Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Tropical Feynman Period integration

Master's Thesis

Andrea Favorito

afavorito@student.ethz.ch

Institute of Theoretical Physics
ETH Zürich

Supervisors:
Prof. Dr. Charalampos Anastasiou
Dr. Michael Borinsky

March 30, 2023

Acknowledgements

I would like to express my deepest gratitude to my advisor Prof. Dr. Charalampos Anastasiou and co-advisor Dr. Michael Borinsky. Their mentorship has been invaluable in shaping my research skills and academic growth.

In particular, Dr. Borinsky's guidance, weekly feedback, and support have helped me navigate the challenges of the research process and provided me with a deeper understanding of the field of physics. I am grateful for the opportunity to learn from him and for the ways in which he has helped me develop as a researcher.

I would also like to thank my mother Alessandra, my father Maurizio and my brother Tati for their unconditional support, love, and patience throughout my academic journey. Their encouragement, motivation, and understanding have been a constant source of strength and inspiration, and I could not have accomplished this without them. My brother's enthusiasm for physics has been contagious and has motivated me to delve deeper into the subject.

I would like to thank all my friends, that have made this past few years studying fun and an enjoyable experience. In particular Marti, that always pushes and helps me become a more complete person in all aspects.

Last but not least, my heartfelt gratitude goes to my girlfriend Silvia, who has been a constant source of love, encouragement, and support throughout my academic journey. Her unwavering belief in me and our life together has given me the strength and motivation to pursue my passion for physics.

Thank you all for your support and encouragement.

Abstract

This thesis investigates the statistical properties and high order behaviour of Feynman periods, a specific class of Feynman integrals that play an important role in physics. Using the tropical Monte Carlo quadrature technique developed by Borinsky, we carried out numerical integrations for 1000 periods for each loop order from 3 to 15. The resulting histogram displays an interesting distribution at high loop orders, which we inferred with good empirical accuracy as being a shifted gamma distribution.

We also evaluated the contribution of period integrals to the loop amplitude and found a factorially divergent dependence on the loop number at the orders considered. The quantity computed is proportional to the contribution of primitive graphs to the beta function, which is believed to be the dominant part at high loop orders.

The factorial growth observed seems to corroborate the behaviour conjectured by Panzer and Borinsky of the constant that normalizes to 1 the expected value of a hypothetical limiting distribution of Feynman integrals.

Additionally, we show data that empirically supports the correlation between Hepp's bound and period discussed in a paper by Panzer.

Contents

Chapter 1

Introduction

Quantum field theory (QFT) is the basic tool to understand the subatomic world. In order to make predictions for any collider experiment, such as those carried out at CERN's Large Hadron Collider (LHC), it is necessary to evaluate the cross-sections of a particular scattering process by computing the *scattering amplitude* [29]. However, computing quantities such as the scattering amplitude directly from a QFT is often impossible. Therefore, perturbation theory is employed to compute approximate expressions; take for instance the following expansion of the scattering amplitude in terms of the parameter $\hbar$:

$$A(\hbar) = \sum_{L=0}^{\infty} A_L \hbar^L. \tag{1.1}$$

The problem of computing *loop amplitudes* A_L can be mapped to the one of evaluating mathematical objects called *Feynman integrals*, by using a set of *Feynman rules* that allow us to write the sum above as a sum over *Feynman graphs* G, which are a collection of edges $E(G)$, vertices $V(G)$, and external edges $Ext(G)$. Furthermore, the Feynman graphs considered in the sum above need to be *one particle irreducible* (1PI), where a 1PI graph is a graph that is connected and stays so upon deletion of one edge. Unless explicitly stated otherwise, all the graphs will be considered to be 1PI.

This amounts to the following expression for the loop amplitude:

$$A_L = \sum_{L(G)=L} \frac{I_G}{|\mathrm{Aut}(G)|}, \tag{1.2}$$

where the sum runs over all the Feynman graphs G that have loops $L(G) = L$, and $\mathrm{Aut}(G)$, sometimes denoted $\mathrm{Sym}(G)$ in the literature, is called symmetry factor of the Feynman diagram, and it is the order of the group of automorphisms of the graph. The coefficients I_G are the Feynman integrals mentioned above, and they are the starting point of this thesis. Their explicit form is shown in eq. (1.8) and it will be discussed in detail below. They are not only relevant for phenomenological predictions, but play a role in the computation of renormalisation group functions [36], and recently they have been used for perturbative gravity calculations [1]. Feynman integrals also have applications outside QFT, as they have shown to be relevant for various branches of pure mathematics [14, 13, 9].

However, Feynman integrals are hard to evaluate analytically and numerically, which poses a significant bottleneck for testing predictions and analysing results of scattering experiments [20]. The analytical evaluation of Feynman integrals requires the development of advanced mathematical techniques, the most powerful of which currently is the *differential equation method* [25, 30]. Other possible methods are reported in books such as [39, 33]. Analytical methods can fail in some cases, especially if the integrand becomes very complicated, and in these cases approaches based on numerical integration are the only feasible option.

The most popular numerical method for evaluating Feynman integrals is the *sector decomposition* approach. This was first introduced by Hepp [21] and then used for Feynman integral evaluation by Binoth and Heinrich [4]. Lately, a new numerical method based on *tropical geometry* [22] by Borinsky [7] showed promising results in the numerical evaluation of Feynman integrals up to order 17. This powerful numerical method is related to the *geometric sector decomposition* approach [23], which is the result of using the sector decomposition on a geometric footing.

Borinsky's tropical sampling algorithm makes use of tropical geometry and *convex geometry*, more precisely of *polytopes*, in a broader context than Feynman integrals. But, from chapter 3 of this thesis onwards, we will mostly focus on a specific type of Feynman integral called *period* and labelled $\mathcal{P}_G$, and applying Borinsky's method to the specific case of period integrals does not require any notion of convex geometry. The Feynman graphs associated with period integrals are primitive, i.e. free of subdivergences, and log-divergent, and we will call them *p-log*, as we will see in definition 3.1.

Periods are themselves ubiquitous in theoretical physics, frequently appearing in computations of renormalisation group functions [36], and operator product expansions [18]. These types of integrals are also studied in various branches of pure mathematics [11, 12, 14, 9]. It is conjectured [24] that in the large order regime period integrals give the most contribution to the β function of a ϕ^4 theory, a particular QFT that will be introduced below. There are many implications of this conjecture, mainly that the biggest contribution to the divergence of the β function in the large order regime comes from *instantons*, semiclassical field configurations, rather than *renormalons* [17], which are outcomes of renormalisation of certain types of Feynman graphs at high order.

The main goal of this thesis is to empirically study the large order behaviour mentioned for the loop amplitude restricted to p-log graphs of the ϕ^4 theory, in formulas:

$$A_L^{\text{p-log}} = \sum_{\substack{L(G)=L \\ G \text{ is p-log}}} \frac{\mathcal{P}_G}{|\text{Aut}(G)|}. \tag{1.3}$$

The graphs that are omitted from this sum, if compared with the one of the full loop amplitude in eq. (1.2), are the one that are not logarithmically divergent or have subdivergences (non-primitive). Most of the non-primitive graphs, either have *tadpoles* or *bubbles*, sometimes called *double edges*. These types of subdivergences are represented in fig. 1.1. For a ϕ^4 theory, the relative number of graphs that are not primitive nor have a tadpole nor a double edge scales like $1/L^2$, where L is the loop order [8]. Thus, $A_L^{\text{p-log}}$ does not converge to the value of the loop amplitude, but it still gives valuable information about the large order behaviour of the ϕ^4 theory. Furthermore, in [24, Eq. (B1)] it is shown that $A_L^{\text{p-log}}$ defined above is proportional to the contribution of primitive graphs to the β function of the ϕ^4 theory. Thus, the conjecture mentioned in the previous paragraph states, up to some prefactors:

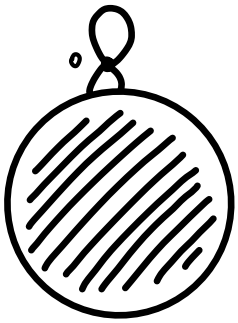
$$\beta_L^{\text{MS}} \sim A_L^{\text{p-log}} \quad \text{for} \quad L \gg 1, \tag{1.4}$$

where MS indicates the renormalisation scheme used.

The values of $A_L^{\text{p-log}}$ showed a factorial growth for the loop orders considered, that is $L < 16$, which seems to replicate the analytically expected behaviour for large orders [24, 17]. In particular, the behaviour inferred for the amplitude was:

$$A_L^{\text{p-log}} \sim B L^A C^{L+3} \Gamma(L+3), \tag{1.5}$$

where the values of parameters A, B, C can be found in eq. (4.32).



(a) Example of a tadpole on the vertex 0.

(b) Example of a bubble between the vertices 1 and 2

Figure 1.1: In fig. a and b the two main instances where the sub-divergences occur: Tadpoles and double edges.

In order to numerically estimate the value of (1.3), we implemented an algorithm, outlined in chapter 4, that randomly generates p-log graphs at a given loop order. The numerical computations relevant for the statistical analysis were conducted up to order 15 for 1000 integrals for each order. An interesting result, discussed in section 4.2, is the shape of distribution of period integrals obtained. This was empirically shown to be:

$$f(x; A, B, x_0) = \frac{\beta^\alpha}{\Gamma(\alpha)}(x - x_0)^{\alpha-1} e^{-\beta(x-x_0)}, \tag{1.6}$$

where the expression for the parameters α, β and x_0 for 15 loops can be found in eq. (4.15).

Finally, we also analysed the relation between the period and *Hepp's bound* $\mathcal{H}_G$, an approximated version of the period that bounds $\mathcal{P}_G$ from above. Its explicit computation is a crucial step the tropical sampling algorithm. Recent work [28] showed an interesting correlation between period and Hepp's bound, which we empirically showed to hold up to 15 orders.

1.1 Feynman graphs and Feynman integrals

Let us now discuss the expression in eq. (1.2) in more detail.

First, we remind the reader the sum in eq. (1.2) runs over the graphs G that have $L(G) = L$, where $L(G)$ is the number of loops of the graph G, i.e.

$$L(G) = |E(G)| - |V(G)| + C(G) \tag{1.7}$$

according to *Euler's formula*, see for instance [16, Theorem 4.2.9], with $C(G)$ being the number of the connected components of the graph G. We will consider only connected graphs, unless explicitly stated otherwise, thus $C(G) = 1$.

Secondly, let us explicitly give the expression for a Feynman integral I_G in a D dimensional Euclidean spacetime:

$$I_G = \prod_{e \in E} \int_{\mathbb{R}^D} \frac{d^D k_e}{\pi^{D/2}} \frac{1}{(k_e^2 + m_e^2)^{\nu_e}} \prod_{v \in V \setminus \{v_0\}} \delta^{(D)} \left(\sum_{e' \dashv v} p_{e'} \right), \tag{1.8}$$

where m_e are the masses and p_e the momenta assigned to each edge or external leg e. m_e is a positive real parameter, whereas p_e is a D dimensional vector with real components. Specifically, we will denote q_e as the external momenta and k_e as the internal ones, in formulas

$$p_e(q_e, k_e) = \begin{cases} q_e & \text{if } e \in Ext, \\ k_e & \text{if } e \in E. \end{cases} \tag{1.9}$$

It is important to note that I_G does not depend on the arbitrary direction chosen for the internal momenta k_e in (1.8). The product of delta functions for each vertex v arises due to momentum conservation of the momenta corresponding to the edges e' that meet in v, denoted as $e' \dashv v$. One arbitrary vertex v_0 has been omitted from this product as an overall external momentum conservation $\delta^{(D)}\left(\sum_{e\in Ext} q_e\right)$ has been factored out. The parameters ν_e are called *weights* or *indices*, and they are the power at which each propagator $1/(k_e^2 + m_e^2)$ occurs. In [34] it was shown that Feynman integrals are meromorphic functions in the variables ν_e and D. Sometimes we are interested in evaluating I_G for values of the indices and spacetime dimensions that make the integral (1.8) divergent. In order to deal with finite quantities, we can employ a technique called *dimensional regularization* [37] that makes the resulting expression convergent. This is achieved by analytically continuing the parameter $D = 4 - 2\epsilon$, while keeping $\epsilon > 0$. This condition is enough to ensure convergence of the period integrals that will be dealt with in this thesis.

The integral I_G in the form reported in eq. (1.8) is not well-suited for a numerical integration. Thus, in chapter 2 we shall manipulate its expression to obtain the *Schwinger parametric representation* which is more convenient for Feynman integral evaluation [27]. From this representation we will then derive the *projective representation*, that thanks to *projective geometry* [31, 2] will allow us to manifestly show some interesting geometric properties of the integral. This last representation will be the starting point for tackling the problem of numerical integration.

In order to numerically compute integrals like (1.8) a Monte Carlo integration is a viable and advantageous choice given that the integrand has dimension $|E| - 1$, and the number of edges $|E|$ is usually large. However, when integrating these type of integrals with probabilistic methods, the choice of probability density measure is critical to get finite results for both the numerical value of I_G and its uncertainty. These type of integrals often have poles in the integration domain, which make the variance infinite, but, by changing the probability measure so that the sampling is more concentrated around the poles, one can get finite results. Such a transformation can be performed by using techniques inspired by tropical geometry, as it is discussed in [7] and in chapter 3.

As already mentioned, chapter 4 focuses on estimating the value of $A_L^{\text{p-log}}$ for $L < 16$ for a ϕ^4 QFT, which has the Lagrangian

$$\mathcal{L} = \frac{1}{2}\partial_\mu\phi\partial^\mu\phi - \frac{1}{2}m^2\phi^2 - \frac{\lambda}{4!}\phi^4, \tag{1.10}$$

where λ is the *coupling constant* of the QFT. The vertices of a Feynman graph of this theory have *valency* 4, meaning that each vertex will have 4 edges incident to it. The choice of ϕ^4 theory in 4 spacetime dimensions is due to the large amount of existing research and data at high loop orders.

Chapter 2

Schwinger parametrisation and projective integrals

2.1 Schwinger parametrisation

The form of the Feynman integral in (1.8) is usually the first one derived and presented in many textbooks of quantum field theory. However, the *Schwinger parameter representation* of the Feynman integral, see textbooks such as [29], is a starting point for many modern applications [27]. This representation will be derived from the one in equation (1.8). It reads:

$$I_G(q,m) = \prod_{e\in E}\int_0^\infty \frac{\mathrm{d}x_e\; x_e^{\nu_e-1}}{\Gamma(\nu_e)}\,\frac{e^{-\phi/\psi}}{\psi^{D/2}}, \tag{2.1}$$

where ψ and ϕ are polynomials related to the Feynman graph G. They are sometimes also denoted as $\mathcal{U}$ and $\mathcal{F}$, and are called *Symanzik* or *Kirkoff polynomials* in the literature [5]. To derive their explicit expression, together with (2.1), we shall start with the definition of the Γ function

$$\Gamma(\nu) = \int_0^\infty \mathrm{d}x\, x^{\nu-1}e^{-x}.$$

Then, we can perform the scaling $x \to Ax$, which gives the result

$$\frac{1}{A^\nu} = \frac{1}{\Gamma(\nu)}\int_0^\infty \mathrm{d}x\, x^{\nu-1}e^{-Ax} \quad (\mathrm{Re}(\nu), A>0), \tag{2.2}$$

this is called the *Schwinger trick*. We can make use of this expression to manipulate (1.8) by

$$\frac{1}{(k_e^2+m_e^2)^{\nu_e}} = \frac{1}{\Gamma(\nu_e)}\int_0^\infty \mathrm{d}x_e\, x_e^{\nu_e-1}e^{-(k_e^2+m_e^2)x_e}.$$

We can also use the Fourier representation of the delta function

$$\delta^{(D)}(k) = \int_{\mathbb{R}^D}\frac{\mathrm{d}^D y}{(2\pi)^D}e^{iky},$$

to write the integral (1.8):

$$\begin{aligned} I_G(q,m) = \prod_{e\in E}\int_0^\infty \frac{\mathrm{d}x_e\, x_e^{\nu_e-1}}{\Gamma(\nu_e)} \quad \left(\prod_{v\in V\setminus\{v_0\}}\int_{\mathbb{R}^D}\frac{\mathrm{d}^D y_v}{(4\pi)^{D/2}}\right)\times \\ \prod_{e\in E}\int_{\mathbb{R}^D}\frac{\mathrm{d}^D k_e}{\pi^{D/2}}\exp\left(-\sum_{e\in E}x_e(m_e^2+k_e^2)+i\sum_{v\in V\setminus\{v_0\}}y_v\left(\sum_{e'\dashv v}p_{e'}\right)\right). \end{aligned} \tag{2.3}$$

This integral is a multidimensional Gaussian integral (see eq. (A.6) in appendix A.1) in the variables k_e and y_v.

As stated in section 1.1, it is convenient to pick an arbitrary orientation for the internal edges E, even if the final value of I_G will not depend on the orientation chosen. Having chosen an orientation, we can define the *incidence matrix*,

$$\mathcal{E}_{e,v} = \begin{cases} -1 & \text{if the edge } e \text{ leaves the vertex } v \\ +1 & \text{if the edge } e \text{ enters the vertex } v \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

Using this definition, we can manipulate the imaginary part of the exponent of (2.3). We can separate the sum over the momenta incoming in the vertex v as a sum over the momenta q_e corresponding to external edges and the momenta k_e corresponding to internal edges. In formulas:

$$\sum_v y_v \left(\sum_{e' \dashv v} p_{e'} \right) = \sum_v y_v \left(\sum_{e \in Ext_v} q_e + \sum_{e \in E_v} \mathcal{E}_{e,v}\, k_e \right),$$

where E_v and Ext_v are the set of all the internal and external edges that meet in the vertex v. We can now define the vectors $\mathbf{k} = (k_e)_{e\in E} \in (\mathbb{R}^D)^{|E|}$, $\mathbf{y} = (y_v)_{v\in V\setminus\{v_0\}} \in (\mathbb{R}^D)^{(|V|-1)}$ and the vector

$$\mathbf{q} = \begin{cases} q_e \text{ if } e \in E^{ext} \\ 0 \text{ otherwise,} \end{cases}$$

in such a way that $\mathbf{q} \in \mathbb{R}^{D(|V|-1)}$. We can also define the following matrices to write the exponent of the Gaussian integration in matrix form: The diagonal matrix

$$\Lambda = \operatorname{diag}(x_1, \dots, x_{|E|}), \tag{2.5}$$

and the weighted *Laplacian matrix* $\widehat{\mathcal{L}}$

$$\widehat{\mathcal{L}} = \widetilde{\mathcal{E}}^T \Lambda^{-1} \widetilde{\mathcal{E}} \tag{2.6}$$

where $\widetilde{\mathcal{E}}$ is the incidence matrix $\mathcal{E}$ defined in (2.4) stripped of the column related to the vertex v_0.

Now, as in any Gaussian integration, we need to complete the square, similarly to what is done in eq. (A.5):

$$\begin{aligned} -\mathbf{k}^T\Lambda\mathbf{k} + i\mathbf{k}\widetilde{\mathcal{E}}\mathbf{y} + i\mathbf{q}\widehat{\mathcal{L}}^{-1}\widehat{\mathcal{L}}\mathbf{y} - \frac{i\mathbf{y}^T}{2}\widetilde{\mathcal{E}}^T\Lambda^{-1}\widetilde{\mathcal{E}}\frac{i\mathbf{y}}{2} - \frac{\mathbf{y}^T}{2}\widehat{\mathcal{L}}^{-1}\frac{\mathbf{y}}{2} + \mathbf{q}^T\widehat{\mathcal{L}}^{-1}\mathbf{q} - \mathbf{q}^T\widehat{\mathcal{L}}^{-1}\mathbf{q} - \sum_{e\in E} x_e m_e^2 = \\ = -\left(\mathbf{k} - \frac{i}{2}\Lambda^{-1}\widetilde{\mathcal{E}}\mathbf{y}\right)^T \Lambda \left(\mathbf{k} - \frac{i}{2}\Lambda^{-1}\widetilde{\mathcal{E}}\mathbf{y}\right) - \left(\frac{\mathbf{y}}{2} - i\widehat{\mathcal{L}}^{-1}\mathbf{q}\right)^T \widehat{\mathcal{L}} \left(\frac{\mathbf{y}}{2} - i\widehat{\mathcal{L}}^{-1}\mathbf{q}\right) - \sum_{e\in E} x_e m_e^2. \end{aligned}$$

Then, it is possible to explicitly compute the integral using eq. (A.6), which gives the result in (2.1).

The explicit expression of the Kirkoff polynomials is then

$$\begin{aligned} \psi &= \left(\prod_e x_e\right) \det \widehat{\mathcal{L}} \\ \phi &= \left(\sum_{e\in E} x_e m_e^2 + \mathbf{q}^T\widehat{\mathcal{L}}^{-1}\mathbf{q}\right)\psi. \end{aligned} \tag{2.7}$$

The expression for the ψ and ϕ polynomials can be further simplified when dealing with connected graphs by using *matrix-tree theorem*, see [27, Chapter 2],

$$\begin{aligned} \psi &= \sum_{T} \prod_{e \notin T} x_e \\ \phi &= \psi \sum_{e \in E} x_e m_e^2 + \sum_{F} p(F)^2 \prod_{e \notin F} x_e \end{aligned} \tag{2.8}$$

where the sum runs respectively over all the connected subgraphs $T \subset G$ that have the same vertices as G and do not have any loops, called *spanning trees*, and over all the spanning subgraphs $F \subset G$ that have 2-connected components, called *spanning 2-forests*. Let us note that the polynomial ψ depends only on the parameters x_e, thus it is a graph invariant, whereas ϕ contains all the information about the kinematics of the system.

Even if the expressions in eq. (2.8) are often used for analytical calculations, the matrix representation of the polynomials in (2.7) is the most suitable for numerical computations, which is the central focus of this thesis. In fact, as stated in [7, Section 7.1], the number of spanning trees grows exponentially with the number of vertices V, thus making a numerical computation impossible at high loop orders. On the other hand, the computations of matrix determinants in eq. (2.7) can be done using the Cholesky decomposition, which for a matrix of size $(V-1)\times(V-1)$ takes $\sim \mathcal{O}(V^3)$ time. Further improvements can be achieved by computing the decomposition making use of a nearly-linear time algorithm outlined in [35].

From the expressions above, it is possible to see that they are both homogeneous functions in the variables x_e, which means

$$\psi(\alpha x_1, \dots, \alpha x_{|E|}) = \alpha^{\deg(\psi)} \psi(x_1, \dots, x_{|E|}) \text{ where } \alpha \in \mathbb{R}. \tag{2.9}$$

$\deg(\psi)$ is called *degree* of the polynomial ψ. An explicit computation shows that the two Kirkoff polynomials have degree

$$\deg(\psi) = L \tag{2.10}$$

$$\deg(\phi) = L + 1, \tag{2.11}$$

where L is the loop number of G.

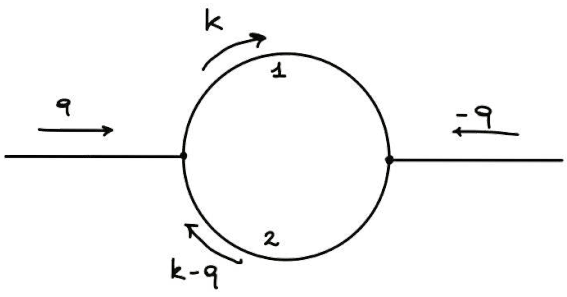


Figure 2.1: Feynman graph associated to the integral in equation (2.12).

Example 2.1. Let us consider the diagram in fig. 2.1. The parameter representation of the integral related to this graph is:

$$I(\nu_1, \nu_2) = \int_{x_1, x_2 \geq 0} \frac{\mathrm{d}x_1 \mathrm{d}x_2}{\Gamma(x_1)\Gamma(x_2)} x_1^{\nu_1 - 1} x_2^{\nu_2 - 1} \frac{e^{-\phi/\psi}}{\psi^{D/2}}. \tag{2.12}$$

(a) Spanning trees (b) Spanning 2-forests

Figure 2.2: Spanning trees and spanning 2-forests of the graph in fig. 2.1

We can now use (2.8) to write explicitly the graph polynomials. Let us start by considering all the spanning trees T of the graph G. These are represented in fig. 2.2b. The expression for ψ will then be

$$\psi = x_2 + x_1.$$

Similarly, we shall now consider the spanning two forests, reported in figure 2.2a. Then, ϕ turns out being

$$\phi = q^2 x_1 x_2 + (x_1 + x_2)(x_1 m_1^2 + x_2 m_2^2).$$

Furthermore, it is possible to see that these are in fact homogeneous polynomials of degree 1 and 2 respectively, in accordance with (2.11).

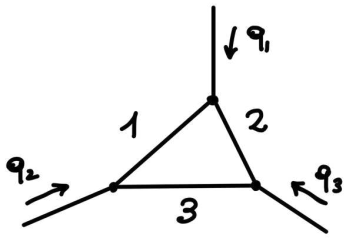


Figure 2.3: Example of Feynman diagram with three incoming particles.

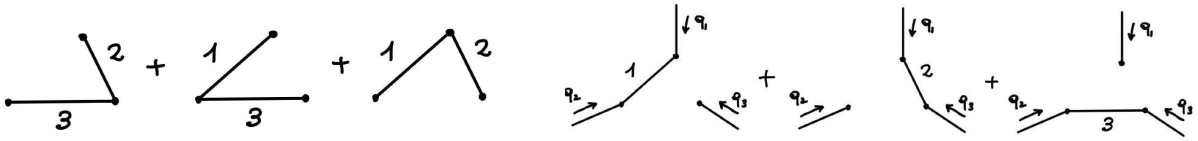
Example 2.2. A slightly harder problem is the one in fig. 2.3. Considering the spanning trees and spanning 2-forests in fig. 2.4, the two polynomials turn out being:

$$\psi = x_2 x_3 + x_1 x_2 + x_1 x_3$$

$$\phi = -\left(q_1(q_2+q_3)x_2x_3 + q_2(q_1+q_3)x_1x_3 + q_3(q_1+q_2)x_1x_2\right) + \psi\left(\sum_{i=1}^{3} x_i m_i^2\right) =$$

$$= q_1^2 x_2 x_3 + q_2^2 x_1 x_3 + q_3^2 x_1 x_2 + \psi\left(\sum_{i=1}^{3} x_i m_i^2\right),$$

where to get the final expression for ϕ we used $q_1 + q_2 + q_3 = 0$, which holds due to momentum conservation.



(a) Spanning trees (b) Spanning 2-forests

Figure 2.4: Spanning trees and 2 spanning forests of the graph in fig. 2.3

2.2 Projective Feynman integral

The parametric representation can be manipulated further to make the Feynman integral better suited for an explicit evaluation. This amounts to perform a change of variable so that the exponential term $e^{-\phi/\psi}$ disappears from (2.1). The form that we will obtain by performing such a transformation, does not manifestly show all the geometric properties of the integral: We will show that the integral is independent on the specific change of variable performed. This can be done by using projective geometry and expressing the integral (2.1) as an integral over the projective space, that will be defined below.

As stated in equation (2.11), the two polynomials ψ and ϕ are homogeneous of degree L and $L+1$ respectively. To do so, we shall start by inserting a delta function in the integral by using the property

$$1 = \int_{-\infty}^{\infty} \mathrm{d}\lambda\, \delta(\lambda - H(x)) = \int_{0}^{\infty} \mathrm{d}\lambda\, \delta(\lambda - H(x)),$$

where $H(x)$ is defined to be any hyperplane in x_e space, i.e. $H(x) = \sum_e H_e x_e$ with $H_e \geq 0$ not all zero. To proceed further we can perform the change of variables $x_e \to \lambda x_e$ where λ is a real parameter. The integral thus becomes

$$I_G = \prod_e \int_0^\infty \frac{\mathrm{d}x_e\, x_e^{\nu_e - 1}}{\Gamma(\nu_e)} \frac{\delta(1 - H(x))}{\psi^{D/2}} \int_0^\infty \mathrm{d}\lambda\, e^{-\lambda\phi/\psi} \lambda^{\omega - 1}, \tag{2.13}$$

where ω is the *superficial degree of ultraviolet divergence* of the graph, defined as

$$\omega(G) = \sum_{e \in E(G)} \nu_e - L(G)\frac{D}{2}, \tag{2.14}$$

where this definition can be generalised to subgraphs $\gamma \subset G$ by considering only the edges $E(\gamma)$ and loops $L(\gamma)$ of the subgraph. The superficial degree of divergence captures the behaviour of the Feynman integral in the UV region, that is when $k_e \to \infty$ for $e \in G$. In particular, a Feynman integral that has all $m_e > 0$ is superficially divergent if it has $\omega(G) \leq 0$. It is superficially convergent if $\omega(G) > 0$, and all the 1PI subgraphs $\gamma \subset G$ have $\omega(\gamma) > 0$ [38, 3]. We will return on the issue of convergence in the next chapter.

To compute the integration in λ in (2.13), we can perform the rescaling $\lambda \to \lambda\phi/\psi$, which gives the result

$$\left(\frac{\psi}{\phi}\right)^{\omega} \int_0^\infty \mathrm{d}\lambda\, \lambda^{\omega-1} e^{-\lambda} = \left(\frac{\psi}{\phi}\right)^{\omega} \Gamma(\omega).$$

Overall, the integral will then be

$$I_G = \frac{\Gamma(\omega)}{\prod_e \Gamma(\nu_e)} \int_{\mathbb{R}_{\geq 0}^{|E|}} \prod_e \frac{\mathrm{d}x_e}{x_e}\, \delta(1 - H(x_e))\ \frac{\prod x_e^{\nu_e}}{\psi^{D/2}} \left(\frac{\psi}{\phi}\right)^{\omega}. \tag{2.15}$$

It is possible to simplify this integral further and show explicitly that it is not dependent on the choice of hyperplane H by using the properties of projective geometry, for further details see [31, 2, 10]. In particular, the real projective space $\mathbb{RP}^n$ is given by taking the quotient of $\mathbb{R}^{n+1} \setminus \{0\}$ under the following equivalence relation:

$$(x_0, \ldots, x_n) \sim (\lambda x_0, \ldots, \lambda x_n), \text{ for some } \lambda \in \mathbb{R} \setminus \{0\}. \tag{2.16}$$

Points of $\mathbb{RP}^n$ can be represented as $[x_0 : \cdots : x_n]$ using a coordinate system called *homogeneous coordinates*. Let us also define the positive orthant of real projective space:

$$\mathbb{RP}^n_+ = \left\{ [x_0, \ldots, x_n] \in \mathbb{RP}^n \;\middle|\; x_i > 0 \text{ for all } i = 0, \ldots, n \right\}. \tag{2.17}$$

We can manipulate eq. (2.15) so that it becomes an integral over projective space. Let us start by noticing that the integrand

$$f(x_1,\dots,x_{|E|}) = \frac{\prod x_e^{\nu_e-1}}{\psi^{D/2}}\left(\frac{\psi}{\phi}\right)^{\omega}$$

is a homogeneous function in x_e, as in eq. (2.9), of degree $(-|E|)$. Moreover, the volume form $\mathrm{d}^{|E|}x$ is homogeneous of degree $|E|$. Thus, the integral (2.15) will overall be invariant under a rescaling $x_e \to \alpha x_e$, where α is a constant parameter. This means that the integration variables do not distinguish between a point $(x_1,\dots,x_{|E|})$ and a rescaled point $(\alpha x_1,\dots,\alpha x_{|E|})$. Using this and the fact that we are integrating over the simplex

$$\Delta = \left\{ (x_1,\dots,x_{|E|}) \in \mathbb{R}^{|E|} \;\middle|\; \sum_{e=1}^{|E|} x_e = 1,\ x_e \geq 0 \right\},$$

we can identify the coordinates in the integral (2.15) as homogeneous coordinates over $\mathbb{RP}_+^{|E|-1}$. In fact, by the equivalence relation defined above, when considering a point in projective space with homogeneous coordinates $[x_0,\dots,x_n] \in \mathbb{RP}_+^{|E|-1}$, it is always possible to rescale its homogeneous coordinates so that their sum is equal to 1, i.e. $\sum_{e=1}^{|E|} x_e = 1$. Ultimately, the integral becomes:

$$I_G = \frac{\Gamma(\omega)}{\prod_e \Gamma(\nu_e)} \int_{\mathbb{RP}_+^{|E|-1}} \Omega \ \frac{\prod_e x_e^{\nu_e}}{\psi^{D/2}}\left(\frac{\psi}{\phi}\right)^{\omega}. \tag{2.18}$$

Here we defined the canonical form over the projective space

$$\Omega = \sum_{e=1}^{|E|} (-1)^{|E|-e} \frac{\mathrm{d}x_1}{x_1} \wedge \cdots \wedge \widehat{\frac{\mathrm{d}x_e}{x_e}} \wedge \cdots \wedge \frac{\mathrm{d}x_{|E|}}{x_{|E|}}, \tag{2.19}$$

where $\widehat{\mathrm{d}x}$ indicates the missing differential. The differential form Ω is well-defined over $\mathbb{RP}_+^{|E|-1}$ as it is homogeneous of degree 0 by construction, hence, it is invariant under the scaling $x_e \to \alpha x_e$ for $\alpha \in \mathbb{R}$. In fact, Ω vanishes when it is integrated over any line that passes through the origin in $\mathbb{R}^{|E|}$. Equivalently, we can define the vector field

$$\xi = \sum_{e=1}^{|E|} x_e \partial_e$$

that represents a line passing through the origin, and then verify that the following relation holds

$$\xi \llcorner \Omega = 0,$$

where $\llcorner$ denotes the *interior product*.

The main point behind the integral representation in equation (2.18) is the fact that it is manifestly independent of the choice of hyperplane $H(x_e)$. This means that we can choose any affine chart of Schwinger parameters x_e and project the integral onto $\mathbb{R}_+^{|E|-1}$, where

$$\mathbb{R}_+^{|E|-1} = \left\{ (x_1,\dots,x_{|E|-1}) \in \mathbb{R}^{|E|-1} \;\middle|\; x_i > 0 \text{ for all } i = 1,\dots,|E|-1 \right\}.$$

This is in complete analogy to what happens in gauge theories: Fixing a coordinate chart is like fixing a gauge. For instance, choosing an affine chart such that $x_{|E|} = 1$, gives the result:

$$I_G = \frac{\Gamma(\omega)}{\prod_e \Gamma(\nu_e)} \int_{\mathbb{R}_+^{|E|-1}} \left(\prod_{e=1}^{|E|-1} \frac{\mathrm{d}x_e}{x_e} \right) \ \frac{\prod_e x_e^{\nu_e}}{\psi^{D/2}}\left(\frac{\psi}{\phi}\right)^{\omega}. \tag{2.20}$$

This important result is often referred to as *Cheng-Wu* theorem in the literature.

The projective form of the Feynman integral will be the starting point to tackle the problem of numerical integration, as it encapsulates all the geometrical properties of I_G. In fact, with this form of the integral, the numerical evaluation of the integral is reduced to the construction of a clever way to map random points in a bounded subset of $\mathbb{R}^{|E|-1}$ to points in $\mathbb{RP}^{|E|-1}_+$. This problem is not trivial and there are many subtleties, which will be discussed in detail in the next chapter.

2.2.1 An illustrative example

Let us now consider a simple example of a projective integral. Consider the integral representation of the Euler beta function

$$B(\nu_1,\nu_2) = \int_0^1 \mathrm{d}x\, x^{\nu_1-1}(1-x)^{\nu_2-1} = \frac{\Gamma(\nu_1)\Gamma(\nu_2)}{\Gamma(\nu_1+\nu_2)}. \tag{2.21}$$

By adding a delta function using the property

$$1 = \int_0^\infty \mathrm{d}y\, \delta(1-(x+y)) \tag{2.22}$$

similarly to what was done in the previous section, the integral becomes

$$B(\nu_1,\nu_2) = \int_{\mathbb{R}^2_+} \delta(1-(x+y))\frac{\mathrm{d}x}{x}\frac{\mathrm{d}y}{y}\, x^{\nu_1}y^{\nu_2} = \int_\Delta \frac{\mathrm{d}x}{x}\frac{\mathrm{d}y}{y}\, x^{\nu_1}y^{\nu_2}, \tag{2.23}$$

where Δ is the simplex given by

$$\Delta = \left\{ (x,y)\in\mathbb{R}^2 \,\middle|\, x,y\geq 0,\ x+y=1 \right\}. \tag{2.24}$$

This integral is in fact, up to a normalisation factor, the Feynman integral for the bubble graph in example 2.1 with $D=4$ spacetime dimensions and vanishing masses, $m_1=m_2=0$.

Let us now represent the previous integral as an integral over projective space. In fact, in this form it would be manifest that the integral does not depend on the choice of plane $x+y=1$. Thus, we need both the integrand and the volume form to be homogeneous of degree 0 in the integration variables x and y. To obtain such an expression, we can perform the rescaling $x\to\frac{x}{x+y}$ and $y\to\frac{y}{x+y}$ where (x,y) now are to be taken as homogeneous coordinates for a point $[x,y]\in\mathbb{RP}^1$. The integral then becomes

$$\int_{\widetilde{\Delta}} \left(\frac{\mathrm{d}x}{x}-\frac{\mathrm{d}y}{y}\right)\frac{x^{\nu_1}y^{\nu_2}}{(x+y)^{\nu_1+\nu_2}},$$

where, after performing the change of variables, the region of integration becomes $\widetilde{\Delta}\subseteq\mathbb{RP}^1$ given by the simplex in projective space that has points $[x,y]$ that have homogeneous coordinates that are coordinates of a point in Δ, as shown in fig. 2.5, in symbols

$$[x,y]\in\widetilde{\Delta} \Leftrightarrow (x,y)\in\Delta.$$

The definition of Δ implies that $\widetilde{\Delta}=\mathbb{RP}^1_+$. To show this, we can use an argument identical to the one in the previous section, specifically that it is always possible to rescale a point represented by homogeneous coordinates $[x,y]\in\mathbb{RP}^1_+$ to one point whose homogeneous coordinates sum up to 1.

Finally, it is worth noticing that the top form over the integration domain $\mathbb{R}^2_+$ in eq. (2.23) becomes the canonical volume form in projective space defined in (2.19), i.e.

$$\Omega = \frac{\mathrm{d}x}{x} - \frac{\mathrm{d}y}{y}.$$

On the whole, the projective representation of the beta integral becomes:

$$B(\nu_1, \nu_2) = \int_{\mathbb{RP}_+} \Omega \frac{x^{\nu_1} y^{\nu_2}}{(x+y)^{\nu_1+\nu_2}}. \tag{2.25}$$

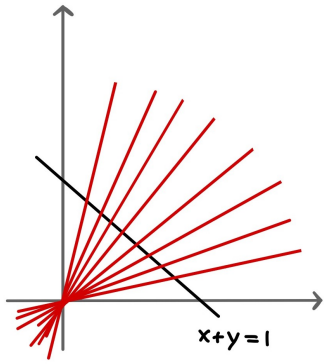


Figure 2.5: A visual representation of the one to one correspondence between the elements of Δ and the ones of $\mathbb{RP}_+$: The integration region of (2.23) is outlined in black, and in red some lines passing through the origin of $\mathbb{R}^2$ which correspond to elements of $\mathbb{RP}_+$.

Putting the mathematical formalism aside, the relevance of the projective integral (2.25) lies in the fact that it is evident that we can take any hyperplane in the variables x_1, x_2 and project the integral over the real line again. This allows us to get different representations of the integral (2.21) by only considering different affine charts. For instance, let us choose the lines $x = 1$ or $y = 1$. Considering the first case, the integral becomes

$$B(\nu_1, \nu_2) = \int_0^\infty \frac{\mathrm{d}x}{x} \frac{x^{\nu}}{(x+1)^{\nu_1+\nu_2}}.$$

Here the convenience of the projective integral is manifest. This integral form can be obtained from the one in (2.21) by performing a change of variables, but with the expression (2.25) the problem reduces to simply choosing an affine chart. In fact, the freedom in changing variables, thus giving different integral representations, gets completely absorbed by the properties of projective geometry.

Chapter 3

Monte Carlo integration

The aim of this chapter is to describe a method that numerically evaluates the integral I_G. As stated in the introduction, a Monte Carlo integration is in general more favourable than a deterministic method considering that the integral has dimension $|E| - 1$. In fact, runtime scales exponentially with the dimension of integration for non-probabilistic methods. This task can be very daunting as the measure of the projective integral (2.18) cannot be reduced to a probability measure, and even when the integral gets projected onto $\mathbb{R}^D$ the integrand can have poles in the integration domain, making a naive Monte Carlo sampling useless. However, if one samples points in the domain using an approach inspired by tropical geometry, one can get finite and consistent results. This approach is based on the *tropical approximation* of a polynomial; it is described in [7, 28] and it will be delineated in section 3.2.

Hereafter, we will only consider a specific class of graphs called *primitive logarithmically divergent* (p-log) graphs G.

Definition 3.1. A graph G is a p-log graph if:

- It has degree of divergence $\omega(G) = 0$;
- All the proper subgraphs of G are superficially convergent, i.e. $\omega(\gamma) > 0$ for all $\gamma \subset G$.

The graphs that respect the second condition are called primitive. The condition $\omega = 0$ implies that the integral associated to the graph is logarithmically divergent. The restriction to p-log graphs, amounts to the following change for the integral in eq. (2.18):

$$\mathcal{P}_G = \int_{\mathbb{RP}^{|E|-1}_+} \Omega \ \frac{\prod x_e^{\nu_e}}{\psi(\mathbf{x})^{D/2}}. \tag{3.1}$$

In the expression above we omitted the pre-factor $\frac{\Gamma(\omega)}{\prod_e \Gamma(\nu_e)}$. It is interesting to notice that the overall divergence in the UV region is completely captured by the prefactor $\Gamma(\omega)$. Thus, when perfoming the analytic continuation of the spacetime dimension to be $D = 4 - 2\epsilon$, mentioned in chapter 1, we only require $\epsilon > 0$. However, we will not consider the issues of logarithmic divergence of the p-log graphs and will only focus on the explicit evaluation of (3.1), while always making sure that the graph is subdivergence free.

As stated in the introduction, the integral in (3.1) is called period in the literature, and it is a graph invariant. One reason why these integrals are relevant, other than the ones mentioned in ch. 1, is that they are independent on the kinematics of the process as the dependence on the polynomial ϕ disappears completely in the integral (2.18). This simplifies greatly the problem of numerical evaluation.

3.1 Naive Monte Carlo integration

In this section we will review the basic working principles of Monte Carlo integration, see textbooks such as [19] for further details. Suppose we want to evaluate an integral over a bounded domain Σ

$$I = \int_\Sigma \mu\, f(\mathbf{x}),$$

where μ is a probability density measure, i.e.

$$\mu > 0 \text{ on } \Sigma, \text{ and } \int_\Sigma \mu = 1.$$

To numerically evaluate I we can approximate the integral with the average $G^{(N)}$ of the function $f(\mathbf{x})$ evaluated in N random points $\mathbf{x}^{(i)}$ sampled from the distribution μ, in formulas:

$$G^{(N)} = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^{(i)}). \tag{3.2}$$

In fact, it holds that:

$$\mathbb{E}[G^{(N)}] = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}[f(\mathbf{x}^{(i)})] = \int_\Sigma \mu\, f(\mathbf{x}) = I. \tag{3.3}$$

This means that we can use $G^{(N)}$ to approximate the integral I. The variance of the average $G^{(N)}$ is given by

$$\mathrm{Var}[G^{(N)}] = \frac{1}{N}\mathrm{Var}[f(\mathbf{x})] = \frac{1}{N}\int_\Sigma \mu \left(f(\mathbf{x}) - \mathbb{E}[G^{(N)}]\right)^2. \tag{3.4}$$

The variance in (3.4) can be numerically estimated using the unbiased sample variance

$$\sigma_I^2 = \frac{1}{N-1}\sum_{i=1}^{N} \left(f(\mathbf{x}^{(i)}) - G^{(N)}\right)^2 = \frac{\sigma_f^2}{N-1}. \tag{3.5}$$

where we defined σ_f as the standard deviation of the function f, which we can use as an estimator for the value of $\sqrt{\mathrm{Var}[f]}$.

From the equations above, we conclude that I can be numerically estimated by $G^{(N)}$ with an error given by $\sigma_I = \sqrt{\sigma_I^2}$. However, a naive Monte Carlo integration does not in general yield sensible results. For the Monte Carlo integration to correctly estimate I with a finite error, both the quantities $G^{(N)}$ and $\mathrm{Var}[G^{(N)}]$ need to be finite. In order for the variance in equation (3.4) to give finite results, $f(\mathbf{x})$ needs to be a square integrable function over the integration domain, i.e.

$$\int_\Sigma \mu\, |f(\mathbf{x})|^2 < \infty. \tag{3.6}$$

If this condition does not hold, the variance will be divergent, i.e. $\mathrm{Var}[G^{(N)}] = \infty$ by eq. (3.4). Consequently, the unbiased sample variance σ_I^2, used to estimate the error of the numerical integral, will give finite but non-convergent results.

This problem is particularly relevant for period integrals. In fact, performing a Monte Carlo integration as described on the integral in eq. (3.1) is not possible. This is because the differential form Ω is not normalisable, i.e.

$$\int_{\mathbb{RP}^{|E|-1}_{>0}} \Omega = \infty.$$

We can solve this problem by choosing an affine chart to project the integral onto the unbounded domain $\mathbb{R}_+^{|E|-1}$, see for instance eq. (2.20). Then, we can perform a change of variables so that the resulting integration domain is bounded, such as the hypercube $[0,1]^{|E|}$. A one dimensional example would be the map $x \to \frac{x}{1+x}$, where the unbounded domain $\mathbb{R}_+$ gets mapped onto $[0,1]$. Performing these steps we would successfully write the integral as an integration over a bounded domain with a measure $\prod_e \mathrm{d}x_e$ that is indeed a probability measure. However, in general the resulting numerical integration would still not give finite results. This is because the integrand can have poles in the integration domain, meaning that the condition in eq. (3.6) does not always hold.

We can further illustrate this problem by considering the toy-example discussed in the next subsection.

3.1.1 Monte Carlo integration failing

Let us consider the integral

$$I = \int_{[0,1]^2} \mathrm{d}x\mathrm{d}y\, h(x,y) = \int_{[0,1]^2} \mathrm{d}x\mathrm{d}y \frac{1}{x+y}. \tag{3.7}$$

This integration can be performed analytically, and it yields the result $I = 2\log 2 \simeq 1.382$. As outlined above, we try to estimate the value of I using the quantity:

$$G^{(N)}[h] = \frac{1}{N}\sum_{i=1}^{N} h(x^{(i)}, y^{(i)}) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{x^{(i)}+y^{(i)}} \tag{3.8}$$

where the points $(x^{(i)}, y^{(i)})$ are sampled from a uniform probability distribution μ in the square $[0,1]^2$, defined as

$$\mu = \begin{cases} 1\mathrm{d}x\mathrm{d}y \text{ if } (x,y) \in [0,1]^2, \\ 0 \text{ otherwise.} \end{cases} \tag{3.9}$$

In eq.(3.8) we made explicit the dependence on the function h to avoid any confusion in the rest of the section.

Performing a numerical integration by sampling from a uniform distribution over the integration domain, we get the results shown in fig. 3.1a. In the first figure the green crosses represent the results of the numerical integral $G^{(N)}[h]$ for values of N that range from 5 to 100 with increments of 1; it is possible to see that the result of the numerical integration fluctuate heavily around the analytical result, represented by a dashed red line, as N increases. Furthermore, if we look at the second and third figures, we can see that error of the integrand σ_h, represented by gray dots in the second graph, and the error of the integral σ_I, represented by blue dots in the third graph, both have values that are finite but do oscillate greatly, and neither parameter converge to any value. Furthermore, we can try to fit the error σ_I with the theoretical $1/\sqrt{N}$ behaviour outlined in eq. (3.5); we can then compare the result of the fit σ_h^{fit} with the mean of the errors in the second picture, that we will call $\overline{\sigma_h}$. The fit gives a result:

$$\sigma_h^{fit} = 1.33 \pm 0.12,$$

and it is represented in the third plot of fig. 3.1a by a dashed red line. Even if this result is compatible with the mean of the errors σ_h in the second graph, which is

$$\overline{\sigma_h} = 1.4 \pm 1.2,$$

by looking at the plot, we can see that the fit is practically worthless. The error of $\overline{\sigma_h}$ is given by the standard deviation of the data σ_h, and the fact that it is so relatively high in value tells

us that the data is very disperse, hinting at the fact that the variance of h is divergent. We can confirm this with the analytical computation. Using (3.4), we get

$$\mathrm{Var}[h(x,y)] = \frac{1}{N}\int_{[0,1]^2} \mathrm{d}x\mathrm{d}y \left(\frac{1}{x+y} - 2\log 2\right)^2 = +\infty,$$

given by the fact that $h(x,y)$ is not a square integrable function. Thus, the estimators σ_h and σ_I will not converge to any specific value.

These results can be compared to a second Monte Carlo integration of the function:

$$g(x,y) = \begin{cases} 1 \text{ if } x,y \geq 0 \text{ and } x^2+y^2 = 1 \\ 0 \text{ otherwise.} \end{cases} \tag{3.10}$$

The analytical integration gives the result:

$$J = \int_{[0,1]^2} \mathrm{d}x\mathrm{d}y\; g(x,y) = \frac{\pi}{4}. \tag{3.11}$$

As $g(x,y)$ is a square integrable function over the integration domain, its variance, computed with eq. (3.4), is finite:

$$\mathrm{Var}[g(x,y)] = \frac{\pi}{4} - \left(\frac{\pi}{4}\right)^2. \tag{3.12}$$

Let us now analyse the results of a numerical integration, reported in fig. 3.1b. In the first figure we can see that the points, which represent the value of the numerical estimator $G^{(N)}[g]$, do indeed converge to the analytical result $\pi/4$ as N increases in the same range as the previous integration. If we look at the second and third plot of fig. 3.1b, we can see that both the error of the integrand σ_g and the error of the inegral σ_J are now less disperse. In fact, the error σ_g does indeed oscillate around its theoretical value $\sqrt{\pi/4 - (\pi/4)^2}$, and we can see that the error of the Monte Carlo integral σ_J exhibits the $\frac{1}{\sqrt{N}}$ behaviour which is expected as of eq. (3.5). This is in contrast to what happens with the third graph of fig. 3.1a, where the error of the integral never stabilises. We can also perform the same analysis done previously: fitting the function $\sigma_g^{fit}/\sqrt{N}$ gives the result

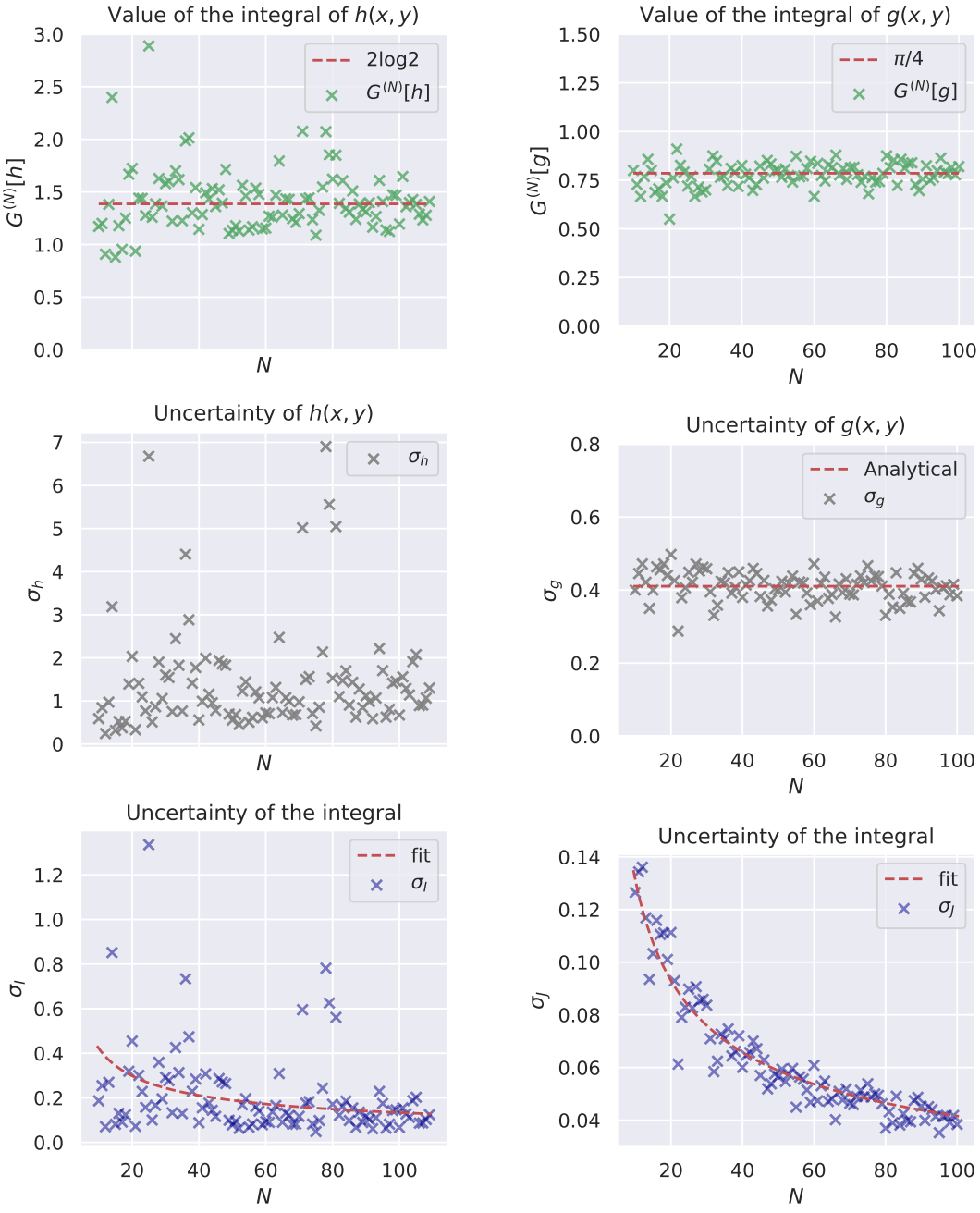
$$\sigma_g^{fit} = 0.4160 \pm 0.0045.$$

The resulting function in represented in the third plot in fig. 3.1b as a dashed red line. The parameters obtained from the fit are compatible with the mean of the values σ_g:

$$\overline{\sigma_g} = 0.410 \pm 0.039.$$

In fact the values σ_g^{fit} and $\overline{\sigma_g}$ are separated by about 1 standard deviation. Note how the standard deviation of the uncertanties σ_g is smaller than the previous example, due to the fact that $\mathrm{Var}[g]$ is finite.

A possible solution to the problem of having infinite variance due to poles in the integration domain is to change the probability distribution so that it cancels the divergence of the integrand. In fact, by performing such a transformation, the sampling will be more concentrated around the point where the integrand is divergent, as that is the region where the integral gets most of its contribution. Doing so is not a trivial task for integrals such as the one in eq. (2.18) and it will be discussed in the next section.



(a) Monte Carlo integration for the function $f(x, y) = \frac{1}{x+y}$.

(b) Monte Carlo integration that computes the area of a quarter circle.

Figure 3.1: In subfig. a we have the results for the function $h(x, y) = 1/(x + y)$ which is not square integrable. In particular, in the first row we can see the values of the numerical integral (green crosses), and the theoretical value (dashed red line), the second row reports the values of the uncertainty σ_h (gray crosses), and the third row the uncertainty of the integral σ_I (blue crosses) and a fit with the relation in (3.5). In subfig. b we can see the behaviour of same quantities for the square integrable function $g(x, y)$, defined in eq. (3.10).

3.2 Tropical approximation

The purpose of this section is to introduce the tropical approximation of a general multivariate polynomial p.

Consider a set of n variables $(x_1, \dots, x_n) = \mathbf{x}$ and a set of indices $(e_1, \dots, e_n) = \mathbf{e}$, then we can write a general multivariate polynomial as

$$p(\mathbf{x}) = \sum_{\mathbf{e}} c_{\mathbf{e}} \prod_{i=1}^{N} x_i^{e_i} = \sum_{\mathbf{e}} c_{\mathbf{e}} \mathbf{x}^{\mathbf{e}},$$

where the sum runs over all those indices such the complex coefficients $c_{\mathbf{e}} \neq 0$. We also defined the multi-index notation $\mathbf{x}^{\mathbf{e}} = \prod_{i=1}^{n} x_i^{e_i}$ as it was done in [7]. Let us now introduce the following *tropical approximation* for the polynomial p:

$$p^{\mathrm{tr}}(\mathbf{x}) = \max_{\mathbf{e}, c_{\mathbf{e}} \neq 0} \mathbf{x}^{\mathbf{e}}, \tag{3.13}$$

where the max is taken over the set of indices $\mathbf{e}$ such that $c_{\mathbf{e}} \neq 0$.

In the next section we will try to apply the tropical approximation to the period integral (3.1), in order to try to find a solution to the problem outlined in the previous section. To this purpose let us introduce the tropical approximation of the polynomial $\psi(\mathbf{x})$ in eq. (2.8):

$$\psi^{\mathrm{tr}}(\mathbf{x}) = \max_{T} \prod_{e \notin T} x_e, \tag{3.14}$$

where the max is taken over all the spanning trees of G. For instance, consider the graph polynomial in example 2.2, its tropical approximation would be

$$\psi^{\mathrm{tr}}(x_1, x_2, x_3) = \max(x_1 x_2, x_1 x_3, x_2 x_3).$$

From the definition of ψ^{tr}, it is possible to deduce the following relation:

$$\psi^{\mathrm{tr}}(\mathbf{x}) \leq \psi(\mathbf{x}). \tag{3.15}$$

3.3 Sector decomposition

We shall now use the tropical approximation introduced above to solve the problem of poles in the integration region outlined in section 3.1. Our aim is to use the properties of ψ^{tr} to "absorb" the singularities of ψ. Thanks to the explicit form of the tropical ψ^{tr} polynomial shown in (3.14), we will then be able to manipulate the integral further by splitting the integration region. This will allow us to carry out a Monte Carlo integration with finite and sensible results.

Let us now use ψ^{tr} to manipulate the expression of the period. The integral in eq. (3.1) factorises as:

$$\mathcal{P}_G = \int_{\mathbb{RP}^{|E|-1}_+} \Omega \ \frac{\prod x_e^{\nu_e}}{(\psi^{\mathrm{tr}}(\mathbf{x}))^{D/2}} f(\mathbf{x}), \tag{3.16}$$

where we defined

$$f(\mathbf{x}) = \left(\frac{\psi^{\mathrm{tr}}(\mathbf{x})}{\psi(\mathbf{x})} \right)^{D/2}. \tag{3.17}$$

The interesting feature of $f(\mathbf{x})$ is that it bounded from above, specifically

$$f(\mathbf{x}) \leq 1,$$

which follows directly from eq. (3.15). Hence, $f(\mathbf{x})$ does not have any poles, as the singularities of the polynomial ψ are suppressed by the zeros of its tropical approximation ψ^{tr}.

To further simplify this integral we can use a method that goes by the name of *sector decomposition* introduced by Hepp in [21]. This amounts to divide the integration region $\mathbb{RP}_+^{|E|-1}$ into different sectors $\mathcal{D}_\sigma$ defined as

$$\mathcal{D}_\sigma = \left\{ [x_{\sigma(1)}, \cdots, x_{\sigma(|E|)}] \in \mathbb{RP}_+^{|E|-1} \mid x_{\sigma(1)} < \cdots < x_{\sigma(|E|)} \right\},$$

where $\sigma \in S_{|E|}$ is a permutation of the $|E|$ homogeneous coordinates. Thus, the integral can be expanded as

$$\mathcal{P}_G = \sum_{\sigma \in S_{|E|}} \int_{\mathcal{D}_\sigma} \Omega \; \frac{\prod_e x_e^{\nu_e}}{(\psi^{\text{tr}}(\mathbf{x}))^{D/2}} f(\mathbf{x}) = \sum_{\sigma \in S_{|E|}} I_\sigma \text{ where } I_\sigma = \int_{\mathcal{D}_\sigma} \Omega \; \frac{\prod_e x_e^{\nu_e}}{(\psi^{\text{tr}}(\mathbf{x}))^{D/2}} f(\mathbf{x}). \quad (3.18)$$

To evaluate each sector integral I_σ, we shall notice that in each sector there is a unique spanning tree T_σ that minimises the factor x_e associated to each edge, called *minimal spanning tree*. Thus, in the sector $\mathcal{D}_\sigma$ the function ψ^{tr} is given by:

$$\psi^{\text{tr}}(\mathbf{x})\Big|_{\mathbf{x} \in \mathcal{D}_\sigma} = \prod_{e \notin T_\sigma} x_e. \quad (3.19)$$

Before explaining how to find the spanning tree T_σ, let us define G_e^σ as the subgraph formed by the first e edges in G, in ascending order according to the permutation σ; in symbols:

$$G_k^\sigma = \{\sigma(1), \ldots, \sigma(k)\} \subset G, \quad (3.20)$$

with $G_0^\sigma = \emptyset$, the empty graph.

In order to find T_σ one can utilise algorithm 1, that goes by the name *"Kruskal's greedy algorithm"* [26]. To implement this algorithm, one starts by considering an empty set T_σ and the set of all the edges of G in the order σ, which is labelled $G_{|E|}^\sigma$, using the notation defined above. Then, starting from the first element of the set $G_{|E|}^\sigma$ and then iterating accordingly, at each step one needs to add an edge $e \in G_{|E|}^\sigma$ to the set T_σ if it does not increase the loop number, until T_σ is a spanning tree. This means that the edges inside the spanning tree will be all those edges $e = \sigma(k)$ for which it holds $L(G_k^\sigma) = L(G_{k-1}^\sigma)$, as these edges do not increase the loop order of the graph.

Algorithm 1 Kruskal's greedy algorithm

```
Create an empty set of edges T_σ = {∅}
Set k = 1
while k ≤ |E| or T_σ is not spanning do
    if T_σ ∪ {σ(k)} is a tree then
        Add the edge σ(k) to the forest T_σ
    end if
    Set k = k + 1
end while
```

As this is one of the crucial points of this derivation, we will discuss an explicit example in order to convince the reader of the last statement. A graph with an explicit ordering of the edges is represented in fig. 3.2. For simplicity, let us consider the identity permutation, i.e. $\sigma(e) = e$ for any edge e of the graph. In this case we have,

$$G_6^\sigma = \{1, 2, 3, 4, 5, 6\}.$$

We can now apply algorithm 1 to construct the minimal spanning tree T_σ. We can immediately add the edges labelled 1, 2, 3. Then, edge 4 cannot be added to T_σ, as it will add a loop to it. Finally, edge 5 gets added to T_σ. Notice that $L(G_4^\sigma) = 1$, and edge 5 keeps the loop order constant $L(G_5^\sigma) = 1$, as in the statement above, i.e. $L(G_5^\sigma) = L(G_4^\sigma)$. This type of mechanism is completely general: when cycling over the edges, in ascending order, every edge either keeps the loop number constant, and thus gets added to the spanning tree, or increases the loop number, and thus it is not added to the tree. For the sake of completion, we shall report the final form of the minimal spanning tree:

$$T_\sigma = \{1, 2, 3, 5\}.$$

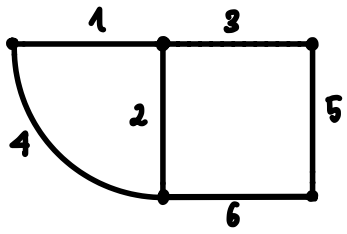


Figure 3.2: Example of graph with an explicit labelling of the edges.

As outlined in [28], the statement just discussed implies the edges $e = \sigma(k)$ that are not in the minimal spanning tree are the ones for which the loop number increases, that is $L(G_k^\sigma) = 1 + L(G_{k-1}^\sigma)$. By eq. (3.19) these are edges are that contribute to ψ^{tr}. We can therefore write

$$\psi^{\text{tr}}(\mathbf{x})\Big|_{\mathbf{x}\in\mathcal{D}_\sigma} = \prod_{e\notin T_\sigma} x_k = \prod_{k=1}^{|E|} x_{\sigma(k)}^{L(G_k^\sigma)-L(G_{k-1}^\sigma)}, \tag{3.21}$$

as for the edges that are in the spanning tree the exponent is vanishing, and for the ones in the spanning tree it gives 1.

Picking an affine chart such that $x_{\sigma(|E|)} = 1$, each sector integral I_σ can be written as:

$$I_\sigma = \int_{x_{\sigma(1)}<\cdots<x_{\sigma(N)}=1} \prod_{k=1}^{|E|-1} \mathrm{d}x_{\sigma(k)}\, x_{\sigma(k)}^{\nu_{\sigma(k)}-1-\frac{D}{2}\left(L(G_k^\sigma)-L(G_{k-1}^\sigma)\right)} f(\mathbf{x}).$$

To proceed we can perform the change of variables $y_k = \frac{x_{\sigma(k)}}{x_{\sigma(k+1)}}$, which yields the result:

$$I_\sigma = \int_{[0,1]^{|E|-1}} \left(\prod_{k=1}^{|E|-1} dy_k\right) \left(\prod_{k=1}^{|E|-1} y_k^{\omega(G_k^\sigma)-1}\right) f(\mathbf{x}(\mathbf{y})), \tag{3.22}$$

where $\omega(G_e^\sigma)$ is the superficial degree of divergence, defined in (2.14), for the subgraph G_e^σ.

Finally, we can perform the change of variables $\xi_k = \frac{1}{\omega(G_k^\sigma)} y_k^{\omega(G_k^\sigma)}$, which gives the result

$$I_\sigma = \frac{1}{\prod_{k=1}^{|E|-1}\omega(G_k^\sigma)} \int_{[0,1]^{|E|-1}} \left(\prod_{k=1}^{|E|-1} d\xi_k\right) f(\mathbf{x}(\boldsymbol{\xi})), \tag{3.23}$$

where we used the fact that all the $\omega(G_k^\sigma) > 0$, as we are considering primitive graphs, that by definition 3.1 are subdivergence free.

As stated above, the integrand $f(\mathbf{x}(\boldsymbol{\xi}))$ does not have any singularities, thus from this expression it is now possible to perform a Monte Carlo integration as outlined in the previous section. In fact, we now have the measure $\prod_{k=1}^{|E|-1} \mathrm{d}\xi_k$ over the hypercube $[0,1]^{|E|-1}$, for which it holds:

$$\int_{[0,1]^{|E|-1}} \prod_{k=1}^{|E|-1} \mathrm{d}\xi_k = 1 \text{ and } \prod_{k=1}^{|E|-1} \mathrm{d}\xi_k \geq 0 \text{ in } [0,1]^{|E|-1}, \tag{3.24}$$

and thus it is a probability measure. To practically implement the Monte Carlo sampling, one starts by drawing N vectors $\boldsymbol{\xi}^{(i)}$, with $i = 1, \ldots, N$, from the probability distribution above. Then one computes $x_k^{(i)}(\boldsymbol{\xi}) = \prod_{l=k}^{|E|-1} \left(\xi_l^{(i)}\right)^{1/\omega(G_i^\sigma)}$, as these will be the coordinates of the point in the positive orthonant of projective space, that is $\mathbf{x}^{(i)} = [x_1^{(i)}, \ldots, x_{|E|}^{(i)}] \in \mathbb{RP}_+^{|E|-1}$. Finally, one can compute the following estimator for I_σ

$$I_\sigma^{(N)} = \frac{1}{\prod_{e=1}^{|E|-1} \omega(G_e^\sigma)} \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}^{(i)}). \tag{3.25}$$

In fact, by using eq. (3.3) it follows $\mathbb{E}[I_\sigma^{(N)}] = I_\sigma$.

Finally, to compute the period, one needs to reiterate the Monte Carlo integration described for each sector $\mathcal{D}_\sigma$. However, as explained in [7, Section 4], the number of permutations σ grows factorially with the dimensionality of the integral. In addition, the contribution of each sector integral I_σ varies greatly, making the evaluation of each one unnecessary. A possible solution, proposed in [7, Section 5], is to use an importance sampling on the sum of the I_σ integrals in eq. (3.18), and it will be outlined in section 3.4.

3.4 Tropical sampling algorithm

Let us now write the period (3.1) as:

$$\mathcal{P}_G = \int_{\mathbb{RP}_+^{|E|-1}} \mu \;\; f(\mathbf{x}),$$

where we defined the measure on $\mathbb{RP}_+^{|E|-1}$

$$\int \mu = \int \Omega \frac{\prod x_e^{\nu_e}}{(\psi^{\mathrm{tr}}(\mathbf{x}))^{D/2}}.$$

This measure is normalisable, as opposed to $\int \Omega$. We can thus define the following prefactor

$$\mathcal{H}_G = \int_{\mathbb{RP}_+^{|E|-1}} \Omega \;\; \frac{\prod x_e^{\nu_e}}{(\psi^{\mathrm{tr}}(\mathbf{x}))^{D/2}}. \tag{3.26}$$

This normalisation factor is the tropical approximation of the period, and it is called *Hepp's Bound* in the literature. Its relationship with the period has been studied in [28]. We can find an explicit expression for the Hepp's bound by using the sector decomposition as it was done in the previous section. This amounts to setting $f(\mathbf{x}) = 1$ and repeating the same steps done above for the integral $\mathcal{P}_G$. Overall, we get the following expression:

$$\mathcal{H}_G = \sum_{\sigma \in S_{|E|}} \mathcal{H}_G^\sigma, \tag{3.27}$$

where each sector gives the contribution

$$\mathcal{H}_G^\sigma = \frac{1}{\prod_{e=1}^{|E|-1} \omega(G_e^\sigma)}. \tag{3.28}$$

We can factorise the period as:

$$\mathcal{P}_G = \mathcal{H}_G \int_{\mathbb{RP}_+^{|E|-1}} \mu^{\text{tr}} \;\; f(\mathbf{x}), \tag{3.29}$$

where we defined the tropical differential form

$$\mu^{\text{tr}} = \; \Omega \frac{1}{\mathcal{H}_G} \frac{\prod x_e^{\nu_e}}{(\psi^{\text{tr}}(\mathbf{x}))^{D/2}}.$$

Furthermore, eq. (3.15) implies that $\mathcal{P}_G \leq \mathcal{H}_G$, which indicates that

$$\int_{\mathbb{RP}_+^{|E|-1}} \mu^{\text{tr}} \;\; f(\mathbf{x}) \leq 1.$$

We can show that μ^{tr} is a probability measure, in fact it holds that:

$$\int_{\mathbb{RP}_+^{|E|-1}} \mu^{\text{tr}} = \frac{1}{\mathcal{H}_G} \int_{\mathbb{RP}_+^{|E|-1}} \Omega \;\; \frac{\prod x_e^{\nu_e}}{(\psi^{\text{tr}}(\mathbf{x}))^{D/2}} = 1, \tag{3.30}$$

and $\mu^{\text{tr}} \geq 0$ for any point in $\mathbb{RP}_+^{|E|-1}$. Therefore, μ^{tr} could allow us to implement a Monte Carlo sampling algorithm. In particular, let us suppose that we find a way to generate N random points $\mathbf{x}^{(i)}$ from the distribution μ^{tr}, then we can evaluate the quantity

$$\mathcal{P}_G^{(N)} = \frac{\mathcal{H}_G}{N} \sum_{i=1}^{N} f(\mathbf{x}^{(i)}), \tag{3.31}$$

and this will be an estimator for $\mathcal{P}_G$, as the average $G^{(N)}$ could be used to estimate the value of the integral I in section 3.1.

However, now we need to find a way to generate a random sample form μ^{tr} that overcomes the bottleneck of the factorially growing number of permutations. This can be achieved by implementing an importance sampling in the sum over the permutations: by eq. (3.27), we can interpret $p(\sigma) = \mathcal{H}_G^\sigma / \mathcal{H}_G$ as a probability of choosing a permutation σ. After all, it holds that $\sum_\sigma p(\sigma) = 1$, and $0 \leq p(\sigma) \leq 1$. Thus, one chooses a permutation, and then generates a random point in $\mathbb{RP}_+^{|E|-1}$, similarly to how it was done in the previous section: first one draws a random point $\boldsymbol{\xi}$ from the uniform distribution over the hypercube $[0,1]^{|E|-1}$, then computes $x_k(\boldsymbol{\xi}) = \prod_{l=k}^{|E|-1} (\xi_l)^{1/\omega(G_l^\sigma)}$, and the resulting vector $\mathbf{x}$ will be in the desired integration region $\mathbb{RP}_+^{|E|-1}$. Finally, one can evaluate the quantity $\mathcal{P}_G^{(N)}$ from (3.31). This will be a good estimator for the period, in fact:

$$\mathbb{E}[\mathcal{P}_G] = \mathcal{H}_G \mathbb{E}[f(\mathbf{x})] = \mathcal{H}_G \sum_{\sigma \in S_{|E|}} \frac{\mathcal{H}_G^\sigma}{\mathcal{H}_G} \int_{[0,1]^{|E|-1}} \left(\prod_{e=1}^{|E|-1} d\xi_e \right) \; f(\mathbf{x}(\boldsymbol{\xi})) = \mathcal{H}_G \int_{\mathbb{RP}_+^{|E|-1}} \mu^{\text{tr}} f(\mathbf{x}) \tag{3.32}$$

where in the last step we used eq. (3.28) and (3.23). The equation above proves that this procedure samples from μ^{tr}, as we can successfully numerically estimate the integral $\int_{\mathbb{RP}_+^{|E|-1}} \mu^{\text{tr}} f(\mathbf{x})$. The schematics of this algorithm are reported in algorithm 2.

Algorithm 2 Tropical sampling algorithm for Feynman periods

for $i \in 1, \dots N$ **do**
  Sample a permutation σ with probability $p(\sigma) = \mathcal{H}_G^\sigma / \mathcal{H}_G$.
  Draw a random vector $\xi \in [0,1]^{|E|-1}$ from a uniform distribution.
  Compute $x_k^{(i)}(\boldsymbol{\xi}) = \prod_{l=k}^{|E|-1} \left(\xi_l^{(i)}\right)^{1/\omega(G_l^\sigma)}$.
  Return $\mathbf{x}^{(i)} = [x_1^{(i)}, \dots, x_{|E|}^{(i)}] \in \mathbb{RP}_+^{|E|-1}$.
end for
Return $\mathcal{P}_G^{(N)} = \frac{\mathcal{H}_G}{N} \sum_{i=1}^N f(\mathbf{x}^{(i)})$.

The main bottleneck of this procedure is that, before implementing the algorithm described above, one needs to compute a table with all the probabilities $\mathcal{H}_G^\sigma/\mathcal{H}_G$ of size $(|E|-1)!$, and a table with all the $\omega(G_k^\sigma)$ for $k = 1, \dots, |E|-1$. This preprocessing was already required for the numerical integration of equation (3.23) described in the previous section, making sampling from μ^{tr} advantageous if only some sector integrals I_σ contribute to most of the period. However, the preprocessing is still by far the biggest downside of this algorithm; due to the factorial dependence of the size of the table on the dimension of integration, the numerical integration becomes impractical at high orders. In [7] there is an interesting solution to this issue that effectively reduces the factorial dependence to an exponential one. This will be discussed in section 3.4.2.

3.4.1 Tropical integration applied

Let us now try to apply the tropical sampling algorithm to the integral I in eq. (3.7) discussed in subsection 3.1.1. Before applying algorithm 2, we need to construct the table of probabilities $p(\sigma)$. Thus, we define the tropical approximation of the integral I:

$$I^{\text{tr}} = \int_{[0,1]^2} \mathrm{d}x\mathrm{d}y \frac{1}{\max(x,y)}, \tag{3.33}$$

where $\max(x,y)$ is the tropical approximation of the polynomial $x+y$.

We can explicitly compute the integral above by splitting the integration domain in the two sectors

$$\mathcal{D}_1 = \left\{(x,y) \in [0,1]^2 \,|\, x < y\right\},$$
$$\mathcal{D}_2 = \left\{(x,y) \in [0,1]^2 \,|\, x > y\right\},$$

which are represented in fig. 3.3.

This implies that the integral splits as

$$I^{\text{tr}} = I_1^{\text{tr}} + I_2^{\text{tr}} \quad \text{where} \quad I_i^{\text{tr}} = \int_{\mathcal{D}_i} \mathrm{d}x\mathrm{d}y \frac{1}{\max(x,y)}.$$

Let us now compute I_1:

$$I_1^{\text{tr}} = \int_{\mathcal{D}_1} \mathrm{d}x\mathrm{d}y \frac{1}{\max(x,y)} = \int_{\mathcal{D}_1} \mathrm{d}x\mathrm{d}y \frac{1}{y}$$

we can perform the change of variables $z = \frac{x}{y}$, in analogy with the change of variable performed to derive eq. (3.22), which implies

$$I_1^{\text{tr}} = \int_{[0,1]^2} dz dy = 1.$$

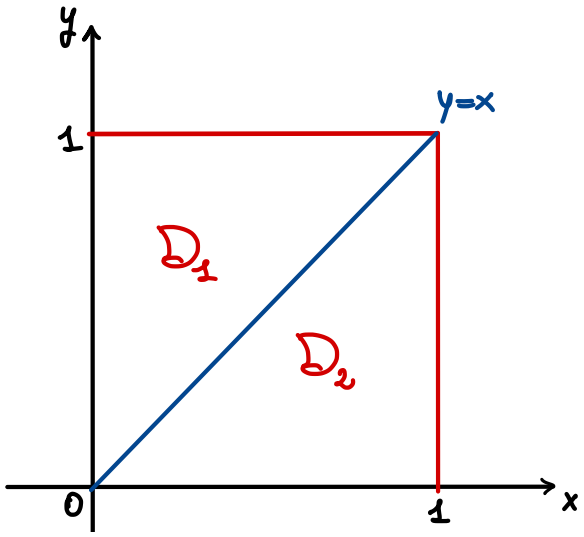


Figure 3.3: Representation of the two sectors $\mathcal{D}_1$ and $\mathcal{D}_2$. In $\mathcal{D}_1$ we have $x < y$ and in $\mathcal{D}_1$ we have $x > y$. This splitting of the square $[0,1]^2$ allows for an explicit computation of the tropical integral in (3.33).

The integration for I_2 is identical and yields the same result. Thus, we get the following result for I^{tr}:

$$I^{\text{tr}} = I_1^{\text{tr}} + I_2^{\text{tr}} = 2.$$

Now we can build our table of probabilities of sampling from each sector. These will be $p(\sigma_1) = p(\sigma_2) = \frac{1}{2}$, where $p(\sigma_i)$ is the probability of sampling from the sector $\mathcal{D}_1$.

Let us now focus on I again. Before applying algorithm 2, we need to manipulate the integral to define the probability measure μ^{tr} relevant for this problem, as it was done in eq. (3.29) for the period, so that we can construct a quantity that we can compute numerically that estimates I, similarly to $\mathcal{P}_G^{(N)}$ in equation (3.31). In analogy with eq.(3.16), the integral factorises as:

$$\int_{[0,1]^2} \frac{\mathrm{d}x\mathrm{d}y}{\max(x,y)} \tilde{h}(x,y) \quad \text{with} \quad \tilde{h}(x,y) = \frac{\max(x,y)}{x+y}, \tag{3.34}$$

where $\max(x,y)$ is the tropical approximation of the polynomial $x+y$. Then, we can define the tropical measure

$$\mu^{\text{tr}} = \frac{1}{I^{\text{tr}}} \frac{\mathrm{d}x\mathrm{d}y}{\max(x,y)},$$

for which it holds that

$$\int_{[0,1]^2} \mu^{\text{tr}} = \frac{1}{I^{\text{tr}}} \int_{[0,1]^2} \frac{\mathrm{d}x\mathrm{d}y}{\max(x,y)} = 1.$$

We can then write I in the following way:

$$I = I^{\text{tr}} \int_{[0,1]^2} \mu^{\text{tr}} \tilde{h}(x,y).$$

Now we can use algorithm 2 to generate N samples $(x^{(i)}, y^{(i)})$, with $i = 1, \ldots, N$, from the probability distribution μ^{tr}, and then evaluate the quantity:

$$G^{(N)} = \frac{I^{\text{tr}}}{N} \sum_{i=1}^{N} \tilde{h}(x^{(i)}, y^{(i)}). \tag{3.35}$$

As explained in the previous section, this will be an estimator for the integral I. Furthermore, we can estimate the error of $G^{(N)}$ by using σ_I defined in eq. (3.5). The results obtained for the numerical integration are in fig. 3.4.

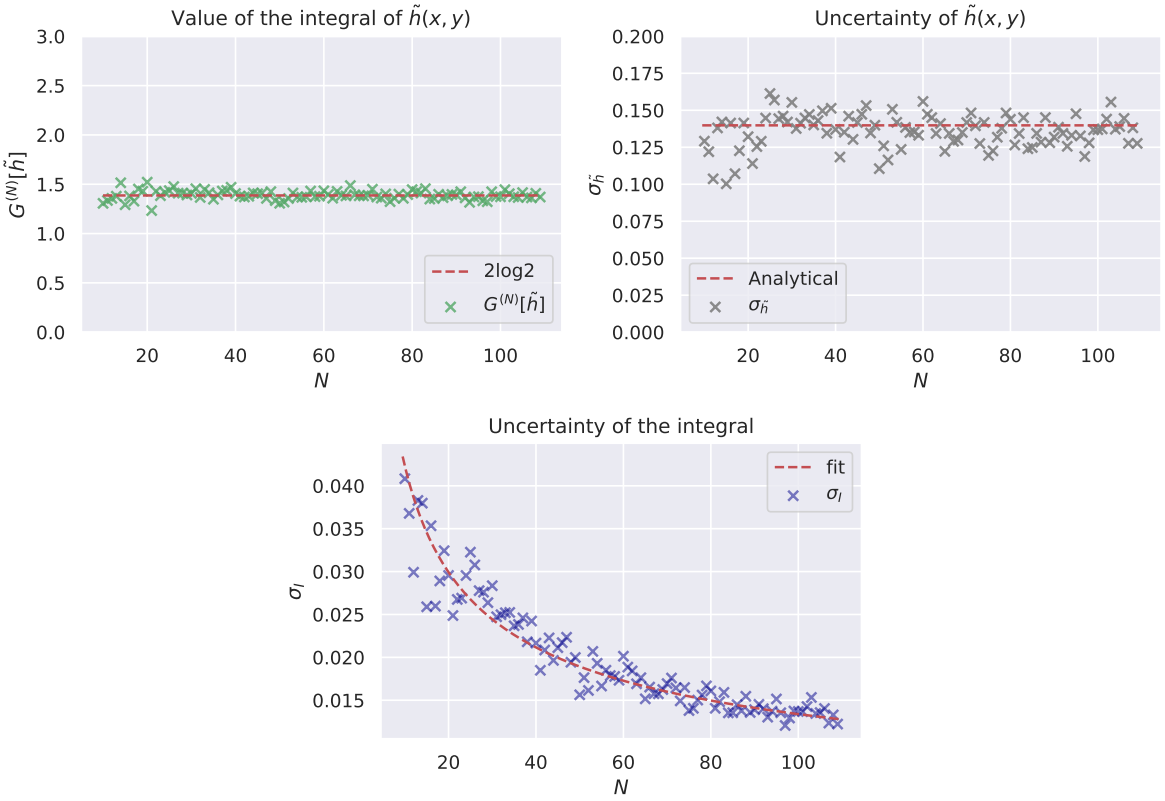


Figure 3.4: Results obtained for the tropical integration of I with varying number of samples N. In the first plot we have the value of the numerical integral (green crosses) compared with its analytical value (red dashed line). The uncertainties of the integrand $\sigma_{\tilde{h}}$ are represented in the second plot with gray crosses. The analytical value of the $\sqrt{\mathrm{Var}[\tilde{h}]}$ is represented as a dashed red line. Finally, in the third plot we can see the uncertainties of the numerical integral σ_I (blue crosses) and a fit (red dashed line) with a function $\propto 1/\sqrt{N}$.

In the first plot of fig. 3.4 we can see the values of $G^{(N)}$, represented as blue crosses, where N varies in the range $[5, 100]$ with increments of 5. The numerical estimator $G^{(N)}$ has values distributed around the theoretical value $2\log 2$, as in the first plot in fig. 3.1a. A thorough comparison of these two plots and a direct computation reveals that the integration with the tropical integration method results in data that is less sparse: The standard deviation is smaller, and the points are on average closer to the analytical value. If we then focus on the uncertainty $\sigma_{\tilde{h}}$, we can appreciate the power of the tropical integration: The data does not exhibit the divergent pattern of the second plot in fig. 3.1a. In fact, it oscillates around the analytical value given by $\sqrt{\mathrm{Var}[\tilde{h}]} \simeq 0.140$, where

$$\mathrm{Var}[\tilde{h}] = \int_{[0,1]^2} \mu^{\mathrm{tr}} \left(\frac{\max(x, y)}{x + y} - \log 2 \right)^2 = \frac{1}{2} - 2\log 2, \tag{3.36}$$

where we used the fact that

$$\int_{[0,1]^2} \mu^{\mathrm{tr}} \tilde{h}(x, y) = \log 2.$$

Let us now analyse the plot of the uncertainties of the numerical integral σ_I. We can notice that the uncertainties are convergent and exhibit the $1/\sqrt{N}$ behaviour expected as of eq. (3.5). Thus, we can fit the σ_I with the function $\sigma_{\tilde{h}}^{fit}/\sqrt{N}$, and this yields the result

$$\sigma_{\tilde{h}}^{fit} = 0.1338 \pm 0.0014$$

which is statistically compatible with the analytical result. We can also compute the mean value and the standard deviation of the $\sigma_{\tilde{h}}$

$$\overline{\sigma_{\tilde{h}}} = 0.136 \pm 0.011,$$

which is also compatible with the analytical value, and with the results of the fits $\sigma_{\tilde{h}}^{fit}$ Note that the observation that the uncertanties $\sigma_{\tilde{h}}$ are less sparse is reflected in the value of their standard deviation, which is an entire order smaller than the standard deviation of σ_h in the example in fig. 3.1a.

Overall, the data generated with the tropical integration algorithm, shows the same features as the integration of the function $g(x, y)$ in eq. (3.10). In fact, by a direct comparison with the plots in fig. 3.1b, we can see that the overall pattern for σ_I and $\sigma_{\tilde{h}}$ is similar, which explicitly points out the fact that the tropical integration produces results which are qualitatively identical to a (convergent) Monte Carlo integration for a square integrable function.

3.4.2 Tropical sampling sped up

The problem of computing $\mathcal{P}_G$ is thus still constrained by the dimensionality of the integral, as the table of probabilities has to be computed and kept in the memory. In [7] it was shown that the preprocessing can be sped up using the following recursion

$$J_G(\gamma) = \sum_{e \in \gamma} \frac{J_G(\gamma \setminus e)}{\omega(\gamma \setminus e)} \tag{3.37}$$

for all subgraphs $\gamma \subset G$ where we set $J_r(\emptyset) = \omega(\emptyset) = 1$. In this recursion, we start with a subgraph γ, and at each step we remove an edge e and compute the superficial degree of divergence according to eq. (2.14) of the resulting subgraph $\gamma \setminus e$. By repeating this process for all the edges $e \in \gamma$, one finds the value of the function $J_G(\gamma)$.

This function is interesting as the original problem of computing all the probabilities $\mathcal{H}_G^\sigma/\mathcal{H}_G$ can be mapped to the one of computing all the $J_G(\gamma)$ with $\gamma \subset G$. To show this, we can start by showing that

$$J_G(G) = \mathcal{H}_G = \sum_{\sigma \in S_{|E|}} \frac{1}{\prod_{e=1}^{|E|-1} \omega(G_e^\sigma)}. \tag{3.38}$$

To prove this statement, let us first explicitly explain the definition of the $J_G(G)$ function in eq. (3.37). One starts by considering $G_n = G$, with $n = |E|$, and then one sums over all the edges $e_n \in G_n$, computes $1/\omega(G_{n-1})$, and considers the subgraph $G_{n-1} = G_n \setminus e_n$, and then iterates the same procedure again for all the n edges of the graph. We can write this recursion explicitly as

$$J_G(G) = \sum_{e_n \in G_n} \frac{1}{\omega(G_{n-1})} \sum_{e_{n-1} \in G_{n-1}} \frac{1}{\omega(G_{n-2})} \cdots \sum_{e_1 \in G_1} \frac{1}{\omega(G_0)} = \sum_{e_n \in G_n} \cdots \sum_{e_1 \in G_1} \frac{1}{\prod_{k=1}^{|E|-1} \omega(G_k)} \tag{3.39}$$

where we used the fact that $G_0 = \emptyset$. Then, we note that a permutation σ is a bijection $G_k^\sigma \to \gamma$ where $|E(\gamma)| = k$. This means that fixing a bijection σ is equivalent to fixing one edge $e \in \gamma$

and another bijection σ' from $G_{k-1}^{\sigma'} \to \gamma'$, where γ' is the subgraph given by removing the edge e from γ, i.e. $\gamma' = \gamma \setminus e$. Thus, we can effectively map the sums over the edges $e_n, \dots, e_1$ to the sum over the permutations σ, which means that we get the expression on the right-hand side of eq. (3.38), and thus prove the statement.

The equality proven allows us to focus only on the quantity in eq. (3.37). In particular, instead of computing the table of probabilities $p(\sigma)$, we can compute the table of values of $J_G(\gamma)$ for all the subgraphs $\gamma \subset G$, which are $2^{|E|}$. This last statement follows from the fact that we defined a subgraph of G as a subset of edges of G; thus, the set of all the subgraphs of G is the powerset of the edges, denoted as $\mathbf{2}^E$, and hence the number of subgraphs is the dimension of the power set, which is $2^{|E|}$. This means that the preprocessing necessary for the Monte Carlo integration is now reduced to a table of size $2^{|E|}$, instead of $(|E|-1)!$.

Now let us modify the part of algorithm 2 that samples from μ^{tr} with alg. 3. We start by considering the full graph $\gamma = G$ and a number κ that is set to 1. Instead of choosing a permutation σ, we pick a random edge e with probability $p_e = \frac{1}{J_G(\gamma)} \frac{J_G(\gamma \setminus e)}{\omega(\gamma \setminus e)}$, and set the e-th coordinate of the point in projective space to be κ. Then, we pick a random real number in the domain $[0,1]$ and set κ to be $\kappa \xi^{1/\omega(\gamma)}$. One iterates this steps until $\gamma = \emptyset$, and returns the sample $\mathbf{x} = [x_1, \dots, x_{|E|}]$ in projective space. This is virtually identical to choosing a permutation σ and then computing $x_e = \prod_{i=e}^{|E|-1} (\xi_i)^{1/\omega(G_i^\sigma)}$ where ξ is a random vector in the hypercube $[0,1]^{|E|-1}$. In fact, by choosing the edges with probability p_e, we are effectively "building" a permutation σ, where the element $\sigma(|E(\gamma)|)$ will be e, with γ being the subgraph at the step where we picked the edge e. Finally, as it was done previously, after generating N samples from μ^{tr}, one evaluates the numerical integral $\mathcal{P}_G^{(N)}$ in eq. (3.31).

Algorithm 3 Tropical sampling algorithm.

Set $\gamma = G$ and $\kappa = 1$.
while $\gamma \neq \emptyset$ **do**
  Pick a random $e \in \gamma$ with probability $p_e = \frac{1}{J_G(\gamma)} \frac{J_G(\gamma \setminus e)}{\omega(\gamma \setminus e)}$.
  Remove e from γ.
  Set $\sigma(|E(\gamma)|) = e$.
  Set $x_e^{(i)} = \kappa$.
  Pick a uniformly distributed random number $\xi \in [0,1]$.
  Set $\kappa \leftarrow \kappa \xi^{1/\omega(\gamma)}$.
end while
Return $\mathbf{x}^{(i)} = [x_1^{(i)}, \dots, x_{|E|}^{(i)}] \in \mathbb{RP}_+^{|E|-1}$ and $\sigma = (\sigma(1), \dots, \sigma(|E|)) \in S_{|E|}$.

To convincingly prove that generating samples according to the algorithm above is equivalent to integrating with the algorithm 2, we can show that $\mathcal{P}_G^{(N)}$ evaluated with points generated with alg. 3 correctly estimates the period. This amounts to show:

$$\mathbb{E}[f(\mathbf{x})] = \sum_{e_n \in G_n} \frac{1}{J_G(G_n)} \frac{J_G(G_n \setminus e_n)}{\omega(G \setminus e_n)} \cdots \sum_{e_1 \in G_1} \frac{1}{J_G(G_1)} \frac{J_G(G_1 \setminus e_1)}{\omega(G_1 \setminus e_1)} \int_{[0,1]^{n-1}} \left(\prod_{k=1}^{n-1} \mathrm{d}\xi_k \right) f(\mathbf{x}(\boldsymbol{\xi})),$$

where we set $n = |E|$ to avoid cluttering the notation and $G_{k-1} = G_k \setminus e_k$, as it was done above. By using the recursion in (3.37) and the result in (3.38), we get

$$\mathbb{E}[f(\mathbf{x})] = \frac{1}{\mathcal{H}_G} \sum_{e_n \in G_n} \cdots \sum_{e_1 \in G_1} \frac{1}{\omega(G_n) \cdots \omega(G_1)} \int_{[0,1]^{n-1}} \left(\prod_{k=1}^{n-1} \mathrm{d}\xi_k \right) f(\mathbf{x}(\boldsymbol{\xi})). \tag{3.40}$$

Finally, we use eq. (3.28) and map the sum over the edges to a sum over the permutations

σ, as it was done above, to get:

$$\mathbb{E}[f(\mathbf{x})] = \sum_{\sigma \in S_E} \frac{\mathcal{H}_G^\sigma}{\mathcal{H}_G} \int_{[0,1]^{n-1}} \left(\prod_{k=1}^{n-1} \mathrm{d}\xi_k \right) f(\mathbf{x}(\boldsymbol{\xi})),$$

which yields the same result as eq. (3.32).

As stated above, algorithm 3 has the clear benefit of having to a faster preprocessing, if compared to the naive algorithm 2. In fact, The table with the values of $J_G(\gamma)$ has 2^n entries, and it can be computed in $\mathcal{O}(n2^n)$ steps by using the recursion in (3.37). This allows us to carry out computations at a higher loop order while being less limited by memory requirements. In [7], this powerful tool allowed for computations of period integrals up to 17 loop orders, which would be practically impossible with a naive Monte Carlo integration.

In the next chapter we will try to use this technique to study statistical properties of period integrals at high loop orders by computing a high number of periods (a thousand) up to $L = 15$.

Chapter 4

Distribution for randomly generated graphs

In this chapter we will focus mostly on a ϕ^4 QFT. As mentioned in chapter 1, the Feynman rules for the Lagrangian in eq. (1.10) imply that we shall only consider graphs that have vertices with 4 valencies, that is, graphs where each vertex has 4 lines attached to it. Examples of ϕ^4 graphs are in figures 4.2 and 4.6. We will also focus on p-log graphs, unless explicitly stated otherwise, like in the examples in fig. 4.2, which will be discussed in section 4.1.1. We will also set the edge weights ν_e, defined in the introduction, to be all 1. This means that according to equation (2.14) for any graph G we will have:

$$\omega = |E| - \frac{D}{2}L.$$

The main goal of this chapter is to find a way to numerically estimate the p-log loop amplitude in (1.3), that we will label A_L to avoid clutter in the notation, for loop orders $L < 16$. Given that the number of graphs increases factorially in the loop order, it becomes impossible to compute A_L directly using the expression in eq. (1.3). However, we can estimate its value by randomly generating L loops graphs G with a probability $p(G)$ proportional to their symmetry factor $|\mathrm{Aut}(G)|$, and then evaluate the mean value of the periods $\mathcal{P}_G$, in formulas:

$$\mathbb{E}[\mathcal{P}_G]|_L = \sum_{L(G)=L} \mathcal{P}_G\, p(G) \simeq Z_L \sum_{L(G)=L} \frac{\mathcal{P}_G}{|\mathrm{Aut}(G)|} = Z_L A_L, \tag{4.1}$$

where the sum runs over all those graphs that have loop order L, and we defined Z_L as the normalisation factor of $p(G)$ limited to the graphs with L loops, i.e.

$$Z_L = \sum_{L(G)=L} \frac{1}{|\mathrm{Aut}(G)|}. \tag{4.2}$$

In section 4.1, we will show how to implement the algorithm, necessary to conduct the analysis on A_L, that generates a graph with probability

$$p(G) = Z_L^{-1} \frac{1}{|\mathrm{Aut}(G)|}, \tag{4.3}$$

where G has L loops.

Practically, one generates a random graph G with this algorithm, and then, using the method described in the previous chapter, numerically evaluates the Feynman period $\mathcal{P}_G$ related to the graph, and averages over the resulting values $\mathcal{P}_G$. Finally, to get the value of A_L one would need to divide the result by the normalisation factor Z_L, but computing it can be extremely

hard due to the factorial dependence of the number of graphs on the loop order. Thankfully, in [6, Section 6.3] it is shown that for ϕ^4 theory for $L \gg 1$,

$$Z_L \sim \frac{e^{-15/4}}{\sqrt{2\pi}} \left(\frac{2}{3}\right)^{L+3} \Gamma(L+3) \left(36 + \mathcal{O}\left(\frac{1}{L}\right)\right). \tag{4.4}$$

This result allows us to estimate the values of A_L. In section 4.2 we will show the numerical results of this computation, and we will discuss the behaviour of the amplitude as the loop order changes. Furthermore, we will also analyse the distribution of $\mathcal{P}_G$ changing the loop order, and the behaviour of the uncertainty of $\mathbb{E}[\mathcal{P}_G]|_L$. In this chapter we will suppress the L label for the expected value to avoid clutter in the notation.

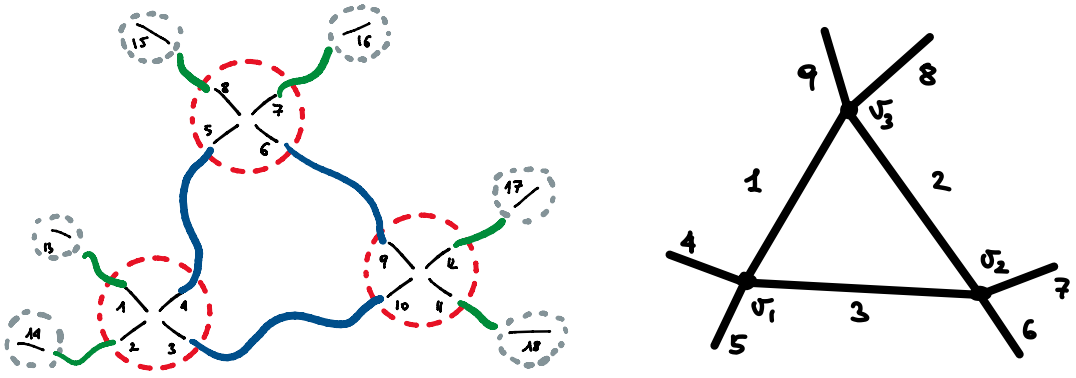
4.1 Generating random graphs

For the purpose of finding an algorithm that generates a graph with a probability proportional to $1/|\mathrm{Aut}(G)|$, we shall give a *combinatorial definition* of a Feynman graph G.

Definition 4.1. A Feynman graph G is a set of *half-edges* $H = \{1, \dots, |H|\}$ along with:

- A set of vertices V, which are subsets of half-edges with more than 2 elements, in symbols
$$v \in V \Leftrightarrow v = \{h_1, \dots, h_N\} \subset H \text{ and } |v| > 2;$$
- A set of external vertices V^{ext}, which are subsets of half-edges with 1 element;
- A set of external legs Ext which are pairs of vertices and external vertices, where at least one of the elements is in V^{ext}, in symbols
$$e \in Ext \Leftrightarrow e = (v_1, v_2) \subset H \text{ and } v_1 \text{ or } v_2 \in V^{ext};$$
- A set E of pairs of vertices, called edges.

The number of elements in any vertex $v \in V$ is called *valency* of v. From the definition above we can deduce that vertices with 1 or 2 valencies are not allowed.



(a) (b)

Figure 4.1: In subfig. a, example of a graph defined with definition 4.1; in subfigure b the same graph represented with the common graph theory notation.

We can intuitively show that this definition is equivalent to the one reported in most QFT [29, Chapter 1] and graph theory books [16, Chapter 1], by showing the example in fig. 4.1. Let

us focus on fig. 4.1a. First we pictorially represent the half-edges H as black lines and label them as $1, 2, \ldots, 18 = |H|$. The half-edges are then grouped into vertices, drawn as dashed red lines, and external vertices, drawn as dashed gray lines. Lastly, pairs of half-edges are connected and thus form either an edge, represented by thick blue line, or an external leg, represented by a thick green line; if both the half-edges in the pair are elements of a vertex, they form an edge, if at least one of the two is an element of an external vertex, then they form an external leg. This example corresponds to the graph drawn in fig. 4.1b, where in the usual labelling $v_1, v_2, v_3 \in V$ are vertices, $1, 2, 3 \in E$ are edges and $4, \ldots, 9 \in Ext$ are external legs.

Considering that the sum in (4.1) concerns graphs with a fixed loop L, we should try to express the number of half-edges $|H|$, the number of edges $|E|$ and vertices $|V|$ as functions of L. We will do so for graphs that have all vertices with fixed valency N and $|Ext|$ external legs. For these graphs the number of half-edges $|H|$ can be computed both as

$$|H| = N|V| + |V^{ext}|, \tag{4.5}$$

and as

$$|H| = 2(|E| + |Ext|). \tag{4.6}$$

The first equation comes from the fact that each vertex will contribute with N half-edges, and each external vertex with 1 half-edge. The second equation follows from considering the fact that each edge or external leg is a pair of half-edges. We can thus write

$$N|V| = 2|E| + |Ext|, \tag{4.7}$$

where we used the fact that the number of external legs and external vertices is the same, i.e.

$$|V^{ext}| = |Ext|. \tag{4.8}$$

By using equation (4.7), together with Euler's formula in eq. (1.7), we can relate the number of vertices to the number of loops and external legs:

$$|V| = \frac{2}{N-2}\left(L + \frac{|Ext|}{2} - 1\right). \tag{4.9}$$

Using again Euler's formula we can also find the number of edges:

$$|E| = \frac{1}{N-2}\left(N(L-1) + |Ext|\right). \tag{4.10}$$

Hence, we can determine the number of half-edges, by using either eq. (4.5) or (4.6).

We can now tackle the problem of generating random graphs with L loops, with vertices that have valency N, with a probability that follows equation (4.3). A simple algorithm to do so, is to draw a graph using definition 4.1 while connecting the half-edges randomly: We start by generating the sets of vertices and external vertices of size $|V|$, computed with eq. (4.9), and $|V^{ext}|$, respectively. Each element of V is an N-tuple that contains N half-edges and each element of V^{ext} is a 1-tuple that contains one half-edge. Then, all the half-edges are labelled as it was done in the example above. Let us now generate two empty sets of edges E and external legs Ext. To create edges and external legs, one can draw two random integers σ_1, σ_2 in the range $[1, |H|]$, where $|H|$ is computed according to eq. (4.5). Then, the vertices v_1, v_2 are paired together, where v_1, v_2 are the vertices that contain the half-edges that have the label respectively σ_1 and σ_2. Upon checking that the pair $p = (v_1, v_2)$ is not in E nor in Ext, p is added to the set Ext if either v_1 or v_2 is an external vertex, it is added to E otherwise. The process of generating random pairs of half-edges is reiterated until $2(|E| + |Ext|) = |H|$. Finally, this algorithm returns the set of edges and external legs. This procedure is schematically reported in algorithm 4.

Remark 4.2. This algorithm can produce disconnected graphs. But as stated previously, only connected graphs are relevant in our analysis; thus, the *BFS algorithm* [15, Section 22.2] was implemented to check whether a generated graph is connected, the disconnected ones were rejected.

Algorithm 4 Random generation of a graph with L loops, N-valent vertices and $|Ext|$ external legs.

Generate a set of N-tuples V with $|V| = \frac{2}{N-2}\left(L + \frac{|Ext|}{2} - 1\right)$ elements.
Generate a set of 1-tuples V^{ext} with $|V^{ext}|$ elements.
Generate N half-edges for each $v \in V$.
Generate 1 half-edge for each $v \in V^{ext}$.
Label all the $|H| = N|V| + |Ext|$ half-edges $h = 1, \dots, |H|$.
Generate two empty sets E, Ext.
while $2|E| + 2|Ext| < |H|$ **do**
  Draw two random integers σ_1, σ_2 in the range $[1, |H|]$.
  Create the pair $p = (v_1, v_2)$ where $\sigma_i \in v_i$, with $i = 1, 2$.
  if $p \notin e, \forall e \in E, Ext$ **then**
    if v_1 or $v_2 \in V^{ext}$ **then**
      Add $(v_1, v_2) \to Ext$.
    else
      Add $(v_1, v_2) \to E$.
    end if
  end if
end while

4.1.1 Example

Let us now discuss an example that numerically shows that algorithm 4 produces a graph with probability according to eq. (4.3) for $L = 2$. In this example we will focus on Feynman graphs for a ϕ^4 interaction, thus, with vertices that have valency $N = 4$. To make an explicit computation of the symmetry factors and the normalisation factor Z_L possible, we shall only consider graphs with 2 external legs. As we will show in the next section, for a ϕ^4 theory at this loop order there are no p-log graphs, but in this example we will consider 1PI, connected graphs, and we allow for subdivergences, as we will not compute the Feynman integral associated to the graphs. With this considered, all the graphs allowed at this loop order are the ones in fig. 4.2. An explicit computation of the symmetry factors gives the following results:

$$|\mathrm{Aut}(G_1)| = \frac{1}{4}, \quad |\mathrm{Aut}(G_2)| = \frac{1}{4}, \quad |\mathrm{Aut}(G_3)| = \frac{1}{6}.$$

By eq. (4.2), the normalisation factor turns out being:

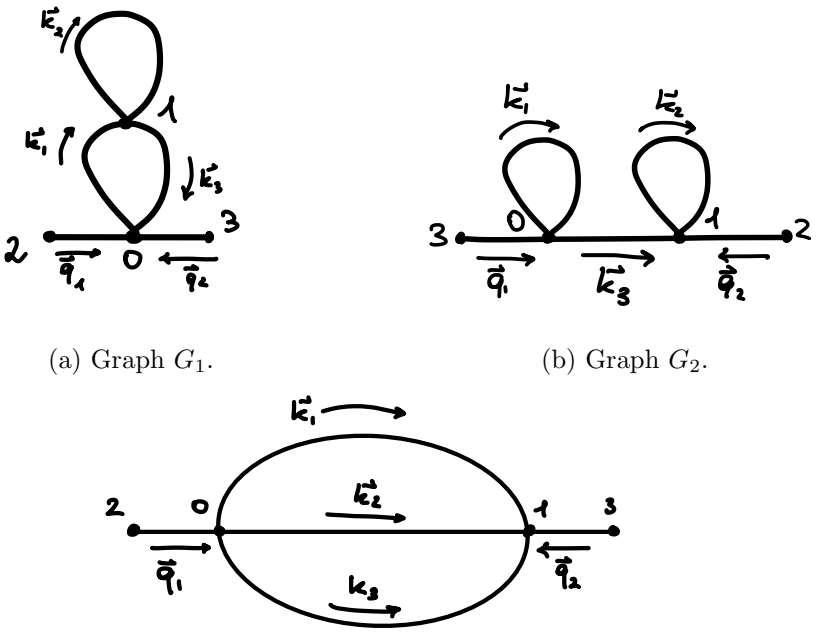
$$Z_{L=2} = \frac{1}{|\mathrm{Aut}(G_1)|} + \frac{1}{|\mathrm{Aut}(G_2)|} + \frac{1}{|\mathrm{Aut}(G_3)|} = \frac{2}{3}.$$

Thus, $p(G_3)$ will be:

$$p(G_3) = \frac{1}{4}.$$

To numerically check that algorithm 4 produces graphs according to the above result, we can use this algorithm to generate M graphs and then compute the ratio

$$r = \frac{\#(G_3)}{M},$$



(a) Graph G_1.

(b) Graph G_2.

(c) Graph G_3.

Figure 4.2: Diagrams at $L = 2$ with $|Ext| = 2$ with 4-valent vertices.

where $\#(G_3)$ is the number of occurrences of the graphs G_3. In fact, the ratio r_3 should approach the probability $p(G_3)$ for large M. This test was executed for values of M between 100 and 5×10^6, and the result are reported in fig. 4.3. We can see that the ratio r, represented in the y axis, approaches the theoretical probability of $1/4$, represented as a red line, as M, reported in the x axis, increases. This confirms the fact that the algorithm 4 generates graphs according to (4.3).

Figure 4.3: Behaviour of the frequency of occurrences of the graph G_3 (blue line). As the number of total graphs generated M increases. It approaches the theoretical value of $1/4$ (red line).

4.1.2 Handling subdivergences

In order to compute the amplitude A_L we focus on p-log graphs, as defined in 3.1. However, algorithm 4 can generate graphs G that have subdivergences, $\omega(G) \neq 0$ or can be not 1PI. Checking for all these three conditions implies that one has to compute the superficial degree of divergence for all the $2^{|E|}$ subgraphs of G, compute $\omega(G)$ and check the one-particle reducibility by checking for the number of connected components of E subgraphs, and if the graph turns out to not be p-log, it has to be rejected and another random graph needs to be generated. In this section we will show that by taking into account some features of period graphs, we can reduce the number of subgraphs that we need to check to $\binom{|E|}{4}$, thus effectively making the algorithm that generates random graphs faster.

Let us start by plugging eq. (1.7) and (4.7) into (2.14), and setting all the indices $\nu_e = 1$, then we get

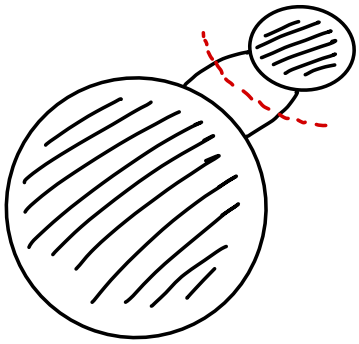
$$\omega = |E| - \frac{D}{2}L = \left(\frac{D}{N} - \frac{D}{2} + 1\right)|E| + \frac{D}{2}\left(\frac{Ext}{N} - C\right). \tag{4.11}$$

Due to the restriction to a ϕ^4 theory in 4 spacetime dimensions, we can write $D = 4$, and $N = 4$. We can then use these conditions together with the fact that we only consider connected graphs, i.e. $C = 1$, and plug them into (4.11), which gives us

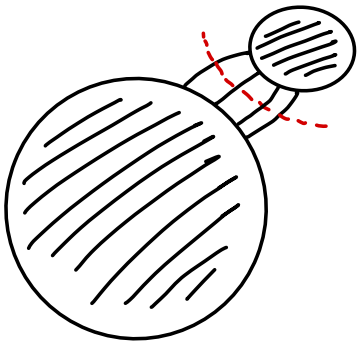
$$\omega(G) = \begin{cases} -1 \text{ if } G \text{ has 2 external legs,} \\ 0 \text{ if } G \text{ has 4 external legs,} \\ > 0 \text{ if } G \text{ has 6 or more external legs.} \end{cases} \tag{4.12}$$

This means that a graph will be log-divergent only if it has $|Ext| = 4$. This consideration limits the type of graphs that we need to generate and makes us avoid the check for $\omega(G)$.

Let us now deal with the subdivergences. We can first notice that eq. (4.11) holds for the subgraphs $\gamma \subset G$ as well, where $\omega(\gamma)$ depends on $E = E(\gamma)$, the edges of the subgraph, and $Ext = Ext(\gamma)$, the external legs of γ. The external legs $Ext(\gamma)$ are pairs of vertices where one of the two elements is $v \in V(\gamma)$ and the other is in $v' \in V^{ext}(\gamma)$. The set of external vertices of the subgraph has elements that either are external vertices of G, $V^{ext}(G)$, or vertices of G that are not in γ. Pictorially speaking, the external legs of the subgraph γ are obtained by "cutting in half" the edges that connect γ to the rest of the graph. Thus, eq. (4.12) implies the divergent subgraphs are the ones given by "cutting" 2 or 4 edges, see fig. 4.4a and 4.4b, and that are connected.



(a) Example of a linear sub-divergence.

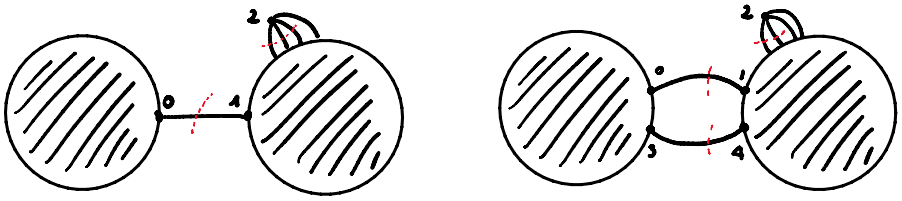


(b) Example of a logarithmic sub-divergence.

Figure 4.4: In fig. a and b all the possible divergent subgraphs that can be generated, and thus excluded from the statistical analysis.

This means that to exclude the non-primitive graphs, we need to check $\binom{|E|}{2} + \binom{|E|}{4}$ subgraphs instead of $2^{|E|}$. But we can improve this further, by noticing the fact that by checking the

log-subdivergences, we implicitly check for the linear subdivergences, i.e. the ones that have $\omega(\gamma) = -1$, and we also check that the graph is 1PI, thus making us avoid $|E|$ more checks. As explained above, when checking for all the possible log-subdivergences, one constructs all the possible subgraphs given by "cutting" 4 edges; if the graph is not 1PI, among these subgraphs there will be one similar to the one represented in fig. 4.5a: We can cut 1 edge that links the two components that become disconnected, the edge between the vertices 0 and 1 in the figure, and 3 other edges, for instance 3 of the ones that link vertex 2 to the rest of the disconnected subgraph. Thus, the graph in fig. 4.5a gets rejected just by checking for log subdivergences. This excludes $|E|$ checks for 1PI graphs. We have a similar pattern for linear subdivergences: While checking for log-divergences, 2 out of 4 cuts are performed for the edges that connect the divergent subgraphs, and thus the resulting subgraph will be disconnected, and thus it gets rejected. An example of this mechanism is represented in fig. 4.5b: one cuts the edges that connect the vertices 0, 1 and 3 and 4, while the other 2 cuts are performed for edges inside the subgraphs, for instance we can cut 2 of the 4 edges of the vertex 2, and the subgraph generated is disconnected.



(a) (b)

Figure 4.5: Representation of a non 1PI graph, in subfig. a, and of a graph with a linear subdivergence, in subfig. b. The red dashed lines represent the "cuts" that one can perform to check that the graph needs to be rejected.

There are two mainly occurring instances where divergent subgraphs are generated, shown in fig. 1.1a and 1.1b. These subgraphs are called *tadpoles* and *bubbles*, respectively. Due to the high probability of these two subgraph being generated, it was computationally favourable to check for their presence beforehand.

4.2 Distribution of Feynman integrals

Let us now discuss the results obtained. We generated 1000 random graphs for each loop order L in the range 3 to 15. The distribution of $\mathcal{P}_G$ for $L < 15$ is shown in fig. 4.7, and the distribution for $L = 15$ is represented in fig. 4.8. By looking at fig. 4.7 one can notice that for low loop orders there are few different values of $\mathcal{P}_G$, this is due to the fact that for small L there are not many p-log graphs. For instance at $L = 3$, there is only one p-log graph, represented in fig. 4.6, and thus the histogram has only one column.

The computations were run on the Euler cluster, that has nodes equipped with 2 64-core AMD EPYC 7742 CPUs, each with 512 GB of DDR4 memory clocked at 3200 MHz. The computations for $L = 15$ took ~ 7 days for 1000 graphs divided in 3 groups. Each group had assigned 10 cores, to make use of the parallelization implemented in the code attached to [7]. The evaluations for $L = 15$ were on average the longest, as there is an exponential dependence, mentioned in the previous chapter, of the runtime of each integral with the dimension of the integrand. The computations for higher loop orders were also the most resource heavy, due to the exponential scaling of the computations of the $J(G)$ table with the size of the graph, and required an average of 15 GBs of RAM.

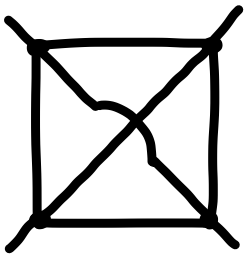


Figure 4.6: The only primitive graph for $L = 3$.

It is interesting to notice that as L increases, the distribution becomes more and more peaked around the left side of the plot, and then quickly decreases towards 0. Furthermore, the position of the peak changes almost quadratically with L, even if there does not seem to be a specific correlation between the position and the loop number.

The overall shape of the distribution can be inferred by plotting a histogram for the variable $x = \log(\mathcal{P}_G/\min \mathcal{P}_G)$, where the $\min \mathcal{P}_G$ is the minimum value of the periods for a fixed loop order. The shape of the distribution of this variable for $L = 15$ is plotted in 4.9, and it is shaped like a shifted gamma distribution:

$$f(x;\alpha,\beta,x_0) = \frac{\beta^\alpha}{\Gamma(\alpha)}(x-x_0)^{\alpha-1}e^{-\beta(x-x_0)}. \tag{4.13}$$

The parameters α and β are called *shape* and *rate*, respectively.

In order to find the best parameters of the gamma distribution to fit the plot in fig. 4.9, we used the *gamma.fit* function of the *stats* module from the library *scipy* of Python. The method implemented for fitting into the *gamma.fit* function is the *maximum likelihood estimation* (MLE) method. It is worth noting that the fit is done for normalised data, which means that the frequencies are rescaled so that the total area of the histogram is 1. The rescaling factor can be computed by multiplying the width of the bins for their height, and then summing over all the bins. To compute the covariance matrix of the parameters fitted, we can employ a result in statistics [32, Theorem 5.6] that proves that the covariance of some vector of parameters $\vec{\theta}$, obtained with the MLE method, is of the order of the inverse of the *Fisher information matrix*, defined as:

$$I(\theta)_{n,m} = -nE_\theta\left[\frac{\partial^2}{\partial\theta_n\partial\theta_m}\log f(x;\vec{\theta})\right],$$

where n is the number of occurrences of the independent occurrences x of the variable X, and E_θ is the expected value with respect to X. Thus, we can write:

$$\mathrm{cov}(\vec{\theta})_{n,m} \sim (I(\theta)^{-1})_{n,m} \tag{4.14}$$

where $\mathrm{cov}(\vec{\theta})_{n,m}$ is the covariance matrix of the parameters. The results of the fit for the parameters α, β, x_0 are:

$$\begin{aligned} \alpha &= 7.11797 \pm 0.00052 \\ \beta &= 4.245958 \pm 0.000022 \\ x_0 &= -0.216087 \pm 0.000029 \end{aligned} \tag{4.15}$$

where the parameters have covariance

$$\begin{aligned} \mathrm{cov}_{\alpha,\beta} &= 1.6 \times 10^{-7} \\ \mathrm{cov}_{\alpha,x_0} &= -5.7 \times 10^{-8} \\ \mathrm{cov}_{\beta,x_0} &= -3.4 \times 10^{-8}. \end{aligned} \tag{4.16}$$

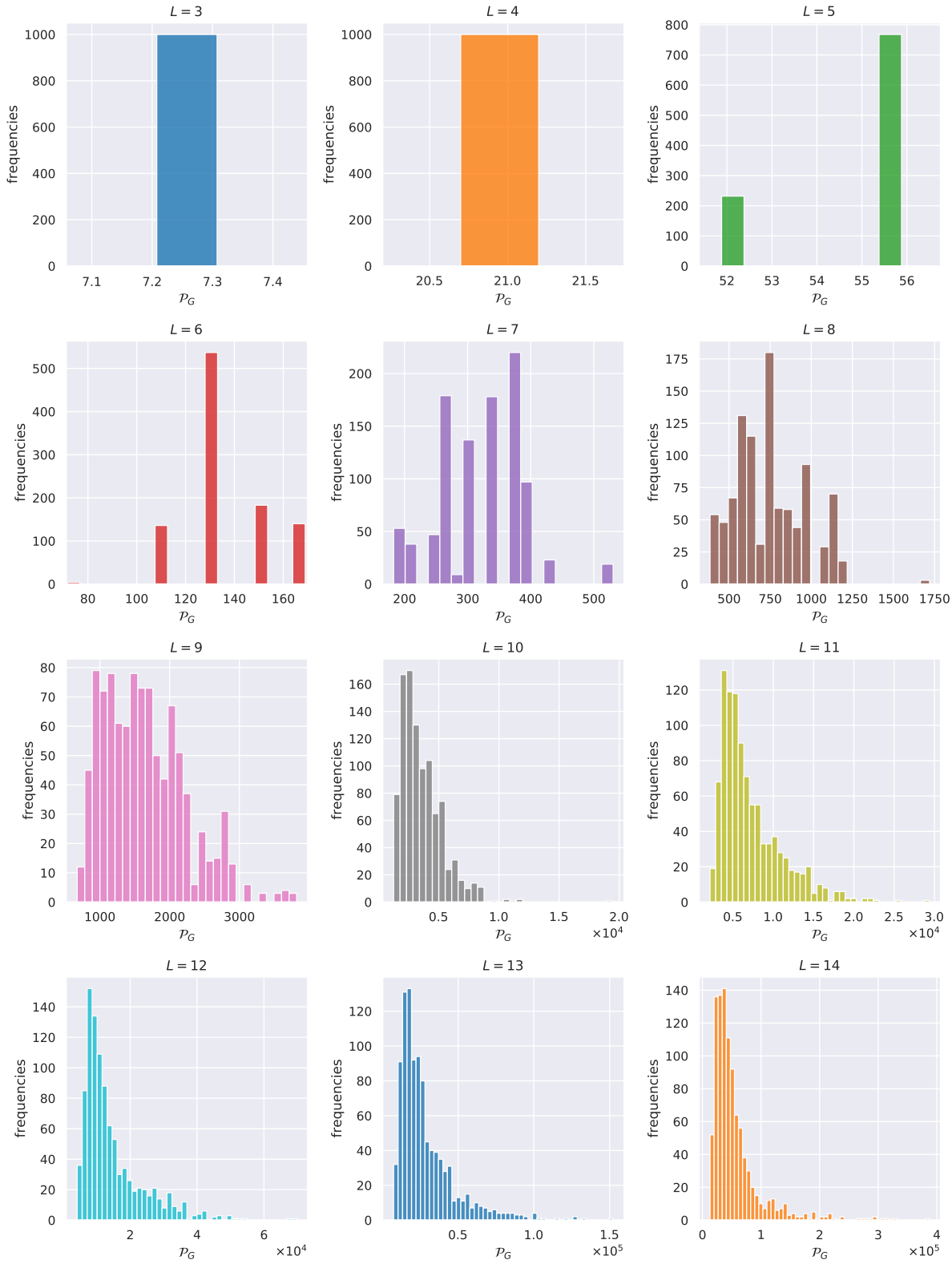


Figure 4.7: Distribution of I_G for different loop orders for primitive random graphs.

The covariance and uncertainty of the parameters was computed according to (4.14).

The resulting distribution, scaled back by multiplying $f(x;\alpha,\beta,x_0)$ by the normalisation factor mentioned above, is represented by a red line in fig. 4.9. The fit is sound, as underlined

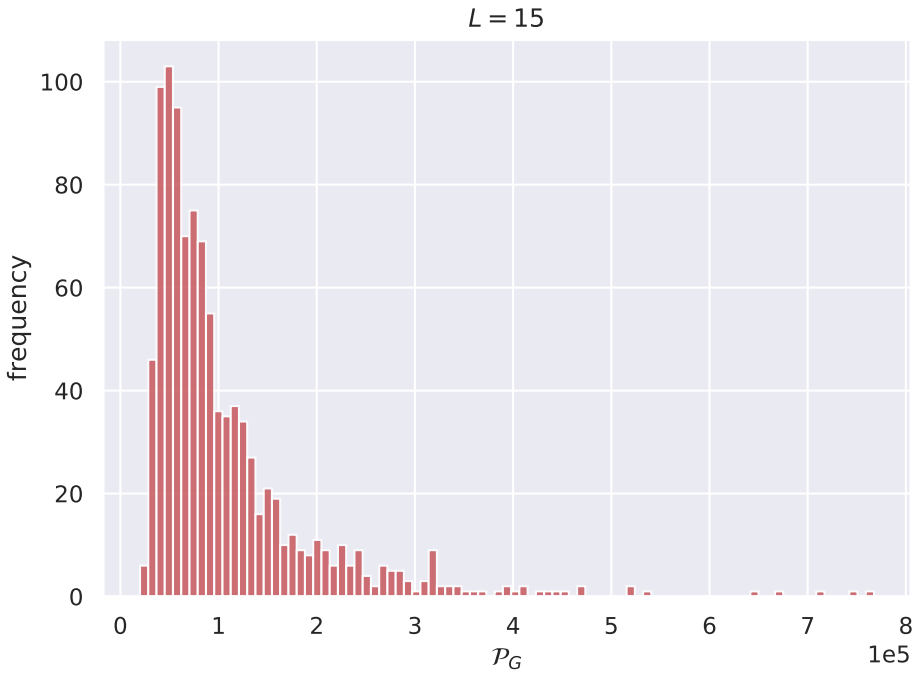


Figure 4.8: Distribution of $\mathcal{P}_G$ for $L = 15$.

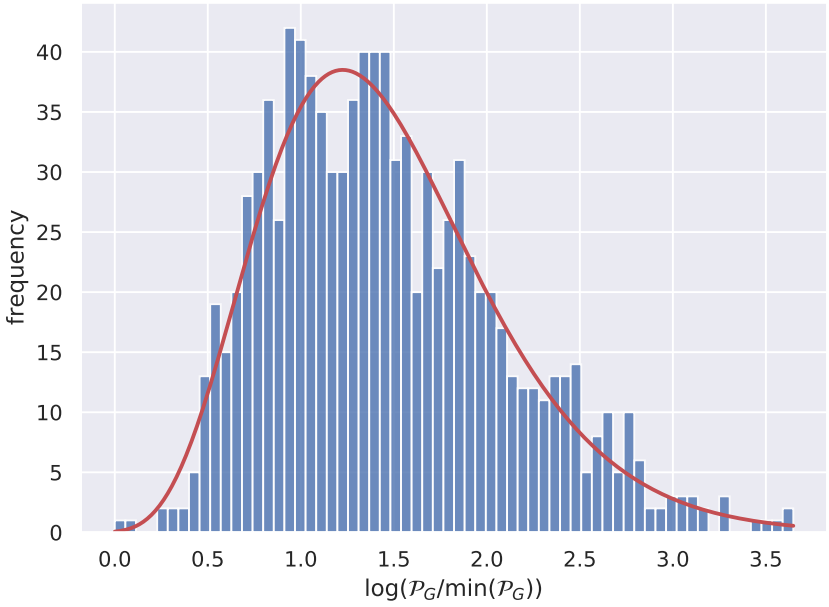


Figure 4.9: Fit of the distribution of $\mathcal{P}_G$ in logarithmic coordinates for $L = 15$.

by the value of the reduced *Pearson's* χ^2 [32]:

$$\chi^2/\nu = 1.043 \sim 1, \tag{4.17}$$

where ν are the number of degrees of freedom of the fit; in this instance $\nu = 62$, and this value is given by the difference of the number of bins (65) of the histogram in fig. 4.9 and the number of parameters (3).

4.2.1 Period and Hepp bound relation

In chapter 3 we introduced the Hepp's bound $\mathcal{H}_G$, and we showed that its computation is a crucial step in the tropical sampling algorithm. Furthermore, we mentioned that the inequality $\mathcal{P}_G \leq \mathcal{H}_G$ holds, and follows from the definition of ψ^{tr} in (3.19). In the spirit of [28], we checked for other possible correlations between $\mathcal{H}_G$ and $\mathcal{P}_G$. We were able to replicate Panzer's results for a power law fit at 7 loop orders, as it is shown in the left plot of fig. 4.10, where the position of each point is given by the value of the Period and Hepp's bound for a particular 7 loop p-log graph. The red line represents the result of the fit with the function $f(x; A, B) = Ax^B$, where the best parameters are:

$$\begin{aligned} A &= (3.985 \pm 0.018) \times 10^{-7} \\ B &= 1.34903 \pm 0.00038 \end{aligned} \tag{4.18}$$

which are compatible with Panzer's results for the same fit. We mimicked this analysis up to 15 loop orders, and noticed that the power law fit was not a good fit for $L > 12$. As an example we can see the fit with the function f, represented as a red curve, for $L = 15$ in the right plot of fig. 4.10. This same plot ($\mathcal{H}_G$ vs $\mathcal{P}_G$) can be found for all loop orders in appendix A.2. A better curve to fit the data was found to be

$$g(x; A, B, C) = A \log(x)^B + C, \tag{4.19}$$

with the parameters:

$$\begin{aligned} A &= (1.94 \pm 0.25) \times 10^{-46} \\ B &= 35.492 \pm 0.039 \\ C &= (-6.76 \pm 0.96) \times 10^2. \end{aligned} \tag{4.20}$$

However, even if this behaviour offers some improvements over the power law fitting, it still fails the χ^2 test for $L > 12$, for instance at 15 loops we have $\chi^2/\nu \simeq 21 \gg 1$, where ν are the degrees of freedom of the fit.

The parameters obtained from the fits with the functions f and g are reported in appendix in table A.1 and A.2.

As explained in [28, Section 5.1], we can also compare different loop orders by setting:

$$\xi(G) = \left(\frac{\log(\mathcal{H}_G/2)}{L(G) - 1}, \frac{\log \mathcal{P}_G}{L(G) - 1} \right), \tag{4.21}$$

where we used Panzer's notation. The points $\xi(G)$ for all the loop orders from 6 to 15 are shown in fig. 4.11. The large amount of data, consisting of 10000 points, shows a remarkable correlation, which can be confirmed by fitting the points with the curve

$$f(x; A, B) = A(\log(x))^B. \tag{4.22}$$

The curve fitting procedure results in the parameters:

$$\begin{aligned} A &= (247.908 \pm 0.015) \times 10^{-2} \\ B &= (-56.1671 \pm 0.0087) \times 10^{-2}, \end{aligned} \tag{4.23}$$

and the resulting curve is represented in fig. 4.11 by a red line. The good agreement of the fitting curve with the data is confirmed by the χ^2 test which gives a result of

$$\chi^2/\nu \simeq 0.87.$$

As Panzer outlines in his paper, the relation between the Hepp's bound and the period is not always monotone, meaning that there are some G_1, G_2 for which $\mathcal{P}(G_1) \geq \mathcal{P}(G_2)$ and $\mathcal{H}(G_1) \leq \mathcal{H}(G_2)$. Nonetheless, we showed that there seems to be a correlation between these two objects, at least up to 15 loops.

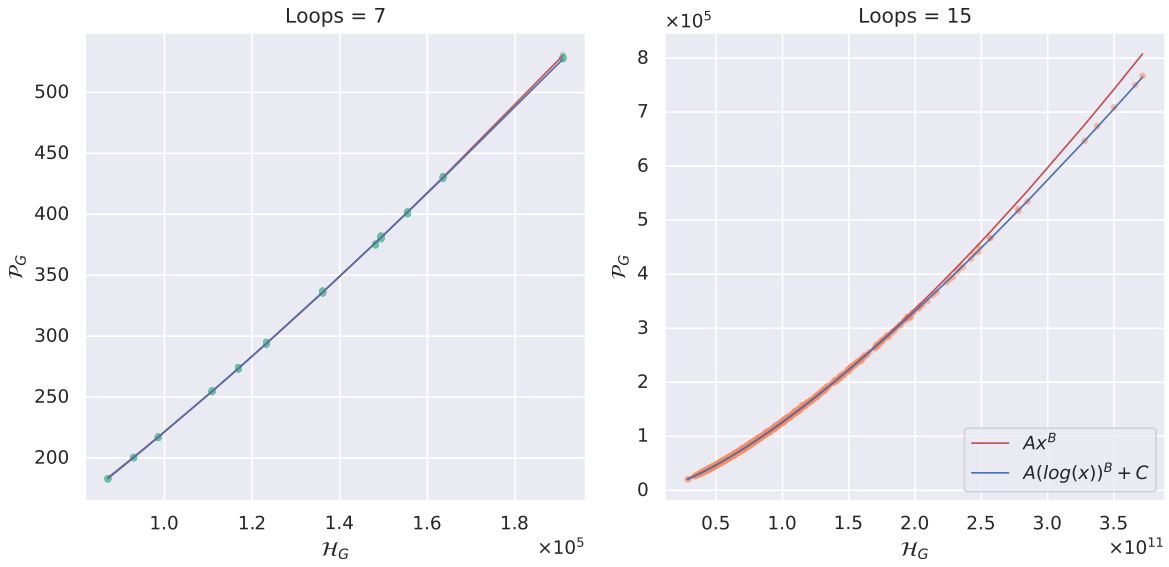


Figure 4.10: Each point has coordinates given by the value of $\mathcal{H}_G$ and $\mathcal{P}_G$ for the same graph G. In the left plot, we can see points corresponding to graphs with 7 loops and in the right plot graphs with 15 loops. The point exhibit an interesting correlation, that has a relation that can be inferred either with the power law fit (red line), or with the fit with the function $g(x; A, B, C)$ defined in eq. (4.19).

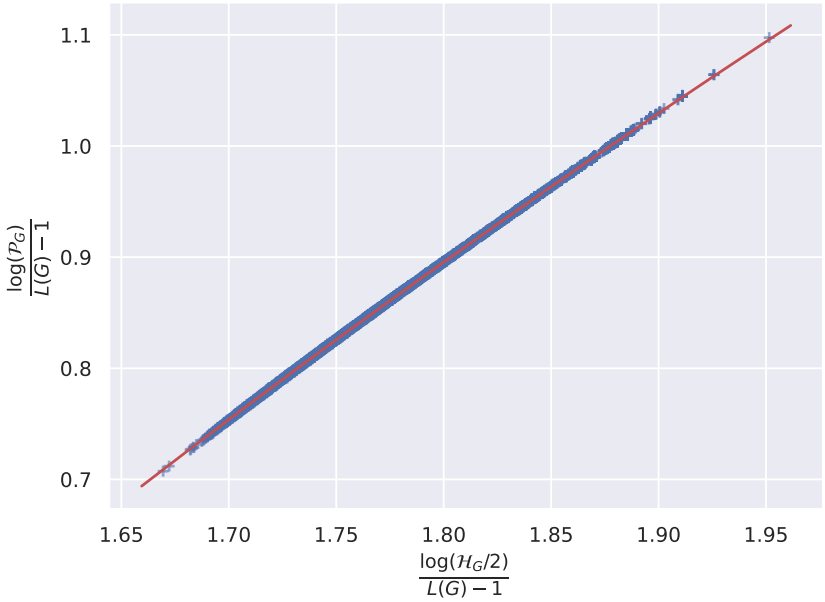


Figure 4.11: An interesting correlation between Hepp's bound and period. Each point in the graph has coordinates given by $\xi(G)$ in eq. (4.21). These points exhibit a remarkable correlation that can be empirically inferred by fitting the data with the function $f(x; A, B) = A(\log(x))^B$. The fit is shown as a red line.

4.3 Amplitude

We shall now discuss the behaviour of the expectation value of the distribution of Feynman integrals $\mathbb{E}[\mathcal{P}_G]$ and of the amplitude A_L.

We can estimate the value of $\mathbb{E}[\mathcal{P}_G]$ by using the average $I_L^{(M)}$ of the values of $\mathcal{P}_G$ weighted with the inverse squared of their uncertainty $\sigma[\mathcal{P}_G]$, in formulas

$$\mathbb{E}[\mathcal{P}_G] \simeq I_L^{(M)} = \frac{1}{\sigma_L^2} \sum_{n=1}^{M} \frac{\mathcal{P}_G^{(n)}}{\sigma[\mathcal{P}_G^{(n)}]^2}, \tag{4.24}$$

where $n = 1, \dots, M$ is an index that labels the different values of the periods related to the $M = 1000$ random graphs generates with algorithm 4, and the prefactor σ_L^2 is the variance of the estimator $I_L^{(M)}$ defined as

$$\sigma_L^2 = \sum_{n=1}^{M} \frac{1}{\sigma[\mathcal{P}_G]^2}. \tag{4.25}$$

In the remainder of the section we will not distinguish between $\mathbb{E}[\mathcal{P}_G]$ and the estimator $I_L^{(M)}$, and we will use them interchangeably.

The values $\mathbb{E}[\mathcal{P}_G]$ for $L > 9$ are represented by blue dots in fig. 4.12. Their relatively uncertainties $\sigma_L/\mathbb{E}[\mathcal{P}_G]$, with $\sigma_L = \sqrt{\sigma_L^2}$, are of the order $\sim 10^{-5}$, and they are therefore not represented.

The behaviour of $\mathbb{E}[\mathcal{P}_G]$ is relevant as it can be related to C_L, the normalisation constant of a conjectured limiting distribution of Feynman integrals for large L, in formulas

$$\lim_{L(G)\to\infty} C_{L(G)}^{-1} \int_{\mathbb{RP}_+^{|E|-1}} \Omega \frac{\prod_e x_e^{\nu_e}}{\psi_G(\mathbf{x})^{D/2}} \left(\frac{\psi_G(\mathbf{x})}{\phi_G(\mathbf{x})}\right)^{\omega(G)}, \tag{4.26}$$

where the normalisation constant C_L is defined so that it sets the expectation value of the distribution to 1.

To relate C_L and $\mathbb{E}[\mathcal{P}_G]$, we start by considering the probability of sampling a particular value I_G from the limiting distribution, which we can write as

$$p(I_G) = \left(Z_I^{(L)}\right)^{-1} \frac{I_G}{|\mathrm{Aut}(G)|},$$

where $Z_I^{(L)}$ is a normalisation constant so that $\sum_{L(G)=L} p(I_G) = 1$. Therefore, we can use eq. (1.2) to show that

$$Z_I^{(L)} = \sum_{L(G)=L} \frac{I_G}{|\mathrm{Aut}(G)|} = A_L.$$

By definition C_L normalises the expectation value of the distribution of I_G to one, which implies that

$$C_L = \frac{Z_I^{(L)}}{Z_L}.$$

Thus, by using eq.(4.1),

$$C_L = A_L/Z_L = \mathbb{E}[\mathcal{P}_G],$$

which means that we can infer the behaviour of C_L for $L \gg 1$, by studying $\mathbb{E}[\mathcal{P}_G]$ in the same large loop regime. In [7] it was outlined that it is possible to compute the value of C_L using non perturbative methods, and this yields the result:

$$C_L = \frac{4e^{-3\gamma_E}}{\sqrt{2\pi}A^6} L^{\frac{5}{2}} \left(\frac{3}{2}\right)^{L+3}, \tag{4.27}$$

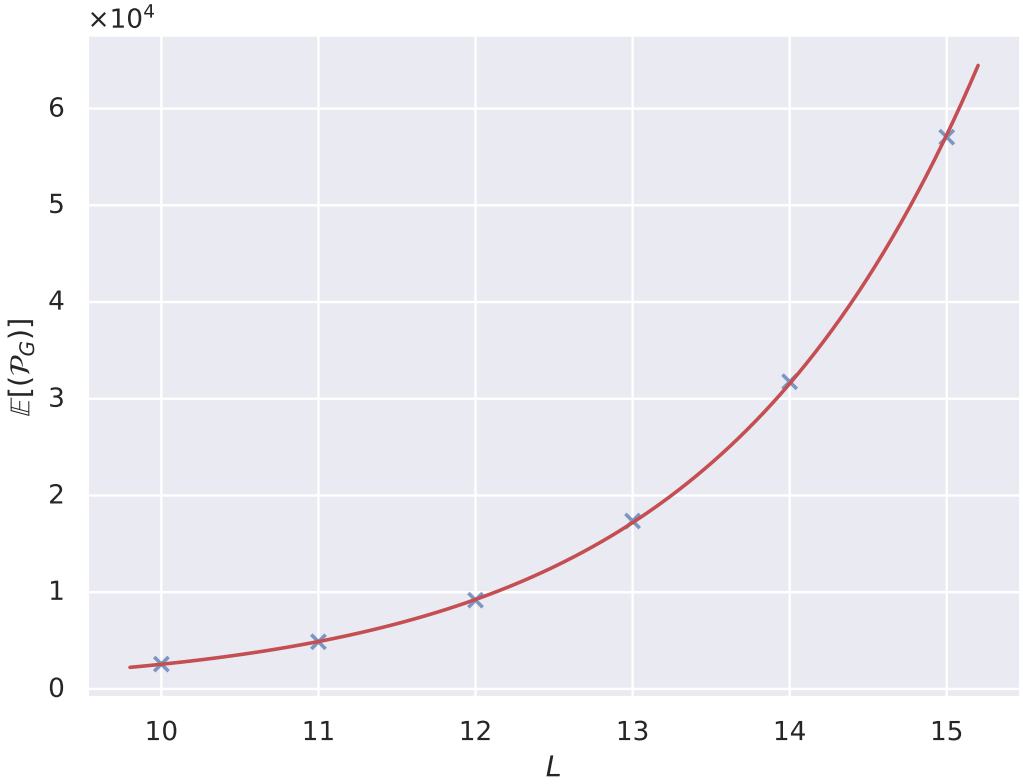


Figure 4.12: Curve fit for the numerical values of $\mathbb{E}[\mathcal{P}_G]$. The data (blue crosses) exhibits an exponential behaviour, which is qualitative confirmed by fit (red line) with the function $f(L)$ in eq. (4.28).

where γ_E is the Euler-Mascheroni constant and A is the Glaisher-Kinkelin constant. We can now compare this with the results of a fit of the points in fig. 4.12. The data was fitted according to behaviour

$$f(L; B, \delta, C) = BL^{\delta}K^{L+3} \tag{4.28}$$

which returned the following parameters:

$$\begin{aligned} \delta &= 2.17 \pm 0.51 \\ B &= 0.052 \pm 0.034 \\ K &= 1.563 \pm 0.063 \end{aligned} \tag{4.29}$$

with covariance:

$$\begin{aligned} \sigma_{\delta,B} &= -0.017 \\ \sigma_{\delta,K} &= -0.032 \\ \sigma_{B,K} &= 0.0021 \end{aligned} \tag{4.30}$$

By confronting these results with eq. (4.27), one can notice that the numerical values of the parameters are all in $\lesssim 1\sigma$ from the theoretical ones. The fit obtained is represented in fig. 4.12 by a red line.

Let us now focus on the amplitude A_L. We can compute the values of this quantity by using eq. (4.1), where Z_L is computed with (4.4). These values are shown in figure 4.13. A glance at the plot already hints to a factorially divergent behaviour, which can be precisely showed by fitting the points with the function

$$f(L; A, B, C) = BL^{A}C^{L+3}\Gamma(L+3). \tag{4.31}$$

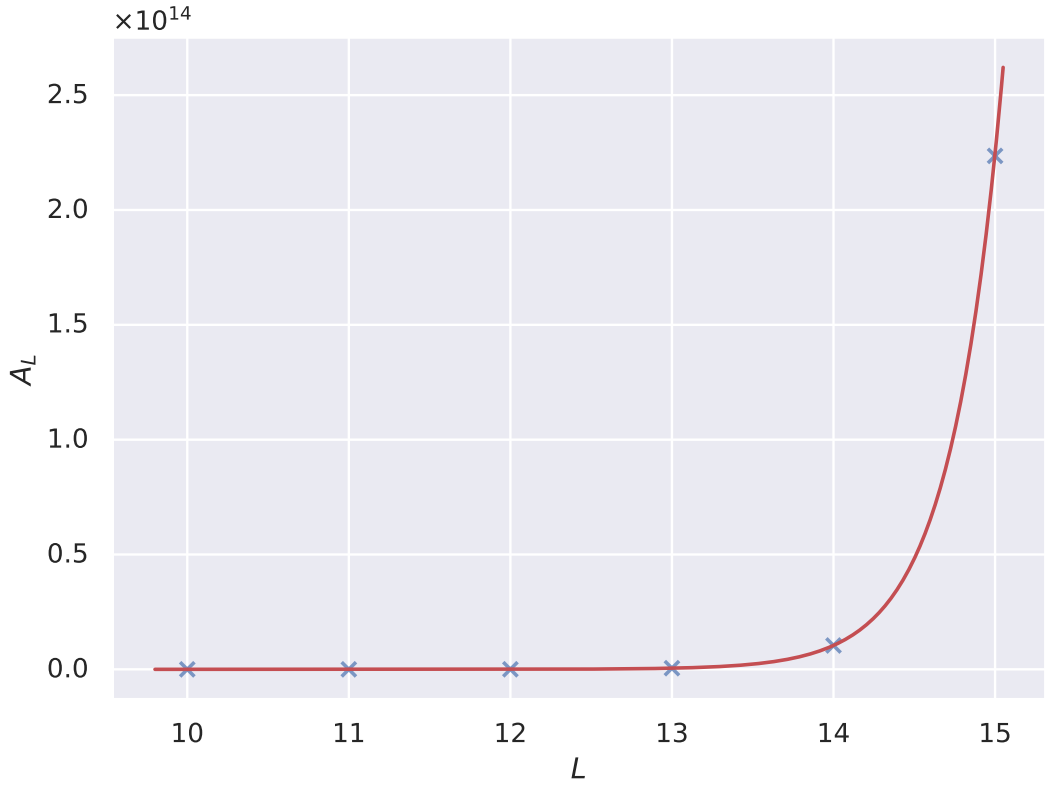


Figure 4.13: The blue crosses represent the values of the p-log loop amplitude A_L, which are given by deviding $\mathbb{E}[\mathcal{P}_G]$, in fig. 4.12, by the normalisation factor Z_L, in eq. (4.4). The results obtained were fitted with a factorially divergent behaviour, and the resulting curve is represented by a red line.

This fit, represented by a red line in fig. 4.13, returns the following results for the parameters A, B, C:

$$\begin{aligned} A &= 6.43 \pm 0.47 \\ B &= (6.8 \pm 3.9) \times 10^{-7} \\ C &= 0.815 \pm 0.032 \end{aligned} \tag{4.32}$$

these parameters have covariance:

$$\begin{aligned} \sigma_{A,B} &= -1.8 \times 10^{-7} \\ \sigma_{A,C} &= -0.015 \\ \sigma_{B,C} &= 1.2 \times 10^{-8}. \end{aligned}$$

This fit gives convincing results about the factorial divergence of the amplitude A_L. However, it is important to mention that both this and the one for $\mathbb{E}[\mathcal{P}_G]$ in fig.4.12 are to be taken qualitatively. In fact, even if the results in eq. (4.29) are in agreement with (4.27), and the factorial behaviour of the amplitude is confirmed by the parameters in eq. (4.32), the theoretical behaviour of C_L and Z_L are valid only in the large order regime, and $L \in [10, 15]$ is a range not high enough to appreciate precisely the effects that occur at $L \to \infty$.

Furthermore, we analysed the behaviour of the uncertainty related to the Monte Carlo integration by considering the uncertainty of $\mathbb{E}[\mathcal{P}_G]$, which we labelled above as σ_L. As it was discussed in sec. 3.1, the uncertainty of a Monte Carlo integral I behaves like $\sigma_I \propto \frac{1}{\sqrt{N}}$ for fixed

number of sampling points N. Hence, by linearity, this behaviour, that holds for all the period integrals $\mathcal{P}_G$, will carry over to σ_L. We can thus write:

$$\sigma_L(N) = \frac{g(L)}{\sqrt{N}}, \tag{4.33}$$

where we denoted explicitly the dependence on N, and that the proportionality constant between $\sqrt{N}$ and σ_L will in general be a function in L, that we called $g(L)$. To infer the behaviour of $g(L)$ we can fix the value of N and see how $\sigma_L[N]$ changes for different loops orders. Then, by changing the value of N, we can check if the behaviour inferred is still valid, and if the results are consistent with one other. The results of this process for different values of N are shown in appendix in fig. A.2. In each plot L takes values in the range $[3, 12]$; this range is smaller if compared to the range used in previous discussions, as the time necessary for each integration scales linearly with the number of sampling points and for each value of N and L, one needs to do 1000 integrations. However, even with a smaller number of loops, we can see that the uncertainties show a quadratic growth. Thus, we fitted the function

$$g(L) = A + BL^2, \tag{4.34}$$

where A and B are parameters. The fits for different Ns yield the results shown in appendix in tab. A.3. The coefficients obtained have values that are not sparse, and they have average and standard deviation of

$$\begin{aligned} \bar{A} &= (2.3146 \pm 0.0014) \times 10^{-2} \\ \bar{B} &= (7.304 \pm 0.023) \times 10^{-4} \end{aligned} \tag{4.35}$$

where $\bar{A}$, $\bar{B}$ indicates the standard deviation for the parameters A and B respectively. These values, compared with the uncertainty of each parameter, indicate a solid agreement of the different fits.

We can take this analysis further by noticing that from eq. (4.33), if we rescale the points $\sigma_L(N)$ in fig. A.2 by $\sqrt{N}$, the data should all follow the $g(L)$ behaviour. The rescaled data is shown in fig. 4.14, by dots with different colours for each value of N. As expected, we can see that it is very hard to differentiate between the points that have same values of L and correspond to different Ns. We can also fit the parameters of $g(L)$ in (4.34), and we get

$$A^{fit} = (2.315 \pm 0.014) \times 10^{-2} \tag{4.36}$$

$$B^{fit} = (7.304 \pm 0.019) \times 10^{-4} \tag{4.37}$$

with covariance

$$\sigma_{A,B} = -2.3 \times 10^{-10}. \tag{4.38}$$

We can notice that these results are quantitatively similar to the ones in tab. A.3 and to their mean in eq. (4.35), which hints to the soundness of the inferred behaviour.

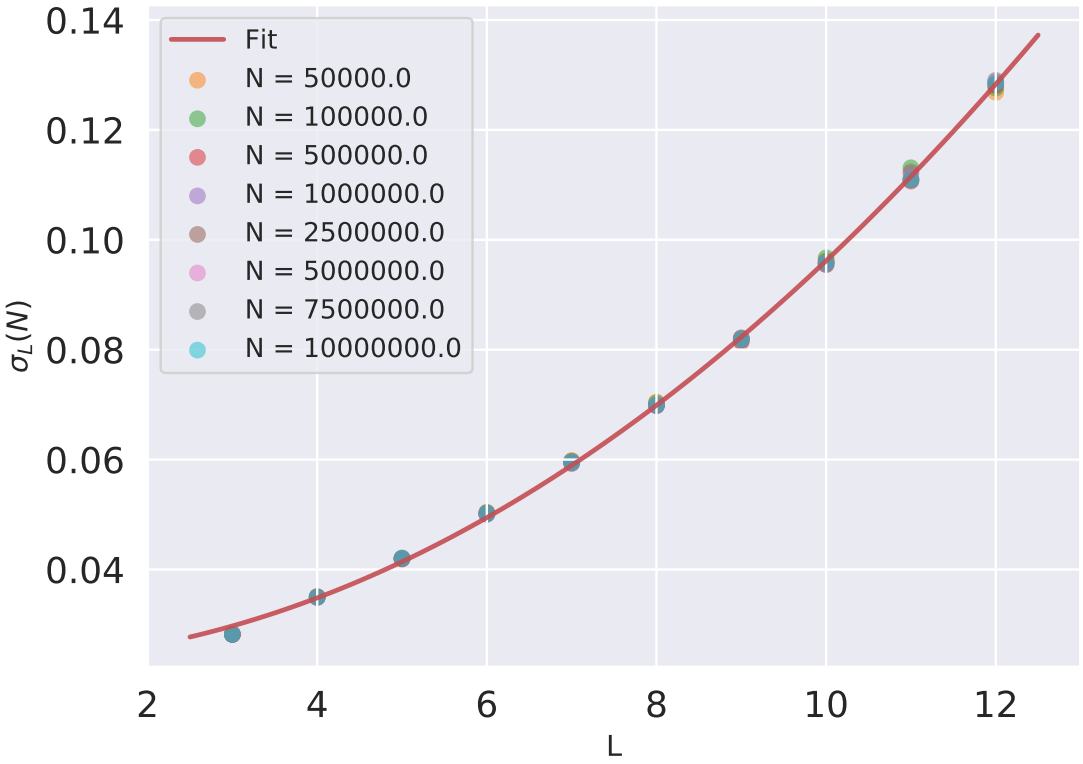


Figure 4.14: Uncertainties σ_L rescaled by $\sqrt{N}$, for different values of N. In red a fit with the function $g(L)$ that has a quadratic dependence on the loop order.

Chapter 5

Conclusions and outlook

The goal of this thesis was to study numerical techniques to integrate Feynman integrals, in particular Feynman periods, and analyse their main statistical properties at high loop orders.

In order to tackle the problem of numerical integration, we needed a form of the Feynman integral that was favourable for this type of problem. Thus, in chapter 2 we manipulated the Feynman integral to derive first the parametric representation, which is one of the most common starting points for modern applications, and then the projective representation, that thanks to the properties of projective geometry, allowed us to write compactly the Feynman integral while explicitly showcasing some of its most relevant geometric properties.

In chapter 3, we showed that Monte Carlo integration fails for functions that are not square integrable, which can be the case for Feynman integrals, as the integrand can have poles in the integration domain. The solution to this problem is a different sampling algorithm, inspired by tropical geometry, which allows us to numerically compute period integrals and get finite results. A further improvement of this algorithm was also discussed, which guarantees an exponential, rather than factorial, scaling of the problem with the dimensionality of the problem.

The tropical sampling algorithm explained, allowed us to integrate Feynman periods efficiently at high loop orders, meaning that we were able to study statistical properties of these integrals at these orders. We choose specifically the ϕ^4 QFT in 4 space-time dimensions, due to the high amount of data and analytical properties available in the literature for this theory. By generating random graphs with an algorithm inspired by a combinatorial definition of a graph, we managed to estimate the p-log scattering amplitude up to 15 loop order. The quantity computed is particularly relevant as it is proportional to the contribution of primitive graphs to the beta function, which is conjectured to approximate the growth of the full beta function for large orders. These computations showed a factorially divergent behaviour, that seems to corroborate the recent computations [24] and analysis of beta function contributions [17] for a ϕ^4 QFT. Furthermore, we have found the shifted gamma distribution in eq. (4.13) to be a possible distribution of values of period integrals, at least up to 15 loop orders, with a very good numerical accuracy. In addition, we showed that the large order behaviour of the normalisation constant of a hypothetical limiting distribution of Feynman integrals reported in [7], holds up to $L = 15$. We also expanded some results of [28], by showing that the power law correlation for the Hepp's bound and period shown in Panzer's paper does not hold to higher loop, but the correlation between $\mathcal{H}_G$ and $\mathcal{P}_G$, both rescaled by a factor that depends on the loop order, holds up to 15 loop order and follows a logarithmic behaviour. Finally, we showed that the uncertainty of the period computed with the tropical integration, follows a quadratic behaviour.

Further data at higher loop order would be beneficial to confirm all these result, in particular the shape of the inferred behaviour of the distribution of Feynman periods. Considering that in [7] it is shown that computations of Feynman integrals are feasible until 17 loop order, further improvements in this could be achieved with more computation time. In particular for this loop

order a quick calculation shows that this computation would take less than 2 weeks, splitting the 1000 integrals in 5 groups, with each integration taking 256 GB of RAM and on average 11 hours. The large amount of memory required for running the 5 groups in parallel (1.28 TB) is the biggest bottleneck for this solution.

A.1 Gaussian Integrals

Gaussian integrals are arguably one of the most important tools in quantum field theory calculations. The basic one dimensional Gaussian Integral is:

$$\int_{\mathbb{R}} dx\, e^{-x^2} = \sqrt{\pi}. \tag{A.1}$$

We can then perform rescaling $x \to \sqrt{a}(x+b)$, which gives us:

$$\int_{\mathbb{R}} dx\, e^{-a(x+b)^2} = \sqrt{\frac{\pi}{a}}. \tag{A.2}$$

This expression can easily generalised to the case of N dimensional vectors. Let $x \in \mathbb{R}^N$ and $A \in \mathrm{Mat}_N(\mathbb{R})$ positive definite. Then the following result holds:

$$\int_{\mathbb{R}^N} d^N x\, \exp\left\{-\frac{1}{2}x^T A x\right\} = \sqrt{\frac{(2\pi)^N}{\det A}}. \tag{A.3}$$

If we introduce a "source term" $J^T \cdot x$ where J is an N dimensional vector, we need to complete the square in the exponent by doing the following:

$$-\frac{1}{2}x^T A x + J^T x = \frac{1}{2}x^T A x + (A^{-1}J)^T A x - \frac{1}{2}(A^{-1}J)^T A A^{-1} J + \frac{1}{2}J^T A^{-1} J = \tag{A.4}$$

$$= -\frac{1}{2}(x - A^{-1}J)^T A (x - A^{-1}J) + \frac{1}{2}J^T A^{-1} J. \tag{A.5}$$

By doing the substitution $x \to y = x - A^{-1}J$ in the integral, gives us:

$$\int_{\mathbb{R}^N} d^N x\, \exp\left\{-\frac{1}{2}x^T A x + J^T x\right\} = e^{\frac{1}{2}J^T A^{-1} J} \int_{\mathbb{R}^N} d^N y\, e^{-\frac{1}{2}y^T A y} = \sqrt{\frac{(2\pi)^N}{\det A}} e^{\frac{1}{2}J^T A^{-1} J}. \tag{A.6}$$

A.2 Period vs Hepp Bound plot for different loop orders

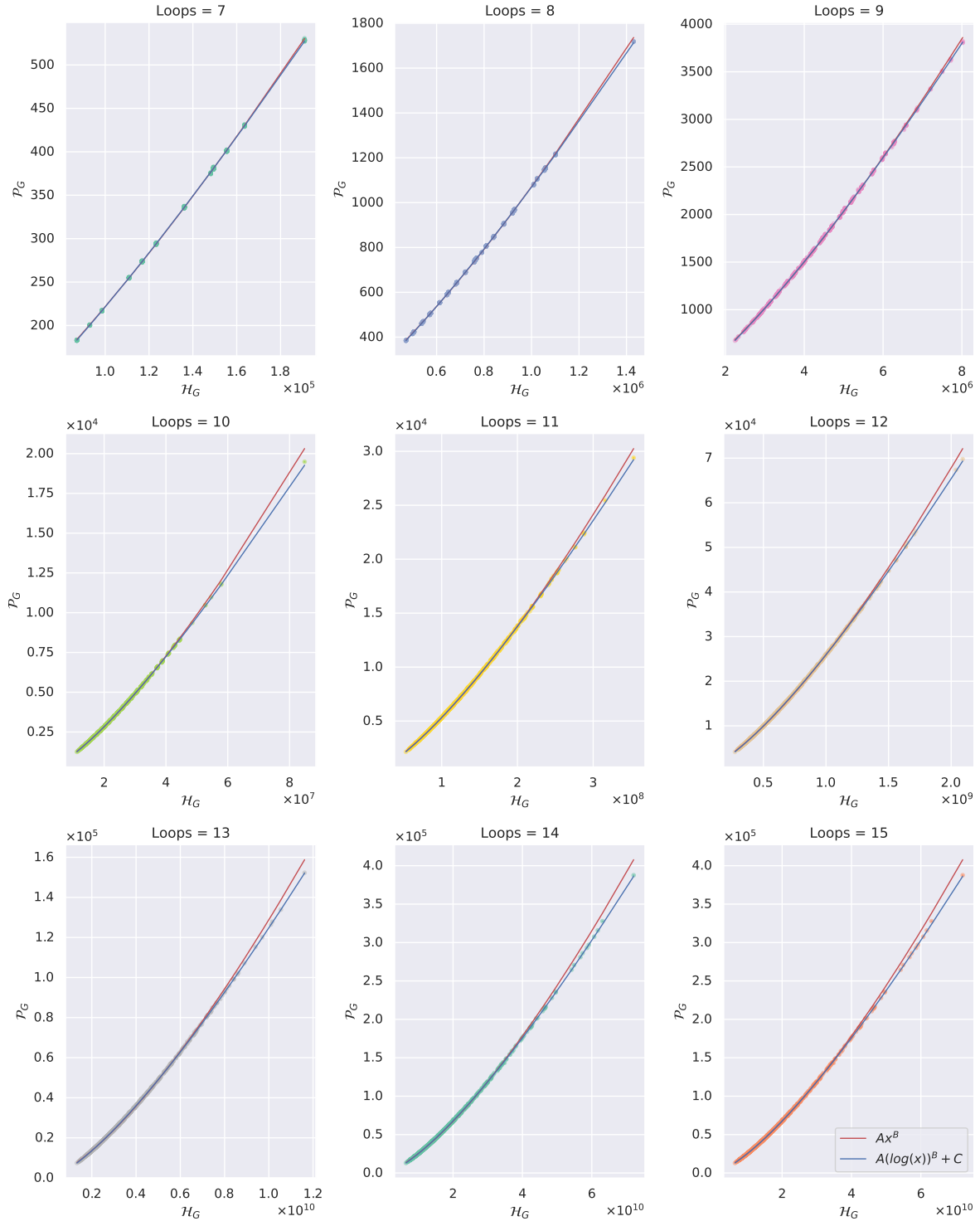


Figure A.1: Plot of $\mathcal{H}_G$ vs $\mathcal{P}_G$. In each plot we can see graphs for loop orders from 3 to 15. All the graphs seem to exhibit a correlation, which for 12 loop orders onwards is neither a power law (red line), nor a $A\log(x)^B + C$ behaviour (blue line).

L	a	σ_a	b	σ_b
7	3.9848×10^{-5}	1.8×10^{-7}	1.3490	3.8×10^{-4}
8	8.3227×10^{-6}	4.3×10^{-8}	1.3516	3.8×10^{-4}
9	1.6235×10^{-6}	1.1×10^{-8}	1.3580	4.4×10^{-4}
10	3.0297×10^{-7}	2.7×10^{-9}	1.3655	5.2×10^{-4}
11	5.0724×10^{-8}	5.4×10^{-10}	1.3775	5.8×10^{-4}
12	8.7522×10^{-9}	1.1×10^{-10}	1.3858	6.2×10^{-4}
13	1.4063×10^{-9}	2.1×10^{-11}	1.3962	6.9×10^{-4}
14	2.2276×10^{-10}	4.3×10^{-12}	1.4057	8.2×10^{-4}
15	3.2601×10^{-11}	6.8×10^{-13}	1.4169	8.3×10^{-4}

Table A.1: Table of the parameters obtained by fitting the function Ax^B to the period and Hepp's bound. The data and the curve obtained are displayed in fig. A.1

L	A	σ_A	B	σ_B	C	σ_C
7	1.399×10^{-15}	0.82×10^{-16}	16.1954	2.3×10^{-2}	6.45	0.42
8	6.459×10^{-19}	0.38×10^{-19}	18.6027	2.2×10^{-2}	12.25	0.80
9	2.634×10^{-22}	0.18×10^{-23}	20.9420	2.5×10^{-2}	18.0	1.8
10	7.033×10^{-26}	0.46×10^{-27}	23.3366	2.3×10^{-2}	22.4	3.1
11	1.679×10^{-29}	0.13×10^{-30}	25.6863	2.6×10^{-2}	5.6	6.3
12	2.867×10^{-33}	0.22×10^{-34}	28.0731	2.5×10^{-2}	-39	11
13	3.109×10^{-37}	0.29×10^{-38}	30.5412	3.0×10^{-2}	-97	23
14	2.915×10^{-41}	0.32×10^{-42}	32.9929	3.4×10^{-2}	-329	48
15	1.941×10^{-45}	0.25×10^{-46}	35.4922	3.9×10^{-2}	-676	95

Table A.2: Table of the parameters obtained by fitting the function $A\log(x)^B + C$ to the period and Hepp's bound. The data and the curve obtained are displayed in fig. A.1

A.3 Results for fits of the uncertainty

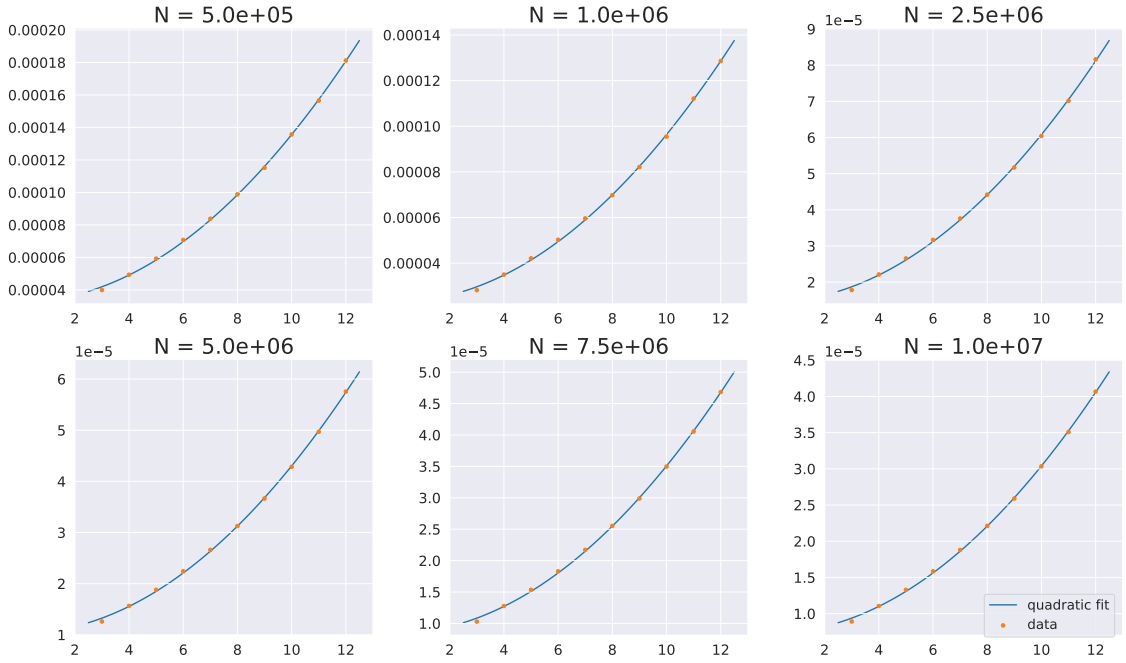


Figure A.2: Fit for $\sigma_L[N]$.

N $[\times 10^4]$	$A[\times 10^{-2}]$	$\sigma_A[\times 10^{-2}]$	$B[\times 10^{-4}]$	$\sigma_B[\times 10^{-4}]$	$\sigma^2_{A,B}[\times 10^{-9}]$
5	2.3468	0.054	7.370	0.069	-3.1
10	2.3196	0.053	7.337	0.068	-3.0
50	2.3091	0.040	7.279	0.051	-1.7
100	2.3054	0.044	7.328	0.057	-2.1
250	2.3026	0.044	7.311	0.057	-2.1
500	2.3082	0.040	7.311	0.051	-1.7
750	2.3147	0.039	7.291	0.051	-1.7
10000	2.310	0.040	7.303	0.051	-1.7

Table A.3: Table of results for the fits in fig. A.2.

Bibliography

[1] C. Anastasiou, D. P. L. Bragança, L. Senatore, and H. Zheng. *Efficiently evaluating loop integrals in the EFTofLSS using QFT integrals with massive propagators*, dec 2022.

[2] N. Arkani-Hamed, Y. Bai, and T. Lam. *Positive geometries and canonical forms.* Journal of High Energy Physics, nov 2017.

[3] M. Bergere and Y.-M. P. Lam. *Bogolubov–Parasiuk theorem in the α-parametric representation.* Journal of Mathematical Physics, 17:1546–1557, 1976.

[4] T. Binoth and G. Heinrich. *An automatized algorithm to compute infrared divergent multi-loop integrals.* Nuclear Physics B, 585(3):741–759, oct 2000.

[5] C. Bogner and S. Weinzierl. *Feynman graph polynomials.* International Journal of Modern Physics A, 25(13):2585–2618, 2010.

[6] M. Borinsky. *Renormalized asymptotic enumeration of Feynman diagrams.* Annals of Physics, 385:95–135, oct 2017.

[7] M. Borinsky. *Tropical Monte Carlo quadrature for Feynman integrals*, 2020.

[8] M. Borinsky. personal communication, 2023-03-29.

[9] M. Borinsky and O. Schnetz. *Recursive computation of Feynman periods.* Journal of High Energy Physics, 2022(8):291, Aug 2022.

[10] J. B. Boyling. *A homological approach to parametric feynman integrals.* Nuovo Cim. (10), 53A: 351-75(Jan. 21, 1968)., 1 1968.

[11] D. Broadhurst and D. Kreimer. *Knots and Numbers in ϕ^4 Theory to 7 Loops and Beyond.* International Journal of Modern Physics C, 06(04):519–524, 1995.

[12] F. Brown. *On the periods of some Feynman integrals.* preprint, 2009.

[13] F. Brown. *Invariant Differential Forms on Complexes of Graphs and Feynman Integrals.* SIGMA, 17:103, 2021.

[14] F. Brown and O. Schnetz. *Single-valued multiple polylogarithms and a proof of the zig–zag conjecture.* J. Number Theor., 148:478–506, 2015.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition.* The MIT Press, 3rd edition, 2009.

[16] R. Diestel. *Planar Graphs*, pages 89–118. Springer Berlin Heidelberg, Berlin, Heidelberg, 2017.

[17] G. V. Dunne and M. Meynig. *Instantons or renormalons? Remarks on $\phi^4_{d=4}$ theory in the MS scheme.* Phys. Rev. D, 105:025019, Jan 2022.

[18] S. Gorishny, S. Larin, and F. Tkachov. *The algorithm for OPE coefficient functions in the MS scheme.* Physics Letters B, 124(3):217–220, 1983.

[19] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods.* Springer Netherlands, Dordrecht, 1964.

[20] G. Heinrich. *Collider physics at the precision frontier.* Physics Reports, 922:1–69, aug 2021.

[21] K. Hepp. *Proof of the Bogoliubov-Parasiuk theorem on renormalization.* Communications in Mathematical Physics, 2(1):301–326, Dec 1966.

[22] M. Joswig, D. Maclagan, and B. Sturmfels. *"Introduction to Tropical Geometry".* Jahresbericht der Deutschen Mathematiker-Vereinigung, 118(3):233–237, Sep 2016.

[23] T. Kaneko and T. Ueda. *A geometric method of sector decomposition.* Computer Physics Communications, 181(8):1352–1361, aug 2010.

[24] M. V. Kompaniets and E. Panzer. *Minimally subtracted six-loop renormalization of $O(n)-$symmetric ϕ^4 theory and critical exponents.* Physical Review D, 96(3), aug 2017.

[25] A. Kotikov. *Differential equations method. New technique for massive Feynman diagram calculation.* Physics Letters B, 254(1):158–164, 1991.

[26] J. B. Kruskal. *On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem.* Proceedings of the American Mathematical Society, 7(1):48–50, 1956.

[27] E. Panzer. *Feynman integrals and hyperlogarithms*, 2015.

[28] E. Panzer. *Hepp's bound for Feynman graphs and matroids.* Annales de l'Institut Henri Poincaré D, 2019.

[29] M. E. Peskin and D. V. Schroeder. *An Introduction to quantum field theory.* Addison-Wesley, Reading, USA, 1995.

[30] E. Remiddi. *Differential equations for Feynman graph amplitudes.* Il Nuovo Cimento A, 110(12):1435–1452, dec 1997.

[31] J. Richter-Gebert. *Calculating with Points on Lines.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[32] R. J. Rossi. *Likelihood-based Estimation*, chapter 5, pages 223–279. John Wiley & Sons, Ltd, 2018.

[33] V. A. Smirnov. *Analytic Tools for Feynman Integrals.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[34] E. R. Speer and M. J. Westwater. *Generic Feynman amplitudes.* Annales de l'institut Henri Poincaré. Section A, Physique Théorique, 14(1):1–55, 1971.

[35] D. A. Spielman and S.-H. Teng. *Nearly-Linear Time Algorithms for Preconditioning and Solving Symmetric, Diagonally Dominant Linear Systems*, 2012.

[36] G. 't Hooft. *Dimensional regularization and the renormalization group.* Nuclear Physics B, 61:455–468, 1973.

[37] G. 't Hooft and M. Veltman. *Regularization and renormalization of gauge fields.* Nuclear Physics B, 44(1):189–213, 1972.

[38] S. Weinberg. *High-Energy Behavior in Quantum Field Theory.* Phys. Rev., 118:838–849, May 1960.

[39] S. Weinzierl. *Feynman Integrals: A Comprehensive Treatment for Students and Researchers*. Springer International Publishing, 2022.