman graph into portions. Each portion contains from 10^6 to 10^8 samples. First, we generate the samples of a given portion and calculate the corresponding integrand values in the fastest precision. After that, the samples requiring an increased precision are collected and calculated. Each CUDA kernel is launched on a GPU in 19 968 parallel threads.[33] To reduce the impact of the latency of CUDA kernel calling, each thread performs approximately 15 samples sequentially in a loop.

B. Interval arithmetic

Interval arithmetic is an easy and reliable way for controlling round-off errors. In this way all calculations are performed with intervals, not with numbers. Arithmetic operations on intervals are defined in such a way that each exact intermediate value x is guaranteed to be in the corresponding interval $[x^-;x^+]$. One can use the following definitions:

$$
\begin{aligned}
[x^-;x^+]+[y^-;y^+] &= [(x^-+y^-)^{\mathrm{down}};(x^++y^+)^{\mathrm{up}}],\\
[x^-;x^+]-[y^-;y^+] &= [(x^--y^+)^{\mathrm{down}};(x^+-y^-)^{\mathrm{up}}],\\
[x^-;x^+]\cdot[y^-;y^+] &= [\min((x^-y^-)^{\mathrm{down}},(x^-y^+)^{\mathrm{down}},\\
&\qquad (x^+y^-)^{\mathrm{down}},(x^+y^+)^{\mathrm{down}});\\
&\qquad \max((x^-y^-)^{\mathrm{up}},(x^-y^+)^{\mathrm{up}},\\
&\qquad (x^+y^-)^{\mathrm{up}},(x^+y^+)^{\mathrm{up}})],\\
1/[x^-;x^+] &= [\min((1/x^-)^{\mathrm{down}},(1/x^+)^{\mathrm{down}});\\
&\qquad \max((1/x^-)^{\mathrm{up}},(1/x^+)^{\mathrm{up}})],
\end{aligned}
$$

where $(*)^{\mathrm{up}}$ and $(*)^{\mathrm{down}}$ mean the operation $(*)$ with rounding up (to $+\infty$) or down (to $-\infty$). Most modern GPUs[34] support specifying the rounding mode for arithmetic operations and working with infinities for handling overflows. Addition, subtraction, and multiplication can be realized directly by using the formulas proposed above.[35] However, for division, it is required to perform additional operations for handling division by zero and overflows. This does not slow down the computation, because the number of divisions in the integrand constructions is very small.

C. Elimination of interval arithmetic

Direct interval arithmetic is a very slow thing. However, there are many ways of increasing speed by weakening the distinctness of the intervals.

We will use the following specificity of the integrands construction. It is known [30][36] how to construct the integrand for a given graph G from the building blocks $V^{G'}$, $Q^{G'}_{a,j}$, $B^{G'}_{ab}$, and $S^{G'}$, where G' is a graph that can be obtained from a subgraph of G by shrinking some subgraphs to points; a, b are internal electron lines of G'; $j=1,2$; $V^{G'}$ is defined through a sum over 1-trees of G'; $Q^{G'}_{a,j}$ is defined through a sum over 1-trees[37] passing a; $B^{G'}_{ab}$ is defined through a sum over trees with a cycle passing a, b; and $S^{G'}$ is defined through a sum over 2-trees. See the full definitions in Ref. [30]. The construction rules described in Ref. [30] allow us to observe that for a high number of independent loops in G, the most part of the integrand computation is the calculation of polynomials with the variables $Q^{G'}_{a,j}/V^{G'}$ and $B^{G'}_{ab}/V^{G'}$.

Suppose we want to calculate a polynomial of the intervals $[x_1^-;x_1^+],\ldots,[x_n^-;x_n^+]$ that is constructed as a

[30]Double precision: 64 bit.

[31]A CUDA kernel is a function in a program that is executed many times in parallel on GPU and is called from the CPU part; see Ref. [47].

[32]GPU device code is compiled very slowly, and the compilation time increases rapidly with the size of functions.

[33]104 blocks of 192 threads.

[34]As well as CPUs.

[35]Also, these formulas will work correctly with "not a number" entries (NANs) despite the fact that the NVidia realization of min and max ignores NANs in the lists of arguments.

[36]See also Refs. [40,41,45].

[37]More precisely, Ref. [30] has a definition of $\hat{Q}^{G'}_a$; the values $Q^{G'}_{a,j}$ can be defined by $\hat{Q}^{G'}_a = Q^{G'}_{a,1}\hat{p}_1 + Q^{G'}_{a,2}\hat{p}_2$ in terms of Ref. [30].

sequence of additions, subtractions, and multiplications. The main ideas of the interval arithmetic elimination are as follows:

(1) We can calculate the center of the resulting interval in the direct double-precision arithmetic using the same polynomial applied to the centers of $[x_j^-; x_j^+]$.
(2) The radius of the resulting interval can be estimated as a function of x_j^-, x_j^+ that is much more simple than the source polynomial.

We will use the following inequality about the machine double-precision arithmetic[38]:

$$|x - x^{\mathrm{rnd}}| \le 2^{-52}|x| + 2^{-1022},$$

where x^{rnd} corresponds to the machine representation of x rounded in any direction.

Let x_j be the exact values corresponding to the intervals $[x_j^-; x_j^+]$, $j = 1, \ldots, n$. By $x_{n+1}, \ldots, x_l$ we denote the exact intermediate values that are obtained sequentially when we calculate the value of the needed polynomial. To each $j = 1, \ldots, n$ we assign a *type* t_j: $t_j = 0$ if x_j is $Q_{a,k}^{G'}/V^{G'}$, $t_j = 1$ if x_j is $B_{ab}^{G'}/V^{G'}$. (We divide all source values into two groups in such a way because $|Q_{a,k}^{G'}/V^{G'}| \le 1$, but $B_{ab}^{G'}/V^{G'}$ are unbounded.[39]) Let us define the numbers x_j^{appr}, M_j, ε_j, $j = 1, \ldots, l$, satisfying the following conditions for all j:

(1) $|x_j^{\mathrm{appr}} - x_j| \le \varepsilon_j$.
(2) $|x_j^{\mathrm{appr}}| \le M_j$.

We define them by using the following rules:

(1) $x_j^{\mathrm{appr}} = ((x_j^- + x_j^+)/2)^{\mathrm{rnd}}$, $j = 1, \ldots, n$. (Thus, x_j^{appr} are the centers of the corresponding intervals; the machine double-precision arithmetic guarantees that we always have $x_j^- \le x_j^{\mathrm{appr}} \le x_j^+$ if an overflow does not occur.)
(2) M_j's are defined for $j = 1, \ldots, n$ by

$$M_j = \max_{t_k = t_j} |x_k^{\mathrm{appr}}|.$$

(3) ε_j's are defined for $j = 1, \ldots, n$ by

$$\varepsilon_j = \varepsilon = \max_{1 \le k \le n} \max\left((x_k^{\mathrm{appr}} - x_k^+)^{\mathrm{up}}, (x_k^+ - x_k^{\mathrm{appr}})^{\mathrm{up}}\right).$$

(4) If x_j is obtained as $x_k * x_r$, where $*$ is addition, subtraction or multiplication, $j = n+1, \ldots, l$, then $x_j^{\mathrm{appr}} = (x_k^{\mathrm{appr}} * x_r^{\mathrm{appr}})^{\mathrm{rnd}}$. (Thus, x_j^{appr} are obtained by the direct double-precision arithmetic without specifying the rounding mode.[40])
(5) Analogously, (M_j, ε_j) is defined by

$$(M_j, \varepsilon_j) = ((M_k + M_r)(1 + 2^{-52}) + 2^{-1022}, \varepsilon_k + \varepsilon_r + 2^{-52}(M_k + M_r) + 2^{-1022})$$

for addition and subtraction, and by

$$(M_j, \varepsilon_j) = (M_k M_r(1 + 2^{-52}) + 2^{-1022}, \varepsilon_k\varepsilon_r + \varepsilon_k M_r + \varepsilon_r M_k + 2^{-52} M_k M_r + 2^{-1022})$$

for multiplication.

It is easy to see that for the final l, the value ε_l can be expressed as a polynomial $P(M_{t=0}, M_{t=1}, \varepsilon)$ with positive coefficients in only three variables, where

$$M_{t=a} = \max_{t_k = a} |x_k^{\mathrm{appr}}|.$$

Thus, the value of ε_l can be obtained directly using the coefficients of this polynomial without calculating the intermediate values M_k, ε_k.

However, the polynomial

$$P(M_{t=0}, M_{t=1}, \varepsilon) = \sum_{u,v,w} C_{u,v,w} (M_{t=0})^u (M_{t=1})^v \varepsilon^w$$

can still have many coefficients and therefore can require a lot of arithmetic operations for computation. We estimate P by another expression in the following way: Let us split P into four parts P_0, P_1, P_2, P_3 by the following rules:

$$P_0 : C_{u,v,w} < 2^{-100}, \quad P_1 : 2^{-100} \le C_{u,v,w} < 0.5,$$
$$P_2 : C_{u,v,w} \ge 0.5, \quad w \le 1, \quad P_3 : C_{u,v,w} \ge 0.5, \quad w \ge 2.$$

Thus, $P = P_0 + P_1 + P_2 + P_3$. By definition, we set

$$u_j^- = \min_{C_{u,v,w}^j > 0} u, \qquad u_j^+ = \max_{C_{u,v,w}^j > 0} u,$$

where

$$P_j(M_{t=0}, M_{t=1}, \varepsilon) = \sum_{u,v,w} C_{u,v,w}^j (M_{t=0})^u (M_{t=1})^v \varepsilon^w, \quad j = 0, 1, 2, 3.$$

Let us define v_j^-, v_j^+, w_j^-, w_j^+ in an analogous way. We set

[38]The last term corresponds to the case when a very small number is converted into zero after rounding.

[39]Generally speaking, we can divide them in any way into any number of pieces. This splitting is selected as a compromise between precision and speed.

[40]In some tests, specifying a rounding mode for addition or multiplication slows down the performance of these operations on an NVidia Tesla K80 by a factor of 7. However, in the considered calculations this was not experienced; see Table XVII.

$$P'_j(M_{t=0}, M_{t=1}, \varepsilon) = \left(\sum_{u,v,w} C^j_{u,v,w}\right) \cdot \max\left((M_{t=0})^{u_j^+}, (M_{t=0})^{u_j^-}\right) \cdot \max\left((M_{t=1})^{v_j^+}, (M_{t=1})^{v_j^-}\right) \cdot \max\left(\varepsilon^{w_j^+}, \varepsilon^{w_j^-}\right).$$

It is obvious that $P'_j \geq P_j$. So, we can use $P' = P'_0 + P'_1 + P'_2 + P'_3$ as a radius of the final interval, if it is calculated by machine arithmetic operations with rounding up.[41] P' is much simpler for calculation than P. Thus, an interval for the final value may be[42]

$$[(x_l^{\text{appr}} - (P')^{\text{up}})^{\text{down}}; (x_l^{\text{appr}} + (P')^{\text{up}})^{\text{up}}].$$

We split P into four polynomials in a way guided by the following considerations:

(1) P_3 contains most of the coefficients' sum; however, its contribution in P'_3 will be compensated by the multiplier ε^2 (when ε is near zero).
(2) P_2 has a large sum of coefficients too; however, it is much less than P_3 has; this sum will be compensated by the multiplier ε in P'_2.
(3) P_1 has a small sum of coefficients; however, in some cases P'_1 can be noticeable; thus, we separate P_1 from P_0 to minimize the contribution of the $\max \cdot \max \cdot \max$ part in the definition of P'_1.
(4) The contribution of the coefficients of P_0 is always small.

D. Algorithm of obtaining accurate integrand values

We obtain the value[43] $I(\underline{z})/g(\underline{z})$ from Eq. (12) first by the eliminated interval arithmetic from Sec. IV C. If the obtained interval $[y^-; y^+]$ does not satisfy the condition $y^+ - y^- \leq \sigma/4$, where σ is the current error estimation[44] for the obtained integral value, we repeat the calculation in the direct double-precision interval arithmetic. If it is not enough, we reiterate this calculation in the interval arithmetic based on floating-point numbers with a 128-bit mantissa and with a 256-bit mantissa (if needed). If the 256-bit-mantissa precision is not enough, we suppose that the value equals 0.

The arithmetic with a 128-bit mantissa is realized on the GPU in such a way that all operations are performed with the GPU register memory. The arithmetic with a 256-bit mantissa works with the global GPU memory. The usage of the register memory improves the performance about tenfold.[45]

We also use a routine for the prevention of the occasional emerging of very large values that is analogous to the one described in Ref. [29], but adapted for GPU parallel computing.

E. Modified probability density functions

The situation in which $g(\underline{z})$ from Eq. (12) is very small; is theoretically possible, but the smallness if $|I(\underline{z})|$ does not correspond to it. An emergence of such situations can make the Monte Carlo convergence worse. For patching in these situations, we use the probability density functions

$$g(\underline{z}) = C_1 g_1(\underline{z}) + C_2 g_2(\underline{z}) + C_3 g_3(\underline{z}) + C_4 g_4(\underline{z})$$

instead of Eqs. (13) and (14). Here g_1 is defined by Eqs. (13) and (14),

$$g_2(z_1, \ldots, z_n) = \frac{\prod_{l=2}^{n} [\text{Deg}(\{j_l, j_{l+1}, \ldots, j_n\})(z_{j_l}/z_{j_{l-1}})^{\text{Deg}(\{j_l, j_{l+1}, \ldots, j_n\})}]}{n! z_1 z_2 \ldots z_n}$$

when the definitions from Sec. III are used, and g_3 is defined by Eqs. (13) and (14), but with the same $\text{Deg}(s) = D$, $g_4(\underline{z}) = (n-1)!$ (the uniform distribution).

To generate a random sample with the distribution $g(\underline{z})$, we should perform the following two steps:

(1) Generate randomly $j = 1, 2, 3, 4$, where the probability of selecting j is C_j.
(2) Generate a sample with the distribution $g_j(\underline{z})$.

The generation with the distribition $g_2(\underline{z})$ is the same as for distributions defined by Eqs. (13) and (14), but at the stage

[41] The coefficients $C_{u,v,w}$ and their sum must be calculated with rounding up too. However, this calculation is performed at the stage of codegeneration.

[42] Overflows, infinities, and NANs do not require an additional consideration at all stages of the calculation.

[43] We cannot use double precision directly for the probability density $g(\underline{z})$, because its value sometimes goes beyond the range of double-precision values. This situation often occurs in the 6-loop case. We use the representation $x \cdot 2^j$ instead, where double precision is used for $0.5 \leq x < 1$, and the number j is a 32-bit integer.

[44] In the beginning of the integral computation, we calculate between 10^5 and 10^7 points in the direct double-precision interval arithmetic, taking the nearest to zero points for each interval.

[45] However, Table XVII shows a gap that is much more than a factor of 10. The reason is that there are very few points requiring 256-bit mantissas, so we cannot use GPU parallelism effectively.

TABLE I. Probability distributions for 5-loop and 6-loop ladders.

Parameter	5-loop ladder	6-loop ladder
Maxlog	23	28
n_0	11	2
n_1	64	8
n_2	393	45
n_3	2 300	174
n_4	11 891	785
n_5	51 840	2 898
n_6	204 817	9 374
n_7	688 060	25759
n_8	1 885 211	62 363
n_9	4 300 121	135 343
n_{10}	8 615 210	267 630
n_{11}	15 701 395	490 720
n_{12}	26 582 404	849 862
n_{13}	42 456 874	1 394 740
n_{14}	64 590 501	2 198 221
n_{15}	94 011 212	3 331 999
n_{16}	13 131 4678	4 892 615
n_{17}	176 228 467	6 965 326
n_{18}	228 021 742	9 626 392
n_{19}	285 614 048	12 965 533

of sector generation we must take sectors with the same probabilities; see Ref. [29]. All computations are performed with the following values for the constants:

$$D = 0.75, \quad C_2 = 0.03, \quad C_3 = 0.035,$$
$$C_4 = 0.035, \quad C_1 = 1 - C_2 - C_3 - C_4.$$

F. Monte Carlo error estimation

Let $\underline{z}_1, \ldots, \underline{z}_N$ be random samples; the formula (12) is used for Monte Carlo integration. By definition, we set $y_j = I(\underline{z})/g(\underline{z})$. The conventional error estimation approach is based on the following formula for the standard deviation:

$$(\sigma_\downarrow)^2 = \frac{\sum_{j=1}^N y_j^2}{N^2} - \frac{(\sum_{j=1}^N y_j)^2}{N^3}.$$

However, this formula has a tendency to underestimate the real standard deviation. Let us consider the 5-loop and 6-loop ladder examples. By definition, we set

$$\text{maxlog} = \max_j \lfloor \log_2 |y_j| + 0.5 \rfloor.$$

Let n_k be the quantity of samples j such that

$$2^{\text{maxlog}-k-0.5} \le |y_j| < 2^{\text{maxlog}-k+0.5}; \tag{16}$$

maxlog and n_j for the 5-loop and 6-loop ladders are presented in Table I. n_k is an approximation for Np_k, where p_k is the probability that a sample is in the interval (16).

We can see that the real standard deviation is highly dependent on the behavior of p_j for $j < 0$. For example, if $p_{j+1}/p_j < 4$ for all $j < j_0$, then the standard deviation is infinite.[46]

We will use the improved estimation[47]

$$(\sigma_\uparrow)^2 = (\sigma_\downarrow)^2 + \triangle_{\text{uncert}} + \triangle_{\text{peak}},$$

where[48]

$$\triangle_{\text{uncert}} = 4 \cdot \max_{k=0}^{19} 4^{\text{maxlog}-k} \sqrt{n_k}$$

is the contribution of the uncertainty of n_k, and $\triangle_{\text{peak}}$ is the contribution of the predicted behavior of p_j for $j < 0$ that is described below.[49]

The idea is to approximate n_j by a geometric progression, taking into account that the n_j's are known with an uncertainty of about $C\sqrt{n_j}$ and that p_{j+1}/p_j changes with j.

We set

$$h_j = \begin{cases} \log_2 n_j, & \text{if } n_j > 0, \\ -2, & \text{if } n_j = 0, \end{cases}$$
$$h_j^\pm = \log_2 \max\left(\frac{1}{8}, n_j + \frac{1}{2} \pm \sqrt{n_j + \frac{1}{4}}\right).$$

Here h_j is an approximated value of $\log_2(Np_j)$, $[h_j^-; h_j^+]$ is an interval for this value that is obtained by taking into account that n_j is known with uncertainty.[50]

We will estimate the absolute value of a difference between neighbors $\log_2(p_{j+1}/p_j)$ by the value d, where

$$d = \max_{0 \le j < k \le 18} \frac{d_{jk}}{k - j},$$

where d_{jk} is the distance from 0 to the interval $[d_{jk}^-; d_{jk}^+]$,

[46]Table I demonstrates that for the 6-loop ladder, such a situation is quite possible.

[47]When we calculate deviation probabilities based on the standard deviation, we use a presupposition based on the central limit theorem that the distribution of $\sum_{j=1}^N y_j/N$ is close to the Gauss normal distribution. However, it is difficult to estimate the difference between the real distribution and the normal one. For example, the Berry-Esseen inequality uses the third central moment of random variables that is infinite if $p_{j+1}/p_j < 8$ for all $j < j_0$. (Table I shows that this situation is quite possible for both 5-loop and 6-loop ladders.)

[48]The definitions of $\sigma_\downarrow$ and $\triangle_{\text{uncert}}$ repeat the ones from Ref. [29].

[49]This procedure is a result of tests on different graph contributions to a_e. It is developed for future calculations of contributions to a_e of higher orders. It should not be treated as a universal procedure that works for all Monte Carlo integrations. However, a large value of $\sigma_\uparrow/\sigma_\downarrow$ indicates that the obtained error estimation is unreliable.

[50]$x = n + \frac{C^2}{2} \pm C\sqrt{n + \frac{C^2}{4}}$ is the solution of the equation $x \mp C\sqrt{x} = n$.

$$d^-_{jk} = (h^-_{k+1} - h^+_k) - (h^+_{j+1} - h^-_j),$$
$$d^+_{jk} = (h^+_{k+1} - h^-_k) - (h^-_{j+1} - h^+_j).$$

For the approximation of the sequence by a progression, we will use other values for the $\log_2(Np_j)$ uncertainty that are obtained by taking into account that errors for lesser j are more critical:

$$u_j = \begin{cases} \frac{1}{2}\left[\log_2\left(n_j + \frac{C_j^2}{2} + C_j\sqrt{n_j + \frac{C_j^2}{4}}\right) - \log_2\left(n_j + \frac{C_j^2}{2} - C_j\sqrt{n_j + \frac{C_j^2}{4}}\right)\right], \\ \text{if } n_j > 0, \\ 3, \text{if } n_j = 0, \end{cases}$$

where

$$C_j = \frac{2}{1 + \frac{2(j+1)}{20}}.$$

For approximating the sequence of logarithms by a linear function $kj + b$, let us introduce the coefficients a^l_j, f^l_j, $2 \le l \le 20$, $0 \le j < l$, for the least squares method[51]:

$$\left(\sum_{j=0}^{l-1} a^l_j x_j; \sum_{j=0}^{l-1} f^l_j x_j\right) = \text{argmin}_{(k;b)} \sum_{j=0}^{l-1} (kj + b - x_j)^2$$

for all l and $x_0, \ldots, x_{l-1}$.

We set

$$k_l = \sum_{j=0}^{l-1} a^l_j h_j - \sqrt{\sum_{j=0}^{l-1} (a^l_j)^2 u_j^2} - d\sum_{j=0}^{l-1} \frac{j(j-1)a^l_j}{2},$$
$$k = \max(k_2, \ldots, k_{20}, h_0 - 1 - u_0).$$

This formula takes into account both the uncertainty of n_j and the shift of p_{j+1}/p_j with j. We take max to prevent excessive overestimation.[52] Also, we set

$$\triangle b = \min_l \left(\sqrt{\sum_{j=0}^{l-1} (f^l_j)^2 u_j^2} + d\sum_{j=0}^{l-1} \frac{j(j-1)f^l_j}{2}\right),$$

$b = \sum_{j=0}^{l-1} f^l_j h_j$, where we take l for which the minimum is achieved. Let us define $\triangle_{\text{peak}}$ by

$$\triangle_{\text{peak}} = 2^{2\cdot\text{maxlog} + b + 0.7\triangle b}\left(\frac{1}{1 - 2^{-w}} - 1\right),$$

where

$$w = \frac{k - \frac{17}{8} + \sqrt{(k - \frac{17}{8})^2 + \frac{1}{16}}}{2} + \frac{1}{8}.$$

The meaning of this definition is that we use the formula for the sum of a geometric progression, taking w instead of $k-2$. w is defined in such a way that $w \sim k-2$ as $k \to +\infty$, and $w \to 1/8$ as $k \to -\infty$.

We use $\sigma_\uparrow$ for all numerical results that are presented in this paper.

TABLE II. Dependence of the estimated error and the difference between the obtained value and the known semianalytical one [8] on the number of Monte Carlo samples N_{total}: $A_1^{(8)}$[no lepton loops]; see a remark about $\sigma_\uparrow$, $\sigma_\downarrow$ calculation in Sec. IV H.

N_{total}	Value	$\sigma_\uparrow$	$\sigma_\downarrow$	Difference	$\sigma_\uparrow/\sigma_\downarrow$
40×10^9	−2.3937	0.2144	0.1168	−0.2168	1.84
10^{11}	−2.2323	0.0710	0.0494	−0.0555	1.44
20×10^{10}	−2.1820	0.0468	0.0345	−0.0051	1.36
50×10^{10}	−2.1851	0.0282	0.0218	−0.0083	1.30
10^{12}	−2.1757	0.0194	0.0154	0.0012	1.26
20×10^{11}	−2.1702	0.0133	0.0109	0.0066	1.23
32×10^{11}	−2.1807	0.0104	0.0086	−0.0038	1.21

TABLE III. Dependence of the estimated error and the difference between the obtained value and the known analytical one [49] on the number of Monte Carlo samples N_{total}: 5-loop ladder.

N_{total}	Value	$\sigma_\uparrow$	$\sigma_\downarrow$	Difference	$\sigma_\uparrow/\sigma_\downarrow$
59×10^5	12.0682	0.8202	0.3288	0.4090	2.49
12×10^7	11.6120	0.1349	0.0720	−0.0472	1.87
24×10^7	11.6934	0.0800	0.0525	0.0342	1.52
60×10^7	11.6798	0.0665	0.0379	0.0206	1.76
10^9	11.6678	0.0427	0.0270	0.0086	1.58
20×10^8	11.6474	0.0277	0.0192	−0.0118	1.44
50×10^8	11.6448	0.0150	0.0120	−0.0144	1.25
10^{10}	11.6509	0.0111	0.0086	−0.0083	1.29
20×10^9	11.6541	0.0073	0.0061	−0.0051	1.19
29×10^9	11.6530	0.0058	0.0050	−0.0062	1.16

[51]The explicit formulas are $a^l_j = \frac{12j - 6(l-1)}{l(l^2-1)}$, $f^l_j = \frac{2(2l-1) - 6j}{l(l+1)}$.

[52]The last argument of max is needed to process the situation when n_0 is quite large: in this case, the absence of n_{-1} is very informative.

TABLE IV. Dependence of the estimated error and the difference between the obtained value and the known analytical one [49] on the number of Monte Carlo samples N_{total}: 6-loop ladder.

N_{total}	Value	$\sigma_\uparrow$	$\sigma_\downarrow$	Difference	$\sigma_\uparrow/\sigma_\downarrow$
15×10^6	34.3209	7.1538	2.0690	−0.0461	3.46
65×10^7	35.4566	1.1201	0.4659	1.0896	2.40
97×10^7	35.0500	0.7556	0.3566	0.6829	2.12
12×10^8	35.0187	0.6808	0.3201	0.6517	2.13
22×10^8	34.5855	0.4217	0.2276	0.2185	1.85
41×10^8	34.3967	0.3020	0.1675	0.0297	1.80
70×10^8	34.3651	0.2320	0.1337	−0.0019	1.74
10^{10}	34.3062	0.1974	0.1137	−0.0608	1.74

Tables II, III, and IV contain the dependence of the error estimations and the real errors on the number of samples N_{total} for $A_1^{(8)}$[no lepton loops], 5-loop, and 6-loop ladders, respectively.

G. Contributions of individual Feynman graphs

The contributions of 2-loop and 3-loop Feynman graphs to $A_1^{(4)}$ and $A_1^{(6)}$ are presented in Tables V and VI. The corresponding Feynman graphs are given in Figs. 3 and 4. Each individual contribution in this paper is given for a Feynman graph without arrow directions on electron lines and includes the contributions of the corresponding graphs with all directions (that are the same). The 4-loop graphs

TABLE V. Contributions of individual Feynman graphs from Fig. 3 to $A_1^{(4)}$.

Number	Graph	Value	N_{total}	$N_{\text{EIA}}^{\text{fail}}$	$N_{\text{IA}}^{\text{fail}}$	N_{128}^{fail}	$\triangle_{\text{EIA}}^{\text{fail}}$	$\triangle_{\text{IA}}^{\text{fail}}$	$\triangle_{128}^{\text{fail}}$	$\sigma_\uparrow/\sigma_\downarrow$
1	2; 1–4, 3–5	−0.0640193(19)	94×10^{10}	26×10^8	32×10^4	0	0.003	2×10^{-6}	0	1.04
2	2; 1–5, 3–4	−0.5899758(14)	58×10^{10}	61×10^7	50×10^6	2	−0.0005	-2×10^{-6}	-2×10^{-19}	1.00
3	3; 1–4, 2–5	−0.4676475(17)	90×10^{10}	44×10^7	57479	0	−0.008	-10^{-5}	0	1.05
4	3; 1–5, 2–4	0.7774774(18)	92×10^{10}	34×10^8	17×10^6	0	0.007	0.0002	0	1.00

TABLE VI. Contributions of individual Feynman graphs from Fig. 4 to $A_1^{(6)}$.

Number	Graph	Value	N_{total}	$N_{\text{EIA}}^{\text{fail}}$	$N_{\text{IA}}^{\text{fail}}$	N_{128}^{fail}	$\triangle_{\text{EIA}}^{\text{fail}}$	$\triangle_{\text{IA}}^{\text{fail}}$	$\triangle_{128}^{\text{fail}}$	$\sigma_\uparrow/\sigma_\downarrow$
1	2; 1–4, 3–6, 5–7	−1.679616(20)	29×10^{10}	57×10^8	33×10^4	0	−0.1	-5×10^{-5}	0	1.08
2	2; 1–4, 3–7, 5–6	0.832792(20)	28×10^{10}	40×10^8	26×10^6	0	0.1	0.0009	0	1.10
3	2; 1–5, 3–6, 4–7	0.214875(14)	19×10^{10}	22×10^8	88×10^4	1	0.01	7×10^{-5}	-4×10^{-24}	1.05
4	2; 1–5, 3–7, 4–6	−0.028928(11)	11×10^{10}	30×10^8	10^6	2	−0.004	-3×10^{-5}	-10^{-11}	1.03
5	2; 1–6, 3–4, 5–7	−0.097163(26)	47×10^{10}	16×10^9	13×10^7	10	−0.002	10^{-5}	3×10^{-16}	1.16
6	2; 1–6, 3–5, 4–7	0.144471(12)	14×10^{10}	31×10^8	23×10^4	3	0.06	9×10^{-6}	-2×10^{-36}	1.02
7	2; 1–6, 3–7, 4–5	0.804106(17)	22×10^{10}	26×10^8	18×10^6	1	0.02	−0.0001	-10^{-61}	1.08
8	2; 1–7, 3–4, 5–6	−2.123267(16)	17×10^{10}	34×10^8	48×10^7	4496	−0.02	−0.0002	-10^{-12}	1.00
9	2; 1–7, 3–5, 4–6	2.524749(18)	19×10^{10}	89×10^8	18×10^6	0	0.07	2×10^{-5}	0	1.00
10	2; 1–7, 3–6, 4–5	−0.058729(11)	11×10^{10}	51×10^8	53×10^6	6	0.009	-2×10^{-5}	-10^{-15}	1.00
11	3; 1–4, 2–6, 5–7	5.042278(27)	57×10^{10}	91×10^8	38×10^5	5	0.5	0.0004	8×10^{-23}	1.09
12	3; 1–4, 2–7, 5–6	−3.500634(25)	50×10^{10}	65×10^8	52×10^6	2	−0.4	−0.02	-3×10^{-22}	1.06
13	3; 1–5, 2–6, 4–7	−1.757945(15)	27×10^{10}	97×10^7	34×10^5	10	−0.05	−0.0002	8×10^{-13}	1.10
14	3; 1–5, 2–7, 4–6	0.140129(14)	18×10^{10}	35×10^8	40×10^5	10	0.003	9×10^{-7}	5×10^{-21}	1.06
15	3; 1–6, 2–4, 5–7	−3.257290(27)	48×10^{10}	16×10^9	28×10^6	5	−0.3	−0.004	-10^{-8}	1.00
16	3; 1–6, 2–5, 4–7	−0.334691(14)	23×10^{10}	13×10^8	30×10^5	7	−0.07	−0.0005	10^{-12}	1.13
17	3; 1–6, 2–7, 4–5	0.315388(22)	43×10^{10}	20×10^8	16×10^6	0	−0.003	-5×10^{-8}	0	1.03
18	3; 1–7, 2–4, 5–6	4.513076(27)	43×10^{10}	22×10^9	46×10^7	2909	0.4	0.04	2×10^{-6}	1.00
19	3; 1–7, 2–5, 4–6	0.611112(21)	28×10^{10}	67×10^8	37×10^5	1	0.1	0.0006	-10^{-24}	1.15
20	3; 1–7, 2–6, 4–5	−2.269647(16)	19×10^{10}	27×10^8	39×10^6	0	−0.09	−0.001	0	1.00
21	4; 1–3, 2–6, 5–7	−2.908437(22)	34×10^{10}	81×10^8	50×10^4	0	−0.4	−0.0005	0	1.01
22	4; 1–3, 2–7, 5–6	6.533883(31)	60×10^{10}	20×10^9	16×10^7	9	0.9	0.03	2×10^{-8}	1.01
23	4; 1–7, 2–3, 5–6	−3.204367(20)	24×10^{10}	35×10^8	56×10^7	5040	−0.2	−0.05	-3×10^{-6}	1.00
24	4; 1–5, 2–6, 3–7	−0.0267956(78)	10^{11}	37×10^7	35×10^5	5	−0.02	−0.0007	-6×10^{-12}	1.07
25	4; 1–5, 2–7, 3–6	1.861914(17)	31×10^{10}	19×10^8	14×10^6	43	0.1	−0.0004	-2×10^{-12}	1.08
26	4; 1–6, 2–7, 3–5	−0.945906(11)	14×10^{10}	36×10^8	13×10^5	0	−0.07	−0.0005	0	1.01
27	4; 1–7, 2–5, 3–6	−2.230794(19)	31×10^{10}	36×10^8	11×10^6	24	−0.2	−0.0009	-5×10^{-21}	1.10
28	4; 1–7, 2–6, 3–5	1.790285(19)	28×10^{10}	83×10^8	22×10^6	1	0.1	0.003	-5×10^{-9}	1.01

$$3;\ 1-8,\ 2-7,\ 4-5,\ 6-9.$$

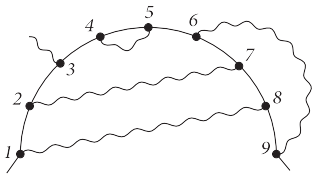


FIG. 2. 4-loop Feynman graph: Example.

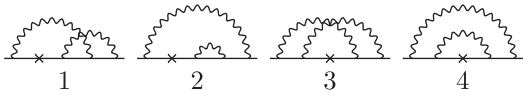


FIG. 3. 2-loop Feynman graphs without lepton loops.

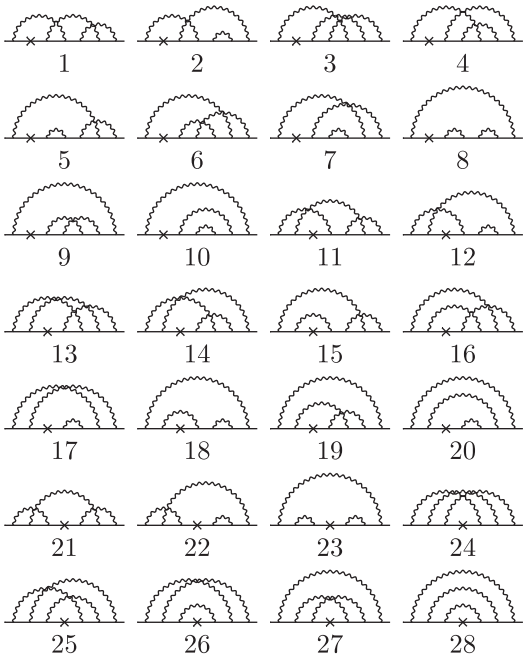


FIG. 4. 3-loop Feynman graphs without lepton loops.

are split into gauge-invariant classes (k, m, m'), where m and m' are the numbers of internal photon lines to the left and to the right from the external photon line (or vice versa), and k is the number of photons with the ends on the opposite sides of the line. We do not provide a picture for 4-loop graphs, but they are encoded in the tables as expressions of the form

$$p;\quad s_1-f_1,\quad s_2-f-2,\quad s_3-f_3,\quad s_4-f_4,$$

where p is the number of vertex that is incident to the external photon line, s_j and f_j are the ends of the jth internal photon line, and the vertices are enumerated from 1 to 9 along the electron path, $s_j < f_j, s_1 < \ldots < s_4$. The graphs are ordered lexicographically, and we guarantee that the code of a graph is the lexicographically minimal one. For example, the code of the graph from Fig. 2 is

$$3;\quad 1\text{–}8,\quad 2\text{–}7,\quad 4\text{–}5,\quad 6\text{–}9.$$

The contributions of the 4-loop graphs are presented in Tables VII, VIII, IX, X, XI, and XII. The numbers of the graphs for which the contribution must coincide with the contribution obtained by direct subtraction on the mass shell in Feynman gauge are marked by a star *; see Sec. IV H.

The fields of the tables have the following meaning:

(1) Value is the obtained value for the contribution with the uncertainty $\sigma_\uparrow$; see Sec. IV F.
(2) $\sigma_\uparrow/\sigma_\downarrow$ is the relation between the improved standard deviation and the conventional one; see Sec. IV F.
(3) N_{total} is the total quantity of Monte Carlo samples.
(4) $N^{\text{fail}}_{\text{EIA}}$ is the quantity of samples for which the eliminated interval arithmetic from Sec. IV C failed.
(5) $\triangle^{\text{fail}}_{\text{EIA}}$ is the contribution of those samples.[53]
(6) $N^{\text{fail}}_{\text{IA}}$ is the quantity of samples for which the direct double-precision interval arithmetic from Sec. IV B failed.
(7) $\triangle^{\text{fail}}_{\text{IA}}$ is the contribution of those samples.
(8) N^{fail}_{128} is the quantity of samples for which the interval arithmetic based on numbers with 128-bit mantissas failed.
(9) $\triangle^{\text{fail}}_{128}$ is the contribution of those samples.[54]

H. Classes of Feynman graphs

The contributions and N_{total} for all classes in this paper are obtained as sums of the corresponding individual values. The values $\sigma_\uparrow$, $\sigma_\downarrow$ for the classes are obtained by

$$\sigma_\uparrow = \sqrt{\sum_j (\sigma_{\uparrow,j})^2}, \qquad \sigma_\downarrow = \sqrt{\sum_j (\sigma_{\downarrow,j})^2},$$

where $\sigma_{\uparrow,j}$ and $\sigma_{\downarrow,j}$ are the corresponding individual values.

The contributions of graph sets to $A_1^{(4)}$, $A_1^{(6)}$, $A_1^{(8)}$ for comparison with the direct subtraction on the mass shell in the Feynman gauge are presented in Tables XIII, XIV, and XV. The 2-loop and 3-loop tables include a comparison with the known analytical results[55] and with the old results from [30][56] Table XV does not include one-element sets; these sets (individual graphs) are marked by a star in the tables containing individual contributions.

[53]Sometimes this contribution can be many times more than the total 4-loop contribution. See, e.g., graph 157 from Table IX. However, the eliminated interval arithmetic significantly improves the computation performance; see Table XVII.

[54]Even these contributions can be noticeable. See, e.g., graph 134 from Table VIII.

[55]The big discrepancy for the sets 14,17 in Table XIV is probably caused by an unstable behavior of the pseudorandom generator MRG32k3a. The generator Philox_4x32_10 [48] seems to work better on this set.

[56]The uncertainties in Ref. [30] correspond to 90% confidential limits (under the assumption that the probability distribution is Gauss normal).

TABLE VII. Contributions of graphs from the gauge-invariant class (1,3,0) to $A_1^{(8)}$.

Number	Graph	Value	N_{total}	$N_{\text{EIA}}^{\text{fail}}$	$N_{\text{IA}}^{\text{fail}}$	N_{128}^{fail}	$\triangle_{\text{EIA}}^{\text{fail}}$	$\triangle_{\text{IA}}^{\text{fail}}$	$\triangle_{128}^{\text{fail}}$	$\sigma_\uparrow/\sigma_\downarrow$
1	2; 1–4, 3–6, 5–8, 7–9	2.19701(73)	15×10^9	92×10^7	27780	0	0.6	0.0001	0	1.56
2	2; 1–4, 3–6, 5–9, 7–8	−3.81327(91)	23×10^9	95×10^7	11×10^5	1	−1	−0.01	2×10^{-45}	1.65
3	2; 1–4, 3–7, 5–8, 6–9	0.55330(31)	49×10^8	23×10^7	31677	1	0.2	0.0001	9×10^{-29}	1.52
4	2; 1–4, 3–7, 5–9, 6–8	1.82177(56)	10^{10}	57×10^7	47919	2	0.5	0.0002	10^{-36}	1.51
5	2; 1–4, 3–8, 5–6, 7–9	−2.43257(93)	23×10^9	13×10^8	29×10^5	1	−0.8	−0.008	10^{-35}	1.72
6	2; 1–4, 3–8, 5–7, 6–9	0.95204(51)	89×10^8	50×10^7	26741	1	0.3	8×10^{-5}	-4×10^{-55}	1.41
7	2; 1–4, 3–8, 5–9, 6–7	−2.19745(69)	15×10^9	57×10^7	54×10^4	0	−0.4	−0.003	0	1.89
8	2; 1–4, 3–9, 5–6, 7–8	2.1481(10)	20×10^9	75×10^7	27×10^6	161	0.8	0.03	-6×10^{-7}	1.65
9	2; 1–4, 3–9, 5–7, 6–8	−2.48196(92)	26×10^9	19×10^8	98×10^4	6	−0.7	−0.004	-9×10^{-40}	1.54
10	2; 1–4, 3–9, 5–8, 6–7	0.98718(84)	19×10^9	12×10^8	33×10^5	3	0.5	0.008	2×10^{-41}	1.61
11	2; 1–5, 3–6, 4–8, 7–9	−1.38009(58)	12×10^9	54×10^7	90636	4	−0.3	−0.0002	-2×10^{-31}	1.63
12	2; 1–5, 3–6, 4–9, 7–8	1.16697(56)	12×10^9	37×10^7	55×10^4	5	0.2	0.002	4×10^{-52}	1.48
13	2; 1–5, 3–7, 4–8, 6–9	0.66741(35)	58×10^8	15×10^7	31×10^4	22	−0.03	−0.001	-3×10^{-14}	1.29
14	2; 1–5, 3–7, 4–9, 6–8	−0.26457(35)	48×10^8	22×10^7	38×10^4	61	−0.06	-10^{-5}	5×10^{-11}	1.25
15	2; 1–5, 3–8, 4–6, 7–9	1.05969(43)	63×10^8	42×10^7	44335	3	0.2	7×10^{-5}	9×10^{-41}	1.31
16	2; 1–5, 3–8, 4–7, 6–9	0.47610(29)	49×10^8	15×10^7	39328	3	0.06	−0.0004	-5×10^{-30}	1.51
17	2; 1–5, 3–8, 4–9, 6–7	0.47497(36)	62×10^8	12×10^7	20×10^4	4	0.2	0.0003	6×10^{-36}	1.21
18	2; 1–5, 3–9, 4–6, 7–8	−1.10746(43)	63×10^8	40×10^7	90×10^4	1	−0.1	−0.0004	7×10^{-35}	1.35
19	2; 1–5, 3–9, 4–7, 6–8	−0.23411(34)	47×10^8	26×10^7	33298	2	−0.09	5×10^{-5}	-3×10^{-35}	1.53
20	2; 1–5, 3–9, 4–8, 6–7	0.13458(26)	37×10^8	12×10^7	16×10^4	0	0.03	0.0001	0	1.08
21	2; 1–6, 3–4, 5–8, 7–9	1.35348(94)	23×10^9	17×10^8	57×10^5	2	0.2	10^{-5}	-10^{-17}	1.45
22	2; 1–6, 3–4, 5–9, 7–8	0.2807(11)	25×10^9	11×10^8	30×10^6	143	−0.1	0.0005	10^{-11}	1.58
23	2; 1–6, 3–5, 4–8, 7–9	3.18477(49)	77×10^8	50×10^7	36801	1	0.5	0.0001	5×10^{-43}	1.14
24	2; 1–6, 3–5, 4–9, 7–8	−2.12704(44)	65×10^8	38×10^7	78×10^4	1	−0.2	−0.003	-5×10^{-42}	1.20
25	2; 1–6, 3–7, 4–8, 5–9	−0.11489(33)	65×10^8	14×10^7	13×10^4	3	0.09	3×10^{-5}	5×10^{-35}	1.38
26	2; 1–6, 3–7, 4–9, 5–8	−0.54446(25)	43×10^8	10^8	83223	4	−0.09	0.0006	-2×10^{-41}	1.34
27	2; 1–6, 3–8, 4–5, 7–9	−4.78772(64)	11×10^9	61×10^7	11×10^5	3	−0.7	−0.002	-2×10^{-36}	1.18
28	2; 1–6, 3–8, 4–7, 5–9	−0.53692(20)	33×10^8	10^8	56191	2	−0.2	−0.0007	4×10^{-12}	1.06
29	2; 1–6, 3–8, 4–9, 5–7	−0.05767(33)	51×10^8	24×10^7	87180	6	−0.04	0.0002	2×10^{-23}	1.33
30	2; 1–6, 3–9, 4–5, 7–8	2.90445(80)	12×10^9	42×10^7	14×10^6	110	0.5	0.004	10^{-11}	1.50
31	2; 1–6, 3–9, 4–7, 5–8	0.57805(26)	41×10^8	18×10^7	50314	1	0.2	0.0003	-2×10^{-62}	1.31
32	2; 1–6, 3–9, 4–8, 5–7	−0.20433(25)	35×10^8	19×10^7	68993	2	−0.05	−0.0002	2×10^{-25}	1.03
33	2; 1–7, 3–4, 5–8, 6–9	−1.38855(31)	47×10^8	34×10^7	10^6	1	−0.2	−0.0001	-2×10^{-65}	1.13
34	2; 1–7, 3–4, 5–9, 6–8	1.11200(60)	10^{10}	85×10^7	27×10^5	3	0.06	-4×10^{-5}	7×10^{-36}	1.27
35	2; 1–7, 3–5, 4–8, 6–9	−1.52611(33)	52×10^8	24×10^7	68690	0	−0.3	−0.0002	0	1.14
36	2; 1–7, 3–5, 4–9, 6–8	−0.12123(28)	38×10^8	25×10^7	38674	0	−0.04	5×10^{-5}	0	1.10
37	2; 1–7, 3–6, 4–8, 5–9	1.10916(23)	39×10^8	10^8	66387	2	0.2	0.0003	-7×10^{-42}	1.29
38	2; 1–7, 3–6, 4–9, 5–8	0.41843(17)	30×10^8	86×10^6	33866	3	−0.02	0.0002	2×10^{-38}	1.04
39	2; 1–7, 3–8, 4–5, 6–9	1.92228(36)	61×10^8	14×10^7	25×10^4	3	0.2	0.002	-3×10^{-54}	1.23
40	2; 1–7, 3–8, 4–6, 5–9	−0.30635(32)	49×10^8	23×10^7	13×10^4	10	−0.04	0.0003	5×10^{-23}	1.30
41	2; 1–7, 3–8, 4–9, 5–6	−0.33355(38)	63×10^8	14×10^7	35×10^4	4	−0.1	−0.0008	-4×10^{-27}	1.22
42	2; 1–7, 3–9, 4–5, 6–8	−1.25068(37)	52×10^8	30×10^7	50×10^4	1	−0.1	-5×10^{-5}	-2×10^{-37}	1.20
43	2; 1–7, 3–9, 4–6, 5–8	−0.24926(28)	37×10^8	19×10^7	44540	4	−0.08	-7×10^{-6}	3×10^{-42}	1.13
44	2; 1–7, 3–9, 4–8, 5–6	0.06345(24)	34×10^8	11×10^7	14×10^4	2	−0.02	−0.0002	10^{-36}	1.11
45	2; 1–8, 3–4, 5–6, 7–9	−0.15958(87)	11×10^9	11×10^8	38×10^6	773	0.07	0.001	2×10^{-11}	1.30
46	2; 1–8, 3–4, 5–7, 6–9	1.53603(54)	88×10^8	70×10^7	25×10^5	0	0.1	-9×10^{-5}	0	1.11
47	2; 1–8, 3–4, 5–9, 6–7	0.95386(50)	67×10^8	33×10^7	82×10^5	31	0.1	0.0008	5×10^{-12}	1.33
48	2; 1–8, 3–5, 4–6, 7–9	−0.08923(76)	13×10^9	14×10^8	19×10^5	1	−0.1	−0.0001	-2×10^{-49}	1.24
49	2; 1–8, 3–5, 4–7, 6–9	−0.25943(28)	38×10^8	22×10^7	19584	1	0.03	-6×10^{-6}	-4×10^{-43}	1.07
50	2; 1–8, 3–5, 4–9, 6–7	0.55223(44)	61×10^8	37×10^7	68×10^4	2	0.3	0.002	10^{-34}	1.12
51	2; 1–8, 3–6, 4–5, 7–9	0.48462(67)	11×10^9	11×10^8	70×10^5	6	0.1	0.0003	3×10^{-20}	1.47
52	2; 1–8, 3–6, 4–7, 5–9	−0.04645(25)	38×10^8	16×10^7	38772	1	−0.03	0.0001	5×10^{-40}	1.42

(Table continued)

TABLE VII. *(Continued)*

Number	Graph	Value	N_{total}	$N_{\text{EIA}}^{\text{fail}}$	$N_{\text{IA}}^{\text{fail}}$	N_{128}^{fail}	$\triangle_{\text{EIA}}^{\text{fail}}$	$\triangle_{\text{IA}}^{\text{fail}}$	$\triangle_{128}^{\text{fail}}$	$\sigma_\uparrow/\sigma_\downarrow$
53	2; 1–8, 3–6, 4–9, 5–7	−0.96878(35)	44×10^8	26×10^7	27×10^4	22	−0.3	−0.002	2×10^{-15}	1.41
54	2; 1–8, 3–7, 4–5, 6–9	−1.28667(34)	51×10^8	15×10^7	20×10^4	3	−0.1	−0.002	2×10^{-48}	1.06
55	2; 1–8, 3–7, 4–6, 5–9	−0.15820(28)	40×10^8	20×10^7	42500	2	−0.06	6×10^{-6}	9×10^{-18}	1.04
56	2; 1–8, 3–7, 4–9, 5–6	−0.47655(29)	39×10^8	13×10^7	23×10^4	1	−0.02	−0.001	8×10^{-48}	1.07
57	2; 1–8, 3–9, 4–5, 6–7	1.52806(52)	69×10^8	21×10^7	72×10^5	45	0.08	−0.01	-4×10^{-7}	1.35
58	2; 1–8, 3–9, 4–6, 5–7	−1.45583(49)	71×10^8	51×10^7	35×10^4	9	−0.07	0.0007	-6×10^{-26}	1.12
59	2; 1–8, 3–9, 4–7, 5–6	−0.08533(42)	61×10^8	47×10^7	10^6	5	−0.02	−0.002	8×10^{-34}	1.25
60	2; 1–9, 3–4, 5–6, 7–8	−3.83273(45)	32×10^8	21×10^7	40×10^6	10177	−0.2	−0.006	-3×10^{-9}	1.00
61	2; 1–9, 3–4, 5–7, 6–8	4.36705(42)	46×10^8	58×10^7	80×10^5	90	0.5	0.001	10^{-12}	1.00
62	2; 1–9, 3–4, 5–8, 6–7	−2.66788(42)	43×10^8	49×10^7	18×10^6	376	−0.2	−0.003	-10^{-11}	1.00
63	2; 1–9, 3–5, 4–6, 7–8	4.36685(43)	46×10^8	58×10^7	76×10^5	87	0.5	0.001	5×10^{-13}	1.01
64	2; 1–9, 3–5, 4–7, 6–8	−3.89486(56)	74×10^8	66×10^7	36×10^4	1	−0.4	-9×10^{-5}	2×10^{-41}	1.04
65	2; 1–9, 3–5, 4–8, 6–7	3.73069(57)	80×10^8	55×10^7	19×10^5	3	0.4	0.006	2×10^{-16}	1.02
66	2; 1–9, 3–6, 4–5, 7–8	−2.66773(43)	43×10^8	48×10^7	17×10^6	402	−0.2	−0.003	-2×10^{-11}	1.00
67	2; 1–9, 3–6, 4–7, 5–8	−1.30095(21)	31×10^8	21×10^7	24×10^4	2	−0.1	−0.002	-2×10^{-51}	1.02
68	2; 1–9, 3–6, 4–8, 5–7	−1.77247(38)	53×10^8	39×10^7	31×10^4	4	−0.2	0.0002	-10^{-45}	1.06
69	2; 1–9, 3–7, 4–5, 6–8	3.73275(57)	82×10^8	56×10^7	17×10^5	7	0.4	0.006	-6×10^{-25}	1.03
70	2; 1–9, 3–7, 4–6, 5–8	−1.77353(38)	51×10^8	38×10^7	28×10^4	4	−0.2	0.0002	-2×10^{-33}	1.05
71	2; 1–9, 3–7, 4–8, 5–6	2.20445(42)	59×10^8	31×10^7	11×10^5	2	0.03	0.002	4×10^{-70}	1.03
72	2; 1–9, 3–8, 4–5, 6–7	0.48560(40)	45×10^8	30×10^7	84×10^5	66	0.06	0.001	4×10^{-7}	1.02
73	2; 1–9, 3–8, 4–6, 5–7	−0.54790(43)	55×10^8	50×10^7	79×10^4	6	−0.1	-10^{-5}	-9×10^{-40}	1.01
74	2; 1–9, 3–8, 4–7, 5–6	−0.57472(34)	45×10^8	48×10^7	26×10^5	5	−0.07	-4×10^{-5}	10^{-15}	1.02

TABLE VIII. Contributions of graphs from the gauge-invariant class (2,2,0) to $A_1^{(8)}$.

Number	Graph	Value	N_{total}	$N_{\text{EIA}}^{\text{fail}}$	$N_{\text{IA}}^{\text{fail}}$	N_{128}^{fail}	$\triangle_{\text{EIA}}^{\text{fail}}$	$\triangle_{\text{IA}}^{\text{fail}}$	$\triangle_{128}^{\text{fail}}$	$\sigma_\uparrow/\sigma_\downarrow$
75	3; 1–4, 2–6, 5–8, 7–9	−10.44260(93)	25×10^9	13×10^8	11×10^4	5	−3	−0.003	-3×10^{-32}	1.29
76	3; 1–4, 2–6, 5–9, 7–8	10.0730(12)	43×10^9	17×10^8	23×10^5	3	3	0.08	2×10^{-6}	1.72
77	3; 1–4, 2–7, 5–8, 6–9	−1.67666(34)	56×10^8	27×10^7	55710	2	−0.6	−0.0003	-10^{-12}	1.32
78	3; 1–4, 2–7, 5–9, 6–8	−5.75797(76)	18×10^9	99×10^7	12×10^4	3	−2	−0.003	-6×10^{-24}	1.31
79	3; 1–4, 2–8, 5–6, 7–9	11.5103(11)	31×10^9	20×10^8	42×10^5	1	5	0.2	4×10^{-24}	1.32
80	3; 1–4, 2–8, 5–7, 6–9	−5.15144(69)	15×10^9	77×10^7	79826	0	−2	−0.003	0	1.12
81	3; 1–4, 2–8, 5–9, 6–7	6.80288(85)	23×10^9	85×10^7	84×10^4	3	2	0.04	-4×10^{-38}	1.63
82	3; 1–4, 2–9, 5–6, 7–8	−10.3320(13)	31×10^9	15×10^8	47×10^6	207	−4	−0.6	-8×10^{-5}	1.45
83	3; 1–4, 2–9, 5–7, 6–8	12.7423(12)	38×10^9	29×10^8	15×10^5	12	5	0.08	3×10^{-22}	1.25
84	3; 1–4, 2–9, 5–8, 6–7	−8.7252(10)	31×10^9	21×10^8	57×10^5	9	−3	−0.2	-10^{-36}	1.36
85	3; 1–5, 2–6, 4–8, 7–9	4.29301(57)	14×10^9	48×10^7	13×10^4	5	0.8	0.001	4×10^{-39}	1.43
86	3; 1–5, 2–6, 4–9, 7–8	−3.37792(58)	15×10^9	29×10^7	46×10^4	6	−0.8	−0.005	-3×10^{-46}	1.38
87*	3; 1–5, 2–7, 4–8, 6–9	0.04665(19)	31×10^8	50×10^6	12×10^4	11	0.08	−0.0002	-4×10^{-15}	1.22
88	3; 1–5, 2–7, 4–9, 6–8	1.37913(29)	49×10^8	16×10^7	81590	3	0.1	0.0004	-5×10^{-20}	1.09
89	3; 1–5, 2–8, 4–6, 7–9	−1.90541(57)	10^{10}	68×10^7	16×10^4	4	−0.08	-2×10^{-5}	-3×10^{-24}	1.42
90*	3; 1–5, 2–8, 4–7, 6–9	0.01638(17)	28×10^8	78×10^6	60829	2	−0.04	-2×10^{-6}	-2×10^{-49}	1.37
91	3; 1–5, 2–8, 4–9, 6–7	−1.82097(33)	55×10^8	94×10^6	39×10^4	41	−0.2	−0.001	8×10^{-8}	1.25
92	3; 1–5, 2–9, 4–6, 7–8	0.81820(59)	11×10^9	71×10^7	14×10^5	1	−0.1	−0.001	-3×10^{-38}	1.44
93	3; 1–5, 2–9, 4–7, 6–8	0.99984(33)	48×10^8	24×10^7	98029	1	0.4	9×10^{-5}	4×10^{-54}	1.24
94	3; 1–5, 2–9, 4–8, 6–7	0.21323(34)	55×10^8	14×10^7	21×10^4	1	0.04	2×10^{-5}	7×10^{-61}	1.39
95	3; 1–6, 2–4, 5–8, 7–9	7.24388(90)	22×10^9	20×10^8	74×10^4	2	2	0.03	9×10^{-41}	1.10
96	3; 1–6, 2–4, 5–9, 7–8	−6.5173(10)	25×10^9	21×10^8	79×10^5	22	−2	−0.2	-2×10^{-6}	1.32
97	3; 1–6, 2–5, 4–8, 7–9	−0.76878(48)	10^{10}	42×10^7	78801	3	0.1	0.001	4×10^{-33}	1.38
98	3; 1–6, 2–5, 4–9, 7–8	0.84511(69)	19×10^9	68×10^7	87×10^4	3	−0.2	−0.006	2×10^{-47}	2.14

(Table continued)

TABLE VIII. *(Continued)*

Number	Graph	Value	N_{total}	$N_{\text{EIA}}^{\text{fail}}$	$N_{\text{IA}}^{\text{fail}}$	N_{128}^{fail}	$\triangle_{\text{EIA}}^{\text{fail}}$	$\triangle_{\text{IA}}^{\text{fail}}$	$\triangle_{128}^{\text{fail}}$	$\sigma_\uparrow/\sigma_\downarrow$
99*	3; 1–6, 2–7, 4–8, 5–9	−0.54587(32)	70×10^8	98×10^6	25×10^4	6	−0.2	0.001	10^{-17}	1.30
100*	3; 1–6, 2–7, 4–9, 5–8	0.21216(20)	36×10^8	83×10^6	81834	2	−0.03	−0.001	-3×10^{-48}	1.32
101	3; 1–6, 2–8, 4–5, 7–9	1.56673(63)	13×10^9	60×10^7	81×10^4	0	−0.02	-8×10^{-5}	0	1.32
102*	3; 1–6, 2–8, 4–7, 5–9	0.80327(22)	34×10^8	10^8	17×10^4	22	0.2	0.0003	7×10^{-16}	1.50
103	3; 1–6, 2–8, 4–9, 5–7	0.57570(31)	58×10^8	19×10^7	91547	5	0.005	0.0004	5×10^{-32}	1.28
104	3; 1–6, 2–9, 4–5, 7–8	−0.26580(66)	12×10^9	37×10^7	10^7	41	0.3	0.003	7×10^{-13}	1.37
105	3; 1–6, 2–9, 4–7, 5–8	−0.52269(26)	40×10^8	18×10^7	10^5	4	−0.1	−0.0003	-2×10^{-28}	1.40
106	3; 1–6, 2–9, 4–8, 5–7	0.25704(41)	68×10^8	36×10^7	17×10^4	5	0.06	-2×10^{-5}	-6×10^{-17}	1.63
107	3; 1–7, 2–4, 5–8, 6–9	2.14029(37)	65×10^8	57×10^7	18×10^4	3	0.6	0.002	4×10^{-44}	1.17
108	3; 1–7, 2–4, 5–9, 6–8	3.31673(81)	20×10^9	17×10^8	67×10^4	5	1	0.02	10^{-34}	1.15
109*	3; 1–7, 2–5, 4–8, 6–9	1.36063(23)	43×10^8	80×10^6	50857	2	0.2	8×10^{-6}	3×10^{-21}	1.33
110	3; 1–7, 2–5, 4–9, 6–8	0.23770(36)	67×10^8	28×10^7	57875	1	0.2	0.0009	-10^{-53}	1.52
111*	3; 1–7, 2–6, 4–8, 5–9	−0.22297(25)	47×10^8	59×10^6	10^5	1	0.02	−0.0003	2×10^{-34}	1.28
112*	3; 1–7, 2–6, 4–9, 5–8	0.44982(20)	37×10^8	70×10^6	61087	4	0.02	−0.0008	5×10^{-40}	1.42
113	3; 1–7, 2–8, 4–5, 6–9	−1.41855(35)	68×10^8	13×10^7	26×10^4	3	−0.03	0.001	10^{-48}	1.39
114	3; 1–7, 2–8, 4–6, 5–9	0.60572(33)	51×10^8	22×10^7	32×10^4	58	−0.02	−0.003	-4×10^{-9}	1.31
115	3; 1–7, 2–8, 4–9, 5–6	−0.79421(38)	73×10^8	12×10^7	36×10^4	9	−0.07	0.0002	-3×10^{-29}	1.73
116	3; 1–7, 2–9, 4–5, 6–8	−0.05379(51)	93×10^8	42×10^7	54×10^4	3	−0.06	-3×10^{-5}	2×10^{-43}	1.26
117	3; 1–7, 2–9, 4–6, 5–8	0.05536(30)	47×10^8	21×10^7	84948	4	−0.1	-6×10^{-6}	-5×10^{-23}	1.24
118	3; 1–7, 2–9, 4–8, 5–6	−0.35767(28)	44×10^8	96×10^6	10^5	2	−0.07	7×10^{-6}	-2×10^{-62}	1.20
119	3; 1–8, 2–4, 5–6, 7–9	−9.3447(11)	24×10^9	29×10^8	20×10^6	316	−4	−0.3	-2×10^{-5}	1.02
120	3; 1–8, 2–4, 5–7, 6–9	3.24250(79)	18×10^9	15×10^8	59×10^4	3	2	0.03	-10^{-34}	1.02
121	3; 1–8, 2–4, 5–9, 6–7	−5.52110(73)	14×10^9	11×10^8	39×10^5	18	−1	−0.1	-9×10^{-6}	1.02
122	3; 1–8, 2–5, 4–6, 7–9	−1.34858(74)	15×10^9	12×10^8	24×10^4	0	−0.8	−0.005	0	1.59
123*	3; 1–8, 2–5, 4–7, 6–9	0.17083(32)	56×10^8	19×10^7	91465	0	0.1	0.0001	0	1.52
124	3; 1–8, 2–5, 4–9, 6–7	−1.91613(37)	65×10^8	19×10^7	21×10^4	2	−0.4	−0.002	-5×10^{-57}	1.23
125	3; 1–8, 2–6, 4–5, 7–9	1.72927(38)	57×10^8	37×10^7	10^6	1	0.5	0.006	6×10^{-20}	1.07
126*	3; 1–8, 2–6, 4–7, 5–9	−0.21815(30)	60×10^8	13×10^7	44124	1	−0.1	-10^{-6}	4×10^{-48}	1.64
127	3; 1–8, 2–6, 4–9, 5–7	−0.10348(33)	50×10^8	21×10^7	25×10^4	25	0.03	0.0009	10^{-15}	1.30
128	3; 1–8, 2–7, 4–5, 6–9	−1.99695(75)	24×10^9	35×10^7	66×10^4	7	−0.4	−0.004	-8×10^{-17}	1.35
129	3; 1–8, 2–7, 4–6, 5–9	0.01814(26)	43×10^8	17×10^7	72758	7	0.03	0.0004	2×10^{-22}	1.19
130	3; 1–8, 2–7, 4–9, 5–6	1.15462(54)	12×10^9	20×10^7	48×10^4	6	0.2	0.003	-9×10^{-28}	1.38
131	3; 1–8, 2–9, 4–5, 6–7	1.26086(63)	13×10^9	29×10^7	71×10^5	29	0.07	0.0005	2×10^{-12}	1.26
132	3; 1–8, 2–9, 4–6, 5–7	−1.83728(67)	14×10^9	89×10^7	47×10^4	8	−0.3	-7×10^{-5}	5×10^{-15}	1.31
133	3; 1–8, 2–9, 4–7, 5–6	0.52838(50)	96×10^8	61×10^7	95×10^4	7	0.002	2×10^{-5}	-3×10^{-38}	1.28
134	3; 1–9, 2–4, 5–6, 7–8	11.8155(12)	18×10^9	22×10^8	85×10^6	12313	4	0.9	0.0005	1.01
135	3; 1–9, 2–4, 5–7, 6–8	−14.1724(13)	34×10^9	40×10^8	12×10^6	107	−5	−0.3	10^{-6}	1.02
136	3; 1–9, 2–4, 5–8, 6–7	9.4205(10)	21×10^9	24×10^8	22×10^6	328	3	0.3	2×10^{-5}	1.05
137	3; 1–9, 2–5, 4–6, 7–8	1.46361(79)	16×10^9	14×10^8	70×10^5	36	0.9	0.03	9×10^{-7}	1.48
138	3; 1–9, 2–5, 4–7, 6–8	−5.30357(87)	21×10^9	10^9	16×10^4	4	−1	−0.002	2×10^{-40}	1.43
139	3; 1–9, 2–5, 4–8, 6–7	1.51767(94)	24×10^9	85×10^7	22×10^5	6	0.6	0.006	-10^{-37}	1.63
140	3; 1–9, 2–6, 4–5, 7–8	−1.68650(46)	57×10^8	36×10^7	11×10^6	66	−0.8	−0.04	-6×10^{-7}	1.01
141	3; 1–9, 2–6, 4–7, 5–8	0.28680(59)	14×10^9	48×10^7	19×10^4	9	0.03	0.001	5×10^{-21}	1.66
142	3; 1–9, 2–6, 4–8, 5–7	−0.44365(44)	69×10^8	38×10^7	14×10^4	2	−0.09	−0.0005	-9×10^{-49}	1.18
143	3; 1–9, 2–7, 4–5, 6–8	1.7563(10)	22×10^9	12×10^8	63×10^5	3	0.08	0.003	-3×10^{-24}	1.48
144	3; 1–9, 2–7, 4–6, 5–8	−0.23678(48)	77×10^8	41×10^7	13×10^4	4	0.2	0.0004	9×10^{-35}	1.28
145	3; 1–9, 2–7, 4–8, 5–6	2.58457(63)	12×10^9	42×10^7	17×10^5	9	0.7	−0.0002	5×10^{-40}	1.28
146	3; 1–9, 2–8, 4–5, 6–7	−6.34999(51)	60×10^8	19×10^7	14×10^6	290	−0.5	−0.03	-2×10^{-6}	1.00
147	3; 1–9, 2–8, 4–6, 5–7	7.46261(54)	82×10^8	60×10^7	78×10^4	12	1	0.004	4×10^{-25}	1.02
148	3; 1–9, 2–8, 4–7, 5–6	−1.98177(39)	55×10^8	43×10^7	21×10^5	7	−0.1	−0.007	-4×10^{-11}	1.01

TABLE IX. Contributions of graphs from the gauge-invariant class (1,2,1) to $A_1^{(8)}$.

Number	Graph	Value	N_{total}	$N_{\text{EIA}}^{\text{fail}}$	$N_{\text{IA}}^{\text{fail}}$	N_{128}^{fail}	$\triangle_{\text{EIA}}^{\text{fail}}$	$\triangle_{\text{IA}}^{\text{fail}}$	$\triangle_{128}^{\text{fail}}$	$\sigma_\uparrow/\sigma_\downarrow$
149	4; 1–3, 2–6, 5–8, 7–9	13.6554(10)	27×10^9	18×10^8	75542	4	5	0.01	-2×10^{-45}	1.08
150	4; 1–3, 2–6, 5–9, 7–8	−12.6376(13)	41×10^9	22×10^8	43×10^5	3	−5	−0.2	-10^{-56}	1.32
151	4; 1–3, 2–7, 5–8, 6–9	2.72526(52)	99×10^8	64×10^7	10^5	4	1	0.0008	5×10^{-48}	1.50
152	4; 1–3, 2–7, 5–9, 6–8	6.70242(77)	17×10^9	11×10^8	11×10^4	3	3	0.01	10^{-26}	1.04
153	4; 1–3, 2–8, 5–6, 7–9	−15.7206(11)	26×10^9	23×10^8	11×10^6	4	−7	−0.4	2×10^{-8}	1.02
154	4; 1–3, 2–8, 5–7, 6–9	5.17997(78)	17×10^9	11×10^8	93380	2	3	0.01	10^{-43}	1.02
155	4; 1–3, 2–8, 5–9, 6–7	−9.33944(81)	18×10^9	97×10^7	14×10^5	4	−3	−0.1	-10^{-49}	1.05
156	4; 1–3, 2–9, 5–6, 7–8	18.6188(12)	20×10^9	18×10^8	60×10^6	1162	7	0.9	0.0001	1.04
157	4; 1–3, 2–9, 5–7, 6–8	−22.2947(12)	33×10^9	32×10^8	34×10^5	7	−8	−0.1	5×10^{-34}	1.01
158	4; 1–3, 2–9, 5–8, 6–7	12.1677(10)	24×10^9	22×10^8	11×10^6	8	5	0.3	2×10^{-6}	1.06
159	4; 1–6, 2–3, 5–8, 7–9	−14.2179(11)	28×10^9	19×10^8	64×10^5	0	−5	−0.2	0	1.15
160	4; 1–6, 2–3, 5–9, 7–8	13.6681(13)	32×10^9	14×10^8	35×10^6	145	4	0.5	8×10^{-5}	1.27
161	4; 1–7, 2–3, 5–8, 6–9	−2.87192(46)	83×10^8	56×10^7	17×10^5	1	−1	−0.02	8×10^{-44}	1.38
162	4; 1–7, 2–3, 5–9, 6–8	−7.13177(83)	18×10^9	13×10^8	42×10^5	3	−3	−0.1	6×10^{-28}	1.12
163	4; 1–8, 2–3, 5–6, 7–9	15.4192(12)	20×10^9	18×10^8	60×10^6	1189	7	0.9	0.0001	1.01
164	4; 1–8, 2–3, 5–7, 6–9	−5.66590(79)	16×10^9	12×10^8	41×10^5	3	−3	−0.1	2×10^{-12}	1.03
165	4; 1–8, 2–3, 5–9, 6–7	10.43578(83)	15×10^9	65×10^7	16×10^6	67	3	0.4	3×10^{-5}	1.05
166	4; 1–9, 2–3, 5–6, 7–8	−17.4838(13)	15×10^9	75×10^7	15×10^7	35968	−5	−2	−0.001	1.03
167	4; 1–9, 2–3, 5–7, 6–8	21.0812(13)	30×10^9	31×10^8	35×10^6	388	7	0.6	4×10^{-5}	1.01
168	4; 1–9, 2–3, 5–8, 6–7	−12.9121(11)	18×10^9	17×10^8	60×10^6	1231	−4	−0.7	−0.0001	1.03

TABLE X. Contributions of graphs from the gauge-invariant class (3,1,0) to $A_1^{(8)}$.

Number	Graph	Value	N_{total}	$N_{\text{EIA}}^{\text{fail}}$	$N_{\text{IA}}^{\text{fail}}$	N_{128}^{fail}	$\triangle_{\text{EIA}}^{\text{fail}}$	$\triangle_{\text{IA}}^{\text{fail}}$	$\triangle_{128}^{\text{fail}}$	$\sigma_\uparrow/\sigma_\downarrow$
169	4; 1–5, 2–6, 3–8, 7–9	−1.02160(39)	67×10^8	28×10^7	29×10^4	26	−0.3	0.002	10^{-8}	1.23
170	4; 1–5, 2–6, 3–9, 7–8	0.82043(44)	83×10^8	26×10^7	83×10^4	64	0.3	0.004	-2×10^{-7}	1.54
171*	4; 1–5, 2–7, 3–8, 6–9	−1.35615(40)	88×10^8	89×10^6	14×10^4	6	−0.3	−0.004	10^{-10}	1.39
172	4; 1–5, 2–7, 3–9, 6–8	−0.88139(29)	42×10^8	14×10^7	56×10^4	115	−0.2	0.009	2×10^{-7}	1.18
173	4; 1–5, 2–8, 3–6, 7–9	−4.37354(62)	14×10^9	61×10^7	31×10^4	3	−2	−0.0004	7×10^{-24}	1.26
174*	4; 1–5, 2–8, 3–7, 6–9	0.16235(32)	59×10^8	87×10^6	21×10^4	3	0.1	0.0009	6×10^{-15}	1.81
175	4; 1–5, 2–8, 3–9, 6–7	0.91185(27)	44×10^8	79×10^6	46×10^4	40	0.1	−0.002	-6×10^{-8}	1.27
176	4; 1–5, 2–9, 3–6, 7–8	4.01347(73)	19×10^9	72×10^7	14×10^5	2	2	0.05	2×10^{-29}	1.41
177	4; 1–5, 2–9, 3–7, 6–8	−2.46028(48)	91×10^8	31×10^7	25×10^4	3	−0.7	0.0002	-10^{-19}	1.27
178	4; 1–5, 2–9, 3–8, 6–7	3.40092(52)	11×10^9	16×10^7	50×10^4	6	0.7	0.008	-2×10^{-12}	1.30
179	4; 1–6, 2–5, 3–8, 7–9	−3.77024(58)	13×10^9	58×10^7	29×10^4	5	−1	0.0003	5×10^{-44}	1.25
180	4; 1–6, 2–5, 3–9, 7–8	3.86148(80)	23×10^9	94×10^7	17×10^5	3	1	0.04	10^{-23}	1.81
181*	4; 1–6, 2–7, 3–8, 5–9	1.19458(39)	93×10^8	10^8	51×10^4	10	0.3	0.006	-9×10^{-11}	1.41
182*	4; 1–6, 2–7, 3–9, 5–8	0.80341(31)	54×10^8	94×10^6	37×10^4	24	0.2	0.0009	10^{-14}	1.46
183	4; 1–6, 2–8, 3–5, 7–9	3.47691(61)	12×10^9	93×10^7	17×10^4	1	1	0.01	-9×10^{-15}	1.07
184*	4; 1–6, 2–8, 3–7, 5–9	−0.41899(25)	39×10^8	55×10^6	40×10^4	53	−0.1	−0.0003	-3×10^{-11}	1.37
185	4; 1–6, 2–8, 3–9, 5–7	0.09060(28)	43×10^8	15×10^7	30×10^4	59	0.06	0.002	6×10^{-8}	1.33
186	4; 1–6, 2–9, 3–5, 7–8	−4.54867(60)	12×10^9	10^9	34×10^5	17	−2	−0.1	-10^{-5}	1.04
187*	4; 1–6, 2–9, 3–7, 5–8	0.14183(24)	39×10^8	12×10^7	23×10^4	0	0.07	0.0001	0	1.46
188	4; 1–6, 2–9, 3–8, 5–7	−1.30271(29)	48×10^8	19×10^7	23×10^4	3	−0.2	0.0001	2×10^{-22}	1.14
189*	4; 1–7, 2–5, 3–8, 6–9	0.24264(22)	34×10^8	65×10^6	24×10^4	30	−0.008	0.0002	4×10^{-15}	1.25
190	4; 1–7, 2–5, 3–9, 6–8	−2.56229(52)	11×10^9	46×10^7	28×10^4	3	−0.9	−0.0005	-2×10^{-24}	1.44
191*	4; 1–7, 2–6, 3–8, 5–9	−1.56685(32)	55×10^8	50×10^6	49×10^4	61	−0.3	−0.0007	5×10^{-8}	1.35
192*	4; 1–7, 2–6, 3–9, 5–8	−0.42860(29)	54×10^8	83×10^6	20×10^4	8	−0.08	0.0006	10^{-18}	1.59
193	4; 1–7, 2–8, 3–5, 6–9	0.11285(31)	58×10^8	36×10^7	11×10^4	4	−0.01	0.0003	-6×10^{-43}	1.10
194*	4; 1–7, 2–8, 3–6, 5–9	0.75665(18)	31×10^8	77×10^6	14×10^4	6	0.1	0.001	3×10^{-18}	1.31

(Table continued)

TABLE X. *(Continued)*

Number	Graph	Value	N_{total}	$N_{\text{EIA}}^{\text{fail}}$	$N_{\text{IA}}^{\text{fail}}$	N_{128}^{fail}	$\triangle_{\text{EIA}}^{\text{fail}}$	$\triangle_{\text{IA}}^{\text{fail}}$	$\triangle_{128}^{\text{fail}}$	$\sigma_\uparrow/\sigma_\downarrow$
195	4; 1–7, 2–8, 3–9, 5–6	−0.61298(33)	65×10^8	96×10^6	42×10^4	5	−0.1	−0.001	4×10^{-37}	1.38
196	4; 1–7, 2–9, 3–5, 6–8	2.62642(55)	11×10^9	73×10^7	17×10^4	1	0.9	0.009	-8×10^{-81}	1.11
197*	4; 1–7, 2–9, 3–6, 5–8	1.02944(34)	55×10^8	15×10^7	40×10^4	49	0.3	0.0003	-3×10^{-14}	1.71
198	4; 1–7, 2–9, 3–8, 5–6	−0.05084(72)	23×10^9	25×10^7	67×10^4	13	−0.1	−0.002	-2×10^{-25}	1.34
199	4; 1–8, 2–5, 3–6, 7–9	11.5072(10)	34×10^9	22×10^8	12×10^5	12	4	0.02	4×10^{-20}	1.34
200*	4; 1–8, 2–5, 3–7, 6–9	−2.26508(42)	89×10^8	18×10^7	18×10^4	1	−0.6	−0.0003	4×10^{-21}	1.34
201	4; 1–8, 2–5, 3–9, 6–7	2.45160(46)	10^{10}	20×10^7	26×10^4	4	0.8	0.02	10^{-10}	1.19
202	4; 1–8, 2–6, 3–5, 7–9	−6.43899(92)	25×10^9	20×10^8	19×10^5	7	−2	−0.05	2×10^{-37}	1.02
203*	4; 1–8, 2–6, 3–7, 5–9	2.17129(39)	84×10^8	11×10^7	13×10^4	1	0.6	0.002	-7×10^{-45}	1.35
204	4; 1–8, 2–6, 3–9, 5–7	−0.69905(42)	79×10^8	25×10^7	21×10^4	5	0.07	0.0008	2×10^{-31}	1.47
205	4; 1–8, 2–7, 3–5, 6–9	0.84604(33)	63×10^8	40×10^7	12×10^4	3	0.2	0.001	10^{-34}	1.27
206*	4; 1–8, 2–7, 3–6, 5–9	−0.21952(37)	84×10^8	17×10^7	18×10^4	9	−0.05	−0.005	-3×10^{-25}	1.42
207	4; 1–8, 2–7, 3–9, 5–6	2.13842(53)	13×10^9	16×10^7	43×10^4	6	0.2	0.001	-5×10^{-36}	1.28
208	4; 1–8, 2–9, 3–5, 6–7	−3.03246(61)	12×10^9	10^9	24×10^5	13	−0.9	−0.06	-6×10^{-7}	1.30
209	4; 1–8, 2–9, 3–6, 5–7	−0.90616(40)	69×10^8	40×10^7	22×10^4	7	−0.5	−0.002	-9×10^{-19}	1.17
210	4; 1–8, 2–9, 3–7, 5–6	0.81006(30)	51×10^8	16×10^7	42×10^4	4	0.3	0.002	-4×10^{-12}	1.09
211	4; 1–9, 2–5, 3–6, 7–8	−12.5566(11)	31×10^9	22×10^8	15×10^6	128	−4	−0.2	-5×10^{-6}	1.27
212	4; 1–9, 2–5, 3–7, 6–8	18.0227(11)	38×10^9	16×10^8	10^6	16	5	0.01	7×10^{-25}	1.26
213	4; 1–9, 2–5, 3–8, 6–7	−12.9501(11)	37×10^9	97×10^7	45×10^5	12	−3	−0.2	-3×10^{-21}	1.27
214	4; 1–9, 2–6, 3–5, 7–8	7.41689(93)	20×10^9	21×10^8	20×10^6	326	2	0.3	3×10^{-5}	1.02
215	4; 1–9, 2–6, 3–7, 5–8	−3.84552(63)	19×10^9	43×10^7	37×10^4	10	−0.8	−0.003	-3×10^{-11}	1.65
216	4; 1–9, 2–6, 3–8, 5–7	1.17277(59)	11×10^9	52×10^7	57×10^4	10	0.2	0.0003	3×10^{-20}	1.50
217	4; 1–9, 2–7, 3–5, 6–8	−13.3320(11)	32×10^9	24×10^8	21×10^5	3	−3	−0.06	3×10^{-44}	1.02
218	4; 1–9, 2–7, 3–6, 5–8	−0.83706(55)	13×10^9	37×10^7	18×10^4	6	−0.4	−0.003	-2×10^{-23}	1.41
219	4; 1–9, 2–7, 3–8, 5–6	−0.25085(86)	25×10^9	55×10^7	17×10^5	24	−0.2	0.0004	-7×10^{-24}	1.49
220	4; 1–9, 2–8, 3–5, 6–7	13.1985(12)	29×10^9	24×10^8	24×10^6	347	3	0.4	4×10^{-5}	1.03
221	4; 1–9, 2–8, 3–6, 5–7	2.13571(75)	17×10^9	10^9	89×10^4	20	0.9	0.008	-7×10^{-19}	1.35
222	4; 1–9, 2–8, 3–7, 5–6	−3.87084(43)	68×10^8	28×10^7	18×10^5	6	−0.7	−0.01	7×10^{-42}	1.00

TABLE XI. Contributions of graphs from the gauge-invariant class (2,1,1) to $A_1^{(8)}$.

Number	Graph	Value	N_{total}	$N_{\text{EIA}}^{\text{fail}}$	$N_{\text{IA}}^{\text{fail}}$	N_{128}^{fail}	$\triangle_{\text{EIA}}^{\text{fail}}$	$\triangle_{\text{IA}}^{\text{fail}}$	$\triangle_{128}^{\text{fail}}$	$\sigma_\uparrow/\sigma_\downarrow$
223	5; 1–3, 2–6, 4–8, 7–9	−6.61670(58)	11×10^9	61×10^7	58456	0	−2	−0.002	0	1.08
224	5; 1–3, 2–6, 4–9, 7–8	10.3187(10)	30×10^9	18×10^8	37×10^5	1	4	0.1	4×10^{-66}	1.53
225	5; 1–3, 2–7, 4–8, 6–9	0.70044(49)	95×10^8	46×10^7	32×10^4	26	0.06	0.0007	-10^{-7}	1.52
226	5; 1–3, 2–7, 4–9, 6–8	−2.37520(44)	73×10^8	50×10^7	10^5	1	−0.5	-2×10^{-5}	-7×10^{-44}	1.16
227	5; 1–3, 2–8, 4–6, 7–9	4.07903(69)	13×10^9	13×10^8	52×10^4	1	1	0.02	-10^{-46}	1.03
228	5; 1–3, 2–8, 4–7, 6–9	2.09761(43)	87×10^8	38×10^7	69528	2	0.7	0.001	10^{-40}	1.30
229	5; 1–3, 2–8, 4–9, 6–7	3.36347(78)	19×10^9	83×10^7	10^6	3	0.5	5×10^{-5}	-2×10^{-17}	1.38
230	5; 1–3, 2–9, 4–6, 7–8	−9.9012(11)	24×10^9	29×10^8	20×10^6	309	−3	−0.3	-2×10^{-5}	1.04
231	5; 1–3, 2–9, 4–7, 6–8	−3.37250(75)	15×10^9	13×10^8	32×10^4	6	−2	−0.008	-2×10^{-23}	1.38
232	5; 1–3, 2–9, 4–8, 6–7	1.69133(37)	55×10^8	36×10^7	10^6	2	0.5	0.006	-6×10^{-32}	1.03
233	5; 1–4, 2–6, 3–9, 7–8	−0.79932(49)	92×10^8	38×10^7	78×10^4	66	−0.2	−0.001	-5×10^{-9}	1.61
234*	5; 1–4, 2–7, 3–8, 6–9	1.03920(23)	41×10^8	61×10^6	63265	1	0.2	0.001	5×10^{-53}	1.44
235	5; 1–4, 2–7, 3–9, 6–8	1.91364(42)	65×10^8	26×10^7	73×10^4	166	0.4	0.003	-4×10^{-8}	1.56
236*	5; 1–4, 2–8, 3–7, 6–9	0.00390(12)	23×10^8	38×10^6	29155	1	−0.04	−0.0006	5×10^{-48}	1.31
237	5; 1–4, 2–8, 3–9, 6–7	−3.32287(43)	91×10^8	16×10^7	31×10^4	6	−0.3	−0.003	-10^{-25}	1.47
238	5; 1–4, 2–9, 3–6, 7–8	−2.51836(53)	12×10^9	47×10^7	63×10^4	1	−0.8	−0.007	3×10^{-54}	1.47
239	5; 1–4, 2–9, 3–7, 6–8	2.33158(48)	96×10^8	34×10^7	74058	4	0.9	0.001	8×10^{-44}	1.23
240	5; 1–4, 2–9, 3–8, 6–7	−1.31498(59)	15×10^9	25×10^7	48×10^4	5	−0.3	−0.004	4×10^{-20}	1.27

(Table continued)

TABLE XI. *(Continued)*

Number	Graph	Value	N_{total}	$N_{\text{EIA}}^{\text{fail}}$	$N_{\text{IA}}^{\text{fail}}$	N_{128}^{fail}	$\triangle_{\text{EIA}}^{\text{fail}}$	$\triangle_{\text{IA}}^{\text{fail}}$	$\triangle_{128}^{\text{fail}}$	$\sigma_\uparrow/\sigma_\downarrow$
241	5; 1–6, 2–3, 4–9, 7–8	−4.16476(65)	11×10^9	37×10^7	11×10^6	49	−1	−0.2	-2×10^{-5}	1.31
242	5; 1–6, 2–4, 3–9, 7–8	0.69243(44)	71×10^8	49×10^7	10^6	5	0.3	0.001	5×10^{-36}	1.30
243	5; 1–6, 2–9, 3–4, 7–8	−1.10140(94)	23×10^9	62×10^7	18×10^6	77	−0.4	−0.002	3×10^{-13}	1.49
244	5; 1–7, 2–4, 3–9, 6–8	1.17746(35)	53×10^8	28×10^7	10^5	4	0.7	0.001	7×10^{-37}	1.05
245	5; 1–7, 2–9, 3–4, 6–8	−2.69013(59)	11×10^9	51×10^7	60×10^4	1	−2	−0.04	-4×10^{-26}	1.07
246	5; 1–8, 2–9, 3–4, 6–7	1.81548(42)	69×10^8	13×10^7	34×10^5	13	0.6	0.07	7×10^{-7}	1.21
247	5; 1–9, 2–3, 4–6, 7–8	5.84579(86)	11×10^9	14×10^8	51×10^6	7424	2	0.4	0.0002	1.19
248	5; 1–9, 2–3, 4–7, 6–8	2.98166(85)	17×10^9	17×10^8	72×10^5	41	2	0.04	2×10^{-7}	1.42
249	5; 1–9, 2–3, 4–8, 6–7	−1.68619(46)	56×10^8	35×10^7	11×10^6	67	−0.8	−0.04	-4×10^{-7}	1.00
250	5; 1–9, 2–4, 3–7, 6–8	−10.38002(90)	19×10^9	11×10^8	36×10^4	7	−3	−0.01	4×10^{-30}	1.05
251	5; 1–9, 2–4, 3–8, 6–7	21.6246(13)	37×10^9	19×10^8	87×10^5	7	6	0.2	-10^{-6}	1.15
252	5; 1–9, 2–8, 3–4, 6–7	−10.34846(83)	14×10^9	31×10^7	25×10^6	478	−2	−0.3	-4×10^{-5}	1.02

TABLE XII. Contributions of graphs from the gauge-invariant class (4,0,0) to $A_1^{(8)}$.

Number	Graph	Value	N_{total}	$N_{\text{EIA}}^{\text{fail}}$	$N_{\text{IA}}^{\text{fail}}$	N_{128}^{fail}	$\triangle_{\text{EIA}}^{\text{fail}}$	$\triangle_{\text{IA}}^{\text{fail}}$	$\triangle_{128}^{\text{fail}}$	$\sigma_\uparrow/\sigma_\downarrow$
253*	5; 1–6, 2–7, 3–8, 4–9	0.29657(24)	49×10^8	57×10^6	22×10^4	8	0.2	0.01	3×10^{-10}	1.36
254*	5; 1–6, 2–7, 3–9, 4–8	−0.47196(32)	55×10^8	70×10^6	64×10^4	97	−0.1	−0.003	-6×10^{-9}	1.65
255*	5; 1–6, 2–8, 3–7, 4–9	−0.57757(12)	21×10^8	31×10^6	23×10^4	31	−0.1	−0.001	-2×10^{-7}	1.42
256*	5; 1–6, 2–8, 3–9, 4–7	0.21265(21)	39×10^8	77×10^6	10^5	0	−0.01	-5×10^{-5}	0	1.62
257*	5; 1–6, 2–9, 3–7, 4–8	−1.01853(40)	79×10^8	16×10^7	40×10^4	6	−0.4	−0.002	-7×10^{-17}	1.48
258*	5; 1–6, 2–9, 3–8, 4–7	−0.01236(43)	95×10^8	27×10^7	46×10^4	12	−0.1	−0.006	10^{-18}	1.54
259*	5; 1–7, 2–6, 3–9, 4–8	0.49710(18)	32×10^8	29×10^6	16×10^4	3	0.09	0.0005	8×10^{-22}	1.40
260	5; 1–7, 2–8, 3–9, 4–6	0.60670(24)	43×10^8	21×10^7	17×10^4	6	0.1	0.0008	-2×10^{-27}	1.23
261*	5; 1–7, 2–9, 3–6, 4–8	−1.03019(37)	63×10^8	12×10^7	44×10^4	51	−0.4	−0.0001	-10^{-13}	1.36
262	5; 1–7, 2–9, 3–8, 4–6	−0.19243(34)	66×10^8	34×10^7	21×10^4	4	0.1	0.001	10^{-22}	1.22
263*	5; 1–8, 2–9, 3–6, 4–7	2.32056(35)	70×10^8	23×10^7	28×10^4	3	0.6	0.002	-3×10^{-39}	1.26
264	5; 1–8, 2–9, 3–7, 4–6	−1.30603(29)	50×10^8	28×10^7	24×10^4	5	−0.3	−0.004	-3×10^{-29}	1.09
265	5; 1–9, 2–6, 3–7, 4–8	0.64498(32)	59×10^8	14×10^7	56×10^4	31	0.2	0.005	-9×10^{-9}	1.38
266	5; 1–9, 2–6, 3–8, 4–7	5.46569(76)	22×10^9	62×10^7	97×10^4	15	1	0.001	-3×10^{-12}	1.48
267	5; 1–9, 2–7, 3–8, 4–6	−2.43882(45)	89×10^8	49×10^7	38×10^4	10	−0.4	−0.007	3×10^{-21}	1.15
268	5; 1–9, 2–8, 3–6, 4–7	−6.78187(74)	20×10^9	69×10^7	11×10^5	20	−1	−0.01	-2×10^{-19}	1.27
269*	5; 1–9, 2–8, 3–7, 4–6	4.29748(67)	14×10^9	89×10^7	19×10^5	18	0.7	0.03	10^{-35}	1.03

TABLE XIII. Contributions to $A_1^{(4)}$ (see Fig. 3) that must coincide with the values that are obtained by direct subtraction on the mass shell in the Feynman gauge, and a comparison of these results with the values from Ref. [5] and with the old values from Ref. [30].

Set of graphs	Value	Analytical value	Value from Ref. [30]
1–2	−0.6539950(23)	−0.653998963627	−0.654032(54)
3	−0.4676475(17)	−0.467645446094	−0.467626(44)
4	0.7774774(18)	0.777478022283	0.777455(52)

The contributions of the gauge-invariant classes (k, m, m') (see the definition in Sec. IV G) and their comparison with the semianalytical results from Ref. [8] are presented in Table XVI.

The equivalence of the subtraction procedure from Sec. II and the direct subtraction on the mass shell for all presented sets can be proved in a combinatorial way.[57] Let us consider an example: the sets 26,27 from Table XIV. The contribution of these sets can be schematically written as

[57]if we do not consider the matter of divergence regularizations.

TABLE XIV. Contributions to $A_1^{(6)}$ (see Fig. 4) that must coincide with the values that are obtained by direct subtraction on the mass shell in the Feynman gauge, and a comparison of these results with the known analytical values and with the old values from Ref. [30].

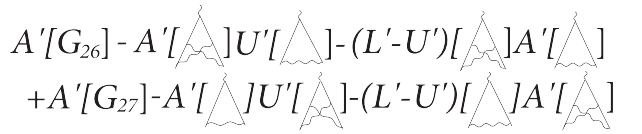
Set of graphs	Value	Analytical value	Reference[a]	Value from Ref. [30]
1–10	0.533289(54)	0.533355	[7,14–17,19,21]	0.5340(18)
11–12	1.541644(37)	1.541649	[15,17]	1.5436(34)
13	−1.757945(15)	−1.757936	[7]	−1.7579(10)
14, 17	0.455517(26)	0.455452	[19,21]	0.4549(14)
15, 18–20	−0.402749(46)	−0.402717	[14,15]	−0.4030(41)
16	−0.334691(14)	−0.334695	[19]	−0.33468(95)
21–23	0.421080(43)	0.421171	[14,15,17]	0.4207(22)
24	−0.0267956(78)	−0.026799	[7]	−0.02688(47)
25	1.861914(17)	1.861908	[19]	1.8629(14)
26–27	−3.176700(22)	−3.176685	[16,21]	−3.1764(22)
28	1.790285(19)	1.790278	[16]	1.7888(19)

[a]More precisely, the expressions from Ref. [17] are semianalytical. The corresponding analytical expressions are given in Ref. [24].

TABLE XV. Contributions to $A_1^{(8)}$ that must coincide with the values that are obtained by direct subtraction on the mass shell in the Feynman gauge.

Set of graphs	Value	N_{total}	$\sigma_\uparrow/\sigma_\downarrow$
1–74	−1.9710(44)	59×10^{10}	1.32
75–78, 82–83, 93–94, 101, 133	−2.0858(26)	19×10^{10}	1.39
79, 89, 104, 116	9.2853(15)	64×10^{9}	1.34
80–81, 84, 92, 105–106, 117–118, 131–132	−7.3999(19)	12×10^{10}	1.35
85–86	0.91509(81)	29×10^{9}	1.40
88, 113	−0.03943(45)	11×10^{9}	1.24
91, 114	−1.21525(47)	10^{10}	1.28
95–96, 107–108, 120–121, 125, 134–139, 141–142, 144–148	11.6975(35)	30×10^{10}	1.14
97–98	0.07633(84)	30×10^{9}	1.77
103, 115	−0.21851(49)	13×10^{9}	1.50
110, 124	−1.67843(52)	13×10^{9}	1.35
119, 122, 140, 143	−10.6235(17)	69×10^{9}	1.20
127–128	−2.10043(82)	29×10^{9}	1.34
129–130	1.17276(61)	17×10^{9}	1.34
149–168	−0.6220(46)	44×10^{10}	1.08
169–170	−0.20117(59)	15×10^{9}	1.38
172, 175	0.03046(39)	87×10^{8}	1.22
173, 180	−0.5121(10)	38×10^{9}	1.53
176, 179	0.24323(93)	33×10^{9}	1.34
177–178	0.94064(71)	20×10^{9}	1.29
183, 208, 212, 219	18.2163(17)	89×10^{9}	1.28
185, 195	−0.52238(43)	10^{10}	1.36
186, 199, 209, 213	−6.8978(17)	91×10^{9}	1.25
188, 198	−1.35354(78)	28×10^{9}	1.30
190, 201	−0.11069(69)	21×10^{9}	1.31
193, 215	−3.73267(70)	25×10^{9}	1.48
196, 210–211, 216	−7.9473(14)	59×10^{9}	1.26
202, 214, 217, 220–222	−0.8907(22)	13×10^{10}	1.05
204, 207	1.43937(67)	21×10^{9}	1.35
205, 218	0.00898(64)	19×10^{9}	1.36
223–224, 241	−0.4627(14)	53×10^{9}	1.36
225, 233	−0.09888(69)	18×10^{9}	1.56
226, 229, 242–243	0.5793(14)	57×10^{9}	1.38
227, 230, 247, 250–252	0.9197(24)	12×10^{10}	1.08
228, 238	−0.42075(69)	21×10^{9}	1.40
231–232, 248–249	−0.3857(13)	44×10^{9}	1.28
235, 237	−1.40923(60)	15×10^{9}	1.52
239–240	1.01660(76)	25×10^{9}	1.25
244–246	0.30280(80)	24×10^{9}	1.10
260, 265	1.25169(40)	10^{10}	1.32
262, 266	5.27326(83)	28×10^{9}	1.42
264, 267–268	−10.52672(91)	34×10^{9}	1.21

(Table continued)



Here, A', L', U' are operators that are applied to Feynman amplitudes and return numbers:

$$A\Gamma_\mu = e\gamma_\mu(A'\Gamma_\mu), \quad L\Gamma_\mu = e\gamma_\mu(L'\Gamma_\mu), \quad U\Gamma_\mu = e\gamma_\mu(U'\Gamma_\mu),$$

where the definitions from Sec. II are used, and a constant multiplier is omitted. Analogously, the corresponding contribution that is obtained by the direct subtraction on the mass shell is

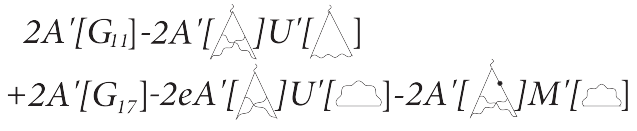
TABLE XVI. Contributions of the gauge-invariant classes (k, m, m') to $A_1^{(8)}$, and a comparison of these results with the semianalytical values from Ref. [8].

Class	Value	Semianalytical value	N_{total}	$\sigma_\uparrow/\sigma_\downarrow$
(1,3,0)	−1.9710(44)	−1.9710756168358	59×10^{10}	1.32
(2,2,0)	−0.1415(56)	−0.1424873797999	96×10^{10}	1.26
(1,2,1)	−0.6220(46)	−0.6219210635351	44×10^{10}	1.08
(3,1,0)	−1.0424(44)	−1.0405424100126	70×10^{10}	1.23
(2,1,1)	1.0842(37)	1.0866983944758	38×10^{10}	1.21
(4,0,0)	0.5120(17)	0.512462047968	13×10^{10}	1.28

TABLE XVII. Summary of the results, comparison with the known (semi)analytical results, technical information.

	2 loops	3 loops	4 loops	5-loop ladder	6-loop ladder
Value	−0.3441651(34)	0.90485(10)	−2.181(10)	11.6530(58)	34.31(20)
(Semi)analytical value for comparison	−0.344166387	0.904979	−2.1769	11.6592	34.367
References for the (semi)analytical value	[5]	[7,14–17,19]	[8]	[49]	[49]
$\sigma_\uparrow/\sigma\downarrow$	1.02	1.05	1.21	1.16	1.74
N_{total}	33×10^{11}	81×10^{11}	32×10^{11}	29×10^{9}	10^{10}
$N_{\text{EIA}}^{\text{fail}}$	71×10^{8}	17×10^{10}	18×10^{10}	32×10^{8}	12×10^{8}
$N_{\text{IA}}^{\text{fail}}$	68×10^{6}	21×10^{8}	13×10^{8}	90×10^{5}	72×10^{5}
N_{128}^{fail}	2	12590	77775	934	4504
$\triangle_{\text{EIA}}^{\text{fail}}$	0.002	0.4	2	5	20
$\triangle_{\text{IA}}^{\text{fail}}$	0.0001	0.002	0.2	0.4	3
$\triangle_{128}^{\text{fail}}$	-2×10^{-19}	-10^{-6}	−0.0006	4×10^{-10}	-5×10^{-5}
Total calculation time	21 h 37 min	5 d 8 h	7 d	4 h 38 min	8 h 24 min
Share in the time: double-precision EIA	19.1%	41.7%	54.5%	56.4%	42.0%
Share in the time: double-precision IA	0.1%	1.6%	9.1%	15.4%	24.4%
Share in the time: 128-bit mantissa IA	0.2%	2.7%	9.2%	6.7%	24.3%
Share in the time: 256-bit mantissa IA	0.0%	0.3%	2.1%	8.1%	5.2%
Share in the time: sample generation	63.7%	45.9%	21.7%	12.0%	3.7%
Share in the time: other operations	16.9%	7.7%	3.4%	1.3%	0.3%
GPU speed: double-precision EIA, GFlop/s	334.24	222.72	234.26	187.93	292.67
GPU speed: double-precision EIA, GInterval/s	53.76	63.51	142.27	103.04	240.91
GPU speed: double-precision IA, GFlop/s	254.11	221.41	255.85	249.00	287.94
GPU speed: double-precision IA, GInterval/s	36.23	35.80	47.22	45.60	55.81
GPU speed: 128-bit mantissa IA, GFlop/s	0.81	1.59	1.58	1.63	1.66
GPU speed: 128-bit mantissa IA, GInterval/s	0.11	0.23	0.26	0.30	0.32
GPU speed: 256-bit mantissa IA, MFlop/s	0.0204	0.0881	0.3503	0.1378	4.8504
GPU speed: 256-bit mantissa IA, MInterval/s	0.0028	0.0124	0.0537	0.0252	0.9401
Integrand code size: not compiled	887 KB	31 MB	2.5 GB	23 MB	186 MB
Integrand code size: compiled	12 MB	115 MB	4 GB	34 MB	252 MB

It is easy to see that these expressions are equivalent. Let us consider another example: the sets 11,17 from Table XIV. The contribution of these sets is

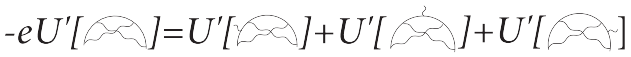


Here, the operators U' and M' that are applied to Feynman amplitudes of self-energy subgraphs are defined by

$$U\Sigma(p) = e[M'\Sigma + (U'\Sigma)(\hat{p} - m)].$$

The terms containing U' are canceled, because U preserves the Ward identity; see Ref. [30]. An analogous cancellation works for the direct subtraction expression and leads to the same result.

Sometimes for proving the equivalence it is necessary to use the Ward identity for individual Feynman graphs; see Ref. [50]. For example, for the operator U' we can use the following equality:



The right part of this equality contains all possible insertions of an external photon line to the graph from the left part.

I. Technical information

Table XVII contains a summary of results and technical information. The meanings of the fields N_{total}, $N_{\text{EIA}}^{\text{fail}}$, $N_{\text{IA}}^{\text{fail}}$, N_{128}^{fail}, $\triangle_{\text{EIA}}^{\text{fail}}$, $\triangle_{\text{IA}}^{\text{fail}}$, and $\triangle_{128}^{\text{fail}}$ are defined in Sec. IV G. The GPU performance[58] for these computations is measured in floating point operations per second (flop/s) and interval operations per second (interval/s) in the sense of Sec. IV B.

V. CONCLUSION

The method for the numerical evaluation of $A_1^{(2n)}$ [no lepton loops] described in Refs. [29,30] was significantly improved. The main improvements are

(1) Probability density functions for Monte Carlo integration giving a better convergence.
(2) A method of Monte Carlo error estimation.
(3) A method of high-speed arithmetic calculations with round-off error control.
(4) A realization on high-speed graphics processors.

The values for $n = 2$, 3, 4 were obtained and compared with the known analytical and semianalytical ones, as well as the contributions of the 5-loop and 6-loop ladder graphs. The results were presented in a form allowing us to check them by parts using other methods. The 2-loop and 3-loop contributions were compared with the known values in detail, and the 4-loop ones were compared for six gauge-invariant classes. All obtained results are in good agreement with the known ones. The results showed that the developed method and its realization allow us to obtain high-precision values for high-order QED contributions to a_e even without appealing to supercomputers.

The ability to use nonadaptive Monte Carlo algorithms for obtaining high-precision results was verified. The behavior of the Monte Carlo samples was analyzed in detail. The necessity of probability distribution extrapolation for obtaining correct error estimations was explained, and the method was presented. The impact of possible round-off errors was investigated in detail, the necessity of controlling them and applying high-precision arithmetic was justified. The developed high-speed method of controlling round-off errors can be used for other calculations in quantum field theory that are based on the numerical subtraction of divergences under the integral sign.

The performed 6-loop calculation showed a big impact of high-precision arithmetic to the calculation speed and the necessity of accurate error estimation, but the 3-loop calculation discovered a sensitivity to a selection of a pseudorandom generator.

The realization on GPUs showed very good performance. For example, the speed of obtaining integrand values was improved by 3000 times in comparison with Ref. [29] for the 5-loop ladder graph.

In closing, let us recapitulate some theoretical problems that still remain open:

(1) To prove mathematically (or disprove) that the developed subtraction procedure leads to finite integrals for any Feynman graph for any order of the perturbation series.
(2) To create a mathematical foundation for the probability density functions that were used for the Monte Carlo integration.
(3) To generalize the concept of I-closure and to develop a method of obtaining $\mathrm{Deg}(s)$ for graphs with lepton loops.
(4) To explain why the contributions of gauge-invariant classes are relatively small, but the contributions of individual graphs or even sets from Sec. IV H are relatively large; is this true for the higher orders of the perturbation series?

ACKNOWLEDGMENTS

The author thanks Andrey Kataev for interesting discussions and helpful recommendations, Andrey Arbuzov for his help in organizational issues, Predrag Cvitanović for fruitful discussion and inspiring ideas, Ivan Krasin for his help in understanding NVidia graphics accelerators and Google services, and Denis Shelomovskij for his help in D programming issues. Also, the author thanks Google for use of their computing services. Special thanks are due to the reviewers for careful reading of the article and valuable advices.

[58]The announced by NVidia peak performance of one GPU of the NVidia Tesla K80 for double precision is 1.45 Tflops.

[1] D. Hanneke, S. Fogwell Hoogerheide, and G. Gabrielse, Phys. Rev. A **83**, 052122 (2011).
[2] T. Aoyama, T. Kinoshita, and M. Nio, Phys. Rev. D **97**, 036001 (2018).
[3] J. Schwinger, Phys. Rev. **73**, 416 (1948).
[4] J. Schwinger, Phys. Rev. **76**, 790 (1949).
[5] A. Petermann, Helv. Phys. Acta **30**, 407 (1957).
[6] C. Sommerfield, Phys. Rev. **107**, 328 (1957).
[7] S. Laporta and E. Remiddi, Phys. Lett. B **379**, 283 (1996).
[8] S. Laporta, Phys. Lett. B **772**, 232 (2017).
[9] J. A. Mignaco and E. Remiddi, Nuovo Cimento A **60**, 519 (1969).
[10] R. Barbieri, M. Caffo, and E. Remiddi, Lett. Nuovo Cimento **5**, 769 (1972).
[11] D. Billi, M. Caffo, and E. Remiddi, Lett. Nuovo Cimento **4**, 657 (1972).
[12] R. Barbieri and E. Remiddi, Phys. Lett. B **49**, 468 (1974).
[13] R. Barbieri, M. Caffo, and E. Remiddi, Lett. Nuovo Cimento **9**, 690 (1974).
[14] M. J. Levine and R. Roskies, Phys. Rev. D **9**, 421 (1974).
[15] M. J. Levine, R. C. Perisho, and R. Roskies, Phys. Rev. D **13**, 997 (1976).
[16] R. Barbieri, M. Caffo, E. Remiddi, S.Turrini, and D.Oury, Nucl. Phys. **B144**, 329 (1978).
[17] M. J. Levine, E. Remiddi, and R. Roskies, Phys. Rev. D **20**, 2068 (1979).
[18] S. Laporta and E. Remiddi, Phys. Lett. B **265**, 182 (1991).
[19] S. Laporta, Phys. Lett. B **343**, 421 (1995).
[20] R. Barbieri, M. Caffo, and E. Remiddi, Phys. Lett. B **57**, 460 (1975).
[21] M. J. Levine and R. Roskies, Phys. Rev. D **14**, 2191 (1976).
[22] K. A. Milton, W. Tsai, and L. L. DeRaad, Jr, Phys. Rev. D **9**, 1809 (1974).
[23] L. L. DeRaad, Jr., K. A. Milton, and W. Tsai, Phys. Rev. D **9**, 1814 (1974).
[24] S. Laporta, Phys. Rev. D **47**, 4793 (1993).
[25] T. Aoyama, M. Hayakawa, T. Kinoshita, and M. Nio, Phys. Rev. D **91**, 033006 (2015).
[26] T. Kinoshita and W. B. Lindquist, Phys. Rev. Lett. **47**, 1573 (1981).
[27] F. Rappl, Ph.D. thesis, Universität Regensburg, 2016.
[28] P. Marquard, A. V. Smirnov, V. A. Smirnov, M. Steinhauser, and D. Wellmann, arXiv:1708.07138.
[29] S. Volkov, Phys. Rev. D **96**, 096018 (2017).
[30] S. Volkov, Zh. Eksp. Teor. Fiz. **149**, 6 (2015) [JETP **122**, 6 (2016)].
[31] M. J. Levine and J. Wright, Phys. Rev. D **8**, 3171 (1973).
[32] R. Carroll and Y. Yao, Phys. Lett. B **48**, 125 (1974).
[33] R. Carroll, Phys. Rev. D **12**, 2344 (1975).
[34] P. Cvitanović and T. Kinoshita, Phys. Rev. D **10**, 4007 (1974).
[35] L. Ts. Adzhemyan and M. V. Kompaniets, J. Phys. Conf. Ser. **523**, 012049 (2014).
[36] N. N. Bogoliubov and O. S. Parasiuk, Acta Math. **97**, 227 (1957).
[37] K. Hepp, Commun. Math. Phys. **2**, 301 (1966).
[38] O. I. Zavialov and B. M. Stepanov, Sov. J. Nucl. Phys. **1**, 922 (1965).
[39] V. A. Scherbina, *Catalogue of Deposited Papers* (VINITI, Moscow, 1964), Vol. 38.
[40] O. I. Zavialov, *Renormalized Quantum Field Theory* (Springer-Verlag, Berlin, 2012).
[41] V. A. Smirnov, Progress in Mathematical Physics, in *Renormalization and Asymptotic Expansions: PPH'14* (Birkhäuser, Basel, Switzerland, 2000).
[42] W. Zimmermann, Commun. Math. Phys. **15**, 208 (1969).
[43] E. Speer, J. Math. Phys. (N.Y.) **9**, 1404 (1968).
[44] P. Cvitanović and T. Kinoshita, Phys. Rev. D **10**, 3991 (1974).
[45] P. Cvitanović and T. Kinoshita, Phys. Rev. D **10**, 3978 (1974).
[46] A. Alexandrescu, *The D Programming Language* (Addison-Wesley, Reading, MA, 2010).
[47] CUDA C Programming Guide, NVIDIA Developer Documentation, https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html.
[48] CURAND Library, Programming Guide, NVIDIA Developer Documentation, https://docs.nvidia.com/cuda/curand/index.html.
[49] M. Caffo, S. Turrini, and E. Remiddi, Nucl. Phys. **B141**, 302 (1978).
[50] M. E. Peskin and D. V. Schroeder, *An Introduction to Quantum Field Theory* (Perseus Books, New York, 1995).