12-5-2023 2:30 PM

High-performance computing in covariant Loop Quantum Gravity

Pietropaolo Frisoni, *The University of Western Ontario*

Supervisor: Vidotto, Francesca, *The University of Western Ontario*
Co-Supervisor: Christensen, Daniel, *The University of Western Ontario*
A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Physics

Abstract

This Ph.D. thesis presents a compilation of the scientific papers I published over the last three years during my Ph.D. in loop quantum gravity (LQG). First, we comprehensively introduce spinfoam calculations with a practical pedagogical paper. We highlight LQG's unique features and mathematical formalism and emphasize the computational complexities associated with its calculations. The subsequent articles delve into specific aspects of employing high-performance computing (HPC) in LQG research. We discuss the results obtained by applying numerical methods to studying spinfoams' infrared divergences, or "bubbles". This research direction is crucial to define the continuum limit of LQG properly. We investigate the self-energy diagram in LQG, analyzing the scaling of the divergence of the associated amplitude. Using the same technique, we compute the spinfoam amplitudes of a class of two-vertex diagrams. Besides divergent graphs, our investigations yield striking and surprising numerical evidence that spinfoam-containing bubbles can have finite transition amplitudes. Furthermore, we adapt Monte Carlo methods to the spinfoam formalism. We employ this technique to analyze the vertex renormalization amplitude. We find numerical solid indications that this amplitude is convergent, opening new perspectives for renormalizing large-volume infrared spinfoam bubbles. In spinfoam cosmology, we investigate the integration of HPC with Markov Chain Monte Carlo simulations, proving the potential to analyze the macroscopic properties of quantum spacetime. We perform a spinfoam refinement process starting from the simplest diagram, demonstrating the effectiveness of this hybrid approach and elucidating the connection of LQG observables with spacetime geometry. Furthermore, we apply the same technique to investigate the spinfoam with a 16-cell boundary using a topological model. Finally, we outline a numerical algorithm to compute the transition amplitude from a black hole to a "white hole". The recently proposed hypothetical decay process via gravitational quantum tunneling is one of the most intriguing hypotheses on the future of black holes. We use the spinfoam approach and HPC to investigate this phenomenon by explicitly computing the associated transition amplitude. The advancements of HPC-assisted LQG research will hopefully enable the study of complex gravitational phenomena at unprecedented scales, paving the way for exploring previously inaccessible physical regimes.

Keywords: High-performance computing, Supercomputing, Loop Quantum Gravity, Spinfoam, Monte Carlo, MCMC, Infrared divergences, Spinfoam bubbles, Spinfoam cosmology, Black hole, White hole, Black-to-white transition.

Summary for the lay audience

Loop quantum gravity is a quantum theory of gravity that offers a promising and compelling framework for understanding the fundamental nature of spacetime. We need a complete non-perturbative description of quantum gravity as we still do not know what happens in certain universe regions, such as objects that fall into black holes. At the same time, we need such a theory to reconstruct the early primordial instants of the universe. We must consider general relativity and quantum mechanics together to describe these physical scenarios. LQG is one of the most successful attempts to consistently incorporate such theories into a cohesive framework. It aims to describe the quantum behavior of the gravitational field rather than being a "theory of everything". It can be approached using two different formalisms: canonical and covariant. The covariant route, usually called the "spinfoam" approach to LQG, is a concrete framework where numerical and analytical calculations of transition amplitudes are possible. My contributions have mainly focused on developing new high-performance computing (HPC) methods relying on spinfoam formalism. Calculations in such a framework are pretty complex, and leveraging the capabilities of HPC systems becomes essential to overcome these challenges and facilitate groundbreaking investigations. Extensive research has been conducted, and this thesis aims to provide a cohesive overview of the advancements and future directions. The potential for HPC-assisted LQG offers new avenues for astrophysics, cosmology, and quantum gravity phenomenology. In summary, this collection of articles demonstrates the role of numerical methods in pushing the boundaries of LQG research.

Co-Authorship Statement

According to the current Western University policies for integrated-article format Ph.D. thesis, I include a co-authorship statement at the beginning of each Chapter, which presents a paper I authored or co-authored. Expressly, I must point out my contributions and estimate the percentage of the work conducted solely by me. In case my co-authors ever read this thesis, I apologize if they find this practice disagreeable, but I must follow the policies. I am immensely grateful to everyone who has worked with me during my Ph.D. I believe cooperation almost always leads to the best results. Interacting and sharing my research with others gave me invaluable benefits and contributions. Finally, in each co-authorship statement, I also included brief historical excursions on how the research articles were born and where the ideas that led to the final publication of the latter originated.

Acknowlegements

I thank my family for their unconditional emotional support. I acknowledge my Ph.D. advisor, Francesca Vidotto, for giving me the opportunity I was eager for and total intellectual freedom throughout my Ph.D. I thank Francesco Gozzini, Pietro Dona, Carlo Rovelli, and Francesca Vidotto for the invaluable exchanges, discussions, and guidance in completing my Ph.D. I am grateful to my colleagues Farshid Soltani, Mateo Pascual, Yichen Luo, Sina Kazemian, Jared Wogan, Pascal Rodriguez, and all the graduate students of the P&A department for the time I spent with them during my years at Western. I thank the members of my ACM committee, Francesca Vidotto, Dan Christensen, Els Peeters, and Martin Houde, for the precious pieces of advice provided during my Ph.D. I am especially grateful to my co-supervisor, Dan Christensen. I acknowledge the Compute Canada and Sharcnet staff for the training with the superclusters and scientific computing. Finally, I am grateful to all the staff of Western University (especially the volunteers of IESC) for the excellent service I received.

Contents

List of Figures

List of Tables

List of Appendices

Introduction

Loop quantum gravity (LQG) is a theory that attempts to combine quantum mechanics and general relativity to provide a quantum description of gravity at its most fundamental level. This implies describing spacetime. For a formal introduction to LQG, we refer to the numerous reviews and books currently available in the literature. Although there are too many to cite them all, a few significant examples are [1, 2, 3, 4, 5, 6, 7, 8, 9]. For the interested reader, I recently posted on ArXiv the full transcript of Carlo Rovelli's lectures on Loop Quantum Gravity, given in Marseille in 2018 [10]. LQG quantizes the spacetime geometry, treating it as a granular structure at the Planck scale. Being a quantum theory, the generic structure of LQG can be described with three core elements: the Hilbert space, an algebra of observables, and the dynamics. The Hilbert space represents the auxiliary structure in which quantum states of the gravitational field are defined. The basis states in the Hilbert space are spin network states, which are graphs with edges labeled by spins. These spins encode the quantum properties of the geometry, offering a discrete description of space. Observables are represented by operators corresponding to various geometric and physical quantities. These operators act on the Hilbert space of states and are associated with quantities like area, volume, length, and curvature. They are quantized in LQG [11], leading to discrete spectra due to the discrete nature of spacetime at the Planck scale. The algebra of observables in LQG reflects the non-commutative nature of quantum geometry. Finally, the dynamics can be described using two main different formalisms: canonical and covariant.

The canonical approach to LQG quantizes gravity by starting from the Arnowitt-Deser-Misner (ADM) formulation of general relativity [12, 13] and quantizing the canonical variables. The dynamics is encoded in a Hamiltonian constraint that generates the evolution of the states. The covariant approach, instead, usually starts from a discrete path integral formulation. It describes gravity's quantum behavior by considering all possible histories (spacetime configurations) and their associated amplitudes. The dynamics is defined by transition amplitudes between quantum states of the gravitational field, governed by a sum over intermediate states and related to the action of classical general relativity. These amplitudes represent the probability of transitioning from one gravitational field configuration to another. The path integral approach to LQG is often realized through spinfoam models. A spinfoam can be intuitively thought of as a higher-dimensional analog of a Feynman diagram used in standard quantum field theory. It is a way to represent a particular spacetime history, describing the transitions between different three-dimensional slices over a fourth dimension. The path integral of spinfoam theory is regularized on a four-dimensional triangulation, more precisely, on its dual structure, called "2-complex", to truncate the degrees of freedom. The fundamental objects of the 2-complex are vertices, edges, and faces, each with an associated elementary

amplitude. The covariant route is usually called the "spinfoam approach" to LQG. This Ph.D. thesis focuses on the spinfoam approach.

Another fundamental principle in LQG is diffeomorphism invariance [1]. General relativity is invariant under diffeomorphisms, reflecting that the laws of physics should be independent of the choice of coordinates or foliations of spacetime. LQG is a background-independent theory that does not rely on an external spacetime metric. Instead, it treats spacetime as a dynamic entity, allowing for the quantization of geometry and gravitational interactions without assuming a fixed background. LQG maintains this diffeomorphism invariance at the quantum level. Physical states in the LQG Hilbert space transform to respect diffeomorphism invariance, ensuring that the physical content of the theory remains unchanged.

While it is still a developing field and faces significant theoretical and mathematical challenges, there have been several promising physical scenarios where LQG is employed to study the behavior of the universe and its fundamental constituents, such as cosmology [14, 15, 16, 17] and black hole physics [18, 19, 20]. Apart from the previously mentioned quantization of area and volume, some of the most striking predictions of LQG are the calculation of black hole entropy [21] and the possibility of a cosmological big bounce [22].

Several different models are proposed in the spinfoam literature, such as the Barrett-Crane [23] model, topological BF model [24], etc. Each spinfoam model usually represents a different way of formulating and quantizing the gravitational field, providing distinct mathematical and conceptual frameworks. The existence of multiple models partially stems from the challenges of fully understanding and quantizing gravity. The Engle-Pereira-Rovelli-Livine (EPRL) model [25, 26] is commonly considered the best current candidate to describe the quantum gravity regime [27]. Freidel and Krasnov independently introduced a similar model [28] (sometimes the model is referred to as the FK-EPRL model). Among its different properties, remarkable connections have been established between the semiclassical limit of such a model (that is, in the double limit of finer discretization and large areas) and discrete general relativity via Regge calculus [29, 30, 31, 32, 27, 33, 34]. Further extensions of the EPRL model include the cosmological constant [35, 36] and a different metric signature [37].

Although LQG is approximately thirty-five years old, numerical computations applied to the covariant version of the theory prior to five years ago are pretty rare. Pioneering and significant examples are [38, 39, 40, 41]. In recent years, considerable progress has been made in advancing our understanding of the theory, thanks partly to the development of high-performance computing (HPC) methods and the usage of supercomputing resources. Starting from the current formulation of the EPRL model, a significant step forward was performed in [42]. In [42], the EPRL vertex amplitude was first written in a form suitable for a numerical evaluation. The key was exploiting the Cartan decomposition of the Lorentz group [43]. This led to many pioneering results, such as the first calculation of the transition amplitudes between two spin networks with dipole graphs [44], the numerical evaluations of the vertex amplitude [45], the initial investigations of infrared divergences with a hybrid approach [46], and the analysis of the flatness problem in spinfoams [31]. On top of theoretical achievements, in [47], an open-source numerical library called *sl2cfoam*, written in C, was released to compute EPRL

[1] Diffeomorphisms are transformations that involve the smooth rearrangement of points and fields within a manifold (spacetime), preserving the underlying structure and physics of the system.

vertex amplitude numerically. Despite the promising initial results, a significant step forward soon appeared to be required. Such a step came with the new version of sl2cfoam, called *sl2cfoam-next* [48]. With this new tool available and the introduction of new techniques and algorithms in LQG, several recent achievements have been obtained during the last few years. Apart from the papers described in this thesis, a few other examples are [49, 50].

This Ph.D. thesis presents a comprehensive compilation of integrated articles highlighting my contributions to covariant LQG relying on scientific computing, primarily based on the decomposition introduced in [42]. Except for *Arbutus*, I have employed all the superclusters offered by Compute Canada for the research projects discussed in this thesis: *Beluga*, *Narval*, *Cedar*, *Graham*, and *Niagara*. Using several different programming languages, I frequently exploited more than one cluster for each project, experimenting with varying parallelization schemes. Each Chapter of this Ph.D. thesis corresponds to a distinct research paper, offering insights into the novel applications of HPC in tackling the computational complexities associated with LQG calculations. The Chapters are presented to form a coherent thesis rather than being in chronological order.

Chapter 1 is a pedagogical guide paper to spinfoam transition amplitude calculations. Although this was not the first project I started working on, it is the most logical choice to introduce the spinfoam approach to LQG and the perspectives of scientific computing into it. We show how to write a transition amplitude, from the definition of the 2-complex to its numerical implementation. We guide the reader using an explicit example of balancing mathematical rigor with a practical approach. We start with the case of a single EPRL vertex amplitude, moving to the computation of an extended spinfoam transition amplitude based on the Δ_4 triangulation. Such a triangulation was first considered in [51] in the context of Lorentzian effective spinfoam models.

Chapters 2, 3, and 4 fit the research direction dedicated to infrared divergences in spinfoams. Ensuring that the path integral and associated amplitudes are consistent, finite, and renormalizable is an ongoing area of research. Achieving this consistency is crucial for computing amplitudes with a finite probability and making meaningful predictions. LQG, being a background-independent and nonperturbative theory, avoids UV divergences by using a fundamentally discrete structure of spacetime at the Planck scale. The quantization process in LQG inherently regularizes the UV behavior, preventing the emergence of UV infinities. However, the unbounded summation over the bulk degrees of freedom appearing in the definition of the transition amplitudes can cause large-volume infrared (IR) divergences [2]. Several renormalization procedures have been proposed over the last few years, both in 2-complex refinement [52, 53], or in group field theory [54, 55, 56]. However, many important questions remain open. These three Chapters address some of such questions. The role of the Barbero-Immirzi parameter in the continuum limit is particularly interesting to our analysis. The Barbero-Immirzi parameter in the theory represents the ambiguity related to the choice of the representation of the classical Poisson brackets as commutators in the quantum theory. The Immirzi parameter affects the quantization of area in LQG. The area spectrum, which represents the quantized area of a surface in Planck units, depends on the value of the Immirzi parameter. Different values of such a parameter correspond to various quantization schemes and, consequently, different

[2] However, IR divergences are absent in the theory extension that includes a cosmological constant [35].

area spectra. It can have different values, typically chosen from the real numbers, leading to slightly different geometry quantizations and other predictions.

The paper presented in Chapter 2 studies numerically the first-order radiative corrections to the self-energy in covariant loop quantum gravity. The self-energy is a spinfoam diagram believed to give the leading-order radiative correction to the theory. We analyze the scaling of the divergence with the infrared cutoff, for which previous analytical estimates provided widely different lower and upper bounds. It represents the first example of HPC numerical methods employed to compute a spinfoam amplitude containing an infrared bubble. Our paper's main result is that the self-energy divergence is approximately linear in the cutoff. We also investigated the role of the Barbero-Immirzi parameter in the asymptotic behavior, the dependence of the scaling on some boundary data, and the expectation values of boundary operators.

In Chapter 3, we numerically estimate the divergence of several two-vertex diagrams contributing to the radiative corrections for the Lorentzian EPRL spinfoam propagator. We compute the amplitudes as functions of a homogeneous cutoff over the bulk quantum numbers, fixed boundary data, and different Immirzi parameters. The main result of our analysis is that those with fewer than six internal faces converge for a specific class of two-vertex diagrams.

Chapter 4 is the final one dedicated to infrared divergences in spinfoams. It introduces a strategy to compute EPRL spinfoam amplitudes with many internal faces numerically using stochastic algorithms. We show that uniform sampling Monte Carlo is exceptionally effective in approximating the sum over internal quantum numbers of a spinfoam amplitude, considerably reducing the computational resources necessary. We apply it to compute large volume divergences of the theory. After using the self-energy as a stress test for validating the algorithm, We focus on the vertex renormalization (or 5-1 Pachner move) amplitude. Such a diagram is essential for studying the spinfoam infrared divergences along with the self-energy. We find surprising numerical evidence that the EPRL vertex renormalization amplitude is finite.

Chapter 5 moves to the application of numerical methods in spinfoam cosmology. We present a numerical analysis of a Hartle-Hawking state for the early universe in the deep quantum regime, using the spinfoam formalism. We study the behavior of the Lorentzian EPRL spinfoam amplitude with homogeneous boundary data under a graph refinement going from five to twenty boundary tetrahedra. This can be interpreted as a wave function of the universe, for which we compute boundary geometrical operators, correlation functions, and entanglement entropy. The numerical calculation is made possible by adapting the Metropolis-Hastings algorithm and computational methods appropriate for the deep quantum regime. We confirm that the transition amplitudes are stable against such refinement. We find that the average boundary geometry does not change, but the new degrees of freedom correct the quantum fluctuations of the boundary and the correlations between spatial patches. The expectation values are compatible with their geometrical interpretation, and the correlations between neighboring patches decay when computed across different spinfoam vertices.

Chapter 6 continues the line of research in spinfoam cosmology. Instead of the EPRL model, we consider a simplified topological case, where the BF theory describes the dynamics, and the 16-cell triangulation characterizes the spinfoam boundary. Using the Markov Chain Monte Carlo algorithm introduced initially in Chapter 5, we compute mean geometry, fluctua-

tions, and correlations. The results are consistent with the hypothesis that refining the triangulation does not substantially affect the global physical picture.

In Chapter 7, we apply numerical methods to investigate the hypothetical process of a black hole decaying via quantum tunneling. In particular, we outline an algorithm to compute the black-to-white hole transition amplitude using the loop quantum gravity covariant formulation and the Lorentzian Engle-Pereira-Rovelli-Livine model. We apply the algorithm to calculate the crossing time of the transition in the deep quantum regime, comparing our result with previous analytical estimates of the same physical observable in the semiclassical limit. Furthermore, we show how to evaluate the crossing time analytically using an alternative approach concerning the one currently in the literature. This method requires much easier calculations and emphasizes that the crossing time does not depend on the extrinsic geometry of the transition.

Before concluding the introduction of this Ph.D. thesis, a final remark is necessary. Apart from the formalism I focused on during my Ph.D., other LQG communities usually employ alternative numerical and analytical approaches to spinfoam. Each research direction sheds new light on various aspects of the covariant formalism, frequently proposing modifications of the mathematical models describing the theory. Among the main alternative routes, we mention the adaptation of the Lefschetz Thimble method to spinfoams to investigate the semiclassical limit [57, 32], the promising opening towards quantum computing [58, 59], the study of the cuboid renormalization [60], tensor network techniques applied to spinfoams [61, 62, 7], and the introduction of effective spinfoam models [63, 64, 65]. In particular, effect spinfoam models are extremely promising frameworks to simplify calculations and enable exploration of quantum gravity phenomena while maintaining essential features of LQG.

Chapter 1

How-To compute EPRL spinfoam amplitudes

Statement of co-authorship

This Chapter presents the paper [9], written by myself and Pietro Dona. It was published in *Universe* on 26 March 2022. My supervisor, Francesca Vidotto, forwarded me the chance to write this paper on the occasion of the special issue "Probing the Quantum Space-Time". I gratefully accepted it because I wanted to write a tutorial paper that could be a valuable introduction to the numerical spinfoam approach to LQG. I had to study and review all the literature on the subject one year before (when I started my Ph.D.). Although the covariant LQG field has become very active, the entry barrier was steep for me, mainly because of its unusual language and notions scattered around the literature. I noticed a cohesive and organized introduction with practical examples was missing, which would have been extremely useful for a student approaching the field. I asked Pietro Dona to write this paper with me since he is one of the most excellent field experts and one of its pioneer explorers. My primary role was to take care of the numerical computations and write the *jupyter* notebooks published in the paper. I also contributed to drafting the sections and creating diagrams and plots. I consider the estimate of the percentage of the work conducted solely by me between 40% and 60%. This version corresponds to the manuscript published in *Universe*, which is compatible with MDPI's copyright and licensing policy.

1.1 Introduction

Spinfoam theory provides a background-independent Lorentz covariant path integral for general relativity. Spinfoams provide dynamics to Loop Quantum Gravity, defining transition amplitudes between spin network states. A triangulation discretizes the space-time manifold, and its 2-complex regularizes the partition function.

The state-of-the-art spinfoam model is the EPRL-FK model [25, 28] (we will refer to it as just EPRL for brevity). There is a large consensus in the community [31, 66, 67, 32, 27] that the classical continuum theory can be recovered with a double limit of finer discretization and vanishing $\hbar$. This observation is supported by the emergence of Regge geometries and the

Regge action in the asymptotics of the 4-simplex vertex amplitude for large quantum numbers [68, 30] and the recent study of many vertices transition amplitudes.

The number of calculations possible within the models recently grew considerably. It was possible because of a paradigm shift in the field. It evolved from a theoretical framework to circumvent the difficulties in imposing the Hamiltonian constraint in the canonical approach [69] into a concrete tool where numerical calculations of transition amplitudes are possible [70, 47, 71, 44, 72, 60, 73].

With increased interest in the field, its entry barrier also increased vastly. Getting into spinfoam is difficult for a student or a researcher from a different field. There are plenty of reviews [74, 75] and excellent books [1] to study and learn the basic theory. On the other end of the spectrum, we have plenty of advanced frontline papers that explore the connection of spinfoam with GR [31, 57, 32] or possible phenomenological implications [76, 18, 77, 78].

We noticed a hole in the literature. There are no papers that give you all the tools needed to complete a spinfoam calculation, from its conception to the number. This paper guides the reader through calculating an EPRL transition amplitude pedagogically. We use an explicit example to help them not feel disoriented when dealing with abstract concepts. We hope this paper can fill that hole and open spinfoam calculation to a new generation of students and researchers.

To read this paper, advanced background knowledge on spinfoam is unnecessary. However, a basic understanding of the topic is helpful. We think of this work as a guide for making spinfoam calculations. We refer to targeted reviews of the EPRL model [74, 1, 75] for a comprehensive discussion of its definition, motivation, and physical significance.

We start with a brief review of the spinfoam theory's construction and the EPRL model's definition in Section 1.2. In the rest of the paper, we show the reader how to compute a spinfoam transition amplitude associated with a triangulation of the space-time manifold. We identify five necessary steps, each illustrated in a different section.

Step 1. Draw the 2-complex.
In Section 1.3, we describe how to build the 2-complex from the triangulation. It is crucial in regularizing the gravitational path integral and writing a finite transition amplitude.

Step 2. Write the EPRL spinfoam amplitude.
In Section 1.4, we give the prescription to write the transition amplitude associated with a 2-complex and introduce a very convenient graphical method to represent the amplitude. For the calculation, we resort to a divide-and-conquer strategy.

Step 3. Divide the EPRL transition amplitude into vertex contributions.
In Section 1.5, we show how to divide any transition amplitude into vertex amplitudes.

Step 4. Compute the EPRL vertex amplitudes.
In Section 1.6, we discuss the calculation of the vertex amplitude in terms of $SU(2)$ invariants and booster functions.

Step 5. Use `sl2cfoam-next` to compute a number.
We perform the numerical evaluation of the amplitude in Section 1.7 using the nu-

merical library `sl2cfoam-next` and discuss the necessary approximations. In this section, we also discuss and improve the extrapolation scheme discussed in [71] as a tentative to lift, at least part of, the approximation used to calculate the amplitude.

We complement our discussion with an explicit example. We compute the EPRL transition amplitude based on the triangulation Δ_4. It was considered first in [51] in Lorentzian the spinfoams. It is 2-complex that at the same time is not trivial (with more than one vertex), simple (with four vertices and some symmetry), but rich enough (with one bulk face) to require a certain degree of optimization to compute the associated amplitude. Moreover, in [51], coherent boundary data corresponding to a Lorentzian geometry was provided, allowing semiclassical calculation with some ease that we leave to future work.

1.2 Overview of the EPRL model

spinfoam theory is a promising approach to quantizing gravity. The goal is to define a path integral for general relativity in a non-perturbative and background-independent way. The spinfoam partition function assigns transition amplitudes between spin network states, a basis of the Loop Quantum Gravity kinematical Hilbert space. For this reason, spinfoam theory gives a dynamic to Loop Quantum Gravity, often referred to as Covariant Loop Quantum Gravity [1].

In the Plebanski formulation of general relativity [79], we formulate gravity as a topological BF theory with constraints [80]. The variables of the BF theory are a 2-form B conjugated to a connection ω (with curvature F)[1].

General relativity is not topological. The *simplictiy* constraints reduce the B-field in BF theory to a γ-simple 2-form $B = \star e \wedge e + \frac{1}{\gamma} e \wedge e$, reducing the action to the familiar Holst action [81].

The path integral of spinfoam theory is regularized on a triangulation, more precisely, its 2-complex, to truncate the degrees of freedom. We discretize and quantize the topological theory first. The B-fields are assigned to the faces of the 2-complex triangles and encode their geometry. The connection is regularized by considering only its holonomy g responsible for the parallel transport along the (half-)edges of the 2-complex (from one tetrahedron to another). The topological theory partition function consists of a collection of delta functions imposing flatness of each face of the 2-complex.

The partition function of the EPRL model enforces the simplicity constraints at the quantum level to reduce the topological theory to gravity. On a simplicial triangulation, we have a *linear* version of the simplicity constraints: we require the proportionality between the boost and rotation generators of $SL(2,\mathbb{C})$ $\vec{K} = \gamma \vec{L}$ at the boundary of any 4-simplex (vertex of the 2-complex). The generalization to arbitrary tessellation is possible [82] but requires complications beyond this work's scope. Therefore, we limit ourselves to 4-simplices.

The key ingredient of the EPRL model is the Y_γ map. It embeds the spin j $SU(2)$ representation into the lowest spin sector of the unitary irreducible representations in the principal series of $SL(2,\mathbb{C})$ labeled by $\rho, k = \gamma j, j$.

[1] This theory has no degrees of freedom: all the solutions of the equations of motion are gauge equivalent to the trivial one $d_\omega B = 0$ and $F(\omega) = 0$. The name derives from the name of the variables used and the simple form of the action $\int_{\mathcal{M}} B \wedge F(\omega)$.

We expand the BF theory partition function in terms of matrix elements of the holonomies in irreducible representations of $SL(2,\mathbb{C})$ $D^{\rho,k}_{jmln}(g)$. See Appendix A.2 and references therein for more details. The EPRL model prescription enforces the Y_γ map at every vertex of the 2-complex and restricts the irreducible representations to γ-simple ones $D^{\gamma j,j}_{jmjn}(g)$ [25].

If the 2-complex has a boundary, the spinfoam partition function maps states from the Loop quantum Gravity kinematical Hilbert space (identified with the boundary space of the spinfoam with the Y_γ map) into the complex numbers (quantum transition amplitudes between these states).

The EPRL spinfoam partition function is given as a state sum over $SU(2)$ spins j_f on the faces and intertwiners i_e on the edges of the 2-complex:

$$Z_\Delta = \sum_{j_f, i_e} \prod_f A_f(j_f) \prod_e A_e(i_e) \prod_v A_v\left(j_f,\ i_e\right)\ , \tag{1.1}$$

defined in terms of the face amplitude A_f, and the edge amplitude A_e and the vertex amplitude A_v. Requiring the correct convolution property of the path integral at a fixed boundary, the form of the face amplitude $A_f(j_f) = 2j_f + 1$ and the edge amplitude $A_e(i_e) = 2i_e + 1$ are fixed [83].

We will not give an explicit form of the amplitude for an arbitrary 2-complex. If the reader is interested, they can be found in many references [25, 74, 1]. Instead, we opt for a constructive approach. In Section 1.4, we guide the reader through the rules to write a general EPRL transition amplitude. In Section 1.5, we divide the transition amplitude in vertex amplitudes. In Section 1.6, we discuss the explicit form of the vertex amplitude and its form best suited for numerical calculations.

1.3 How-To draw the 2-complex

The spinfoam partition function is regularized on the 2-complex of a triangulation of the space-time manifold. Given a triangulation, we can build its 2-complex associating a *vertex* to each 4-simplex.

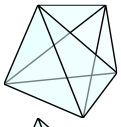
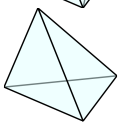
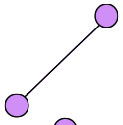
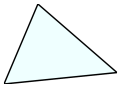
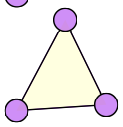
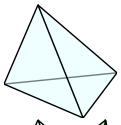
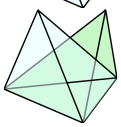
Each 4-simplex shares a tetrahedron with an adjacent 4-simplex. We associate to each tetrahedron of the triangulation an *edge* of the 2-complex. An edge connects two adjacent vertices. Each triangle in a 4-simplex is shared by two tetrahedra, which are generally shared with other 4-simplices. In the whole triangulation, a triangle can be shared by any number of tetrahedra and 4-simplices.

We associate a *face* of the 2-complex to each triangle of the triangulation. A face can contain any number of vertices and all the edges connecting them. We also assign an *orientation* to the faces of the 2-complex. This choice is needed for a well-defined notion of parallel transport (to identify the source and target of the holonomy uniquely).

Since two vertices share each edge for each of them, we can introduce two *half-edges*, one associated with each vertex. We can still picture them as dual to the tetrahedron but "seen" in the 4-simplex it belongs to. In each vertex in a given face, there are two half-edges. This is sometimes referred to as a *wedge*. The orientation of the face allows us to identify one half-edge as the source tetrahedron (reference frame) and the other as the target tetrahedron

(reference frame) of parallel transport along the face from the first to the last. If a boundary is present, the edges intersected by the boundary are severed in half, leaving only one half-edge in the skeleton.

We summarize the nomenclature introduced in this section in the following:

Triangulation		2-complex
 4-simplex	⇄	Vertex 
 Tetrahedron	⇄	Edge 
 Triangle	⇄	Face 
 Tetrahedron within 4-simplex	⇄	Half-edge 
 Oriented couple of tetrahedra in the same simplex	⇄	Wedge 

1.3.1 An example: the Δ_4 triangulation

Four 4-simplices form the triangulation, all sharing a triangle. The triangulation has seven points, nineteen segments, twenty-five triangles, sixteen tetrahedra (twelve in the boundary and four in the bulk), and four 4-simplices. We label the points with numbers from 1 to 7, segments with couples of different numbers (points), triangles with triples of distinct numbers (the shared triangle is 123), tetrahedra with a quadruple of distinct numbers, and 4-simplices with five distinct numbers. See Figure 1.1 for a pictorial representation of the triangulation.

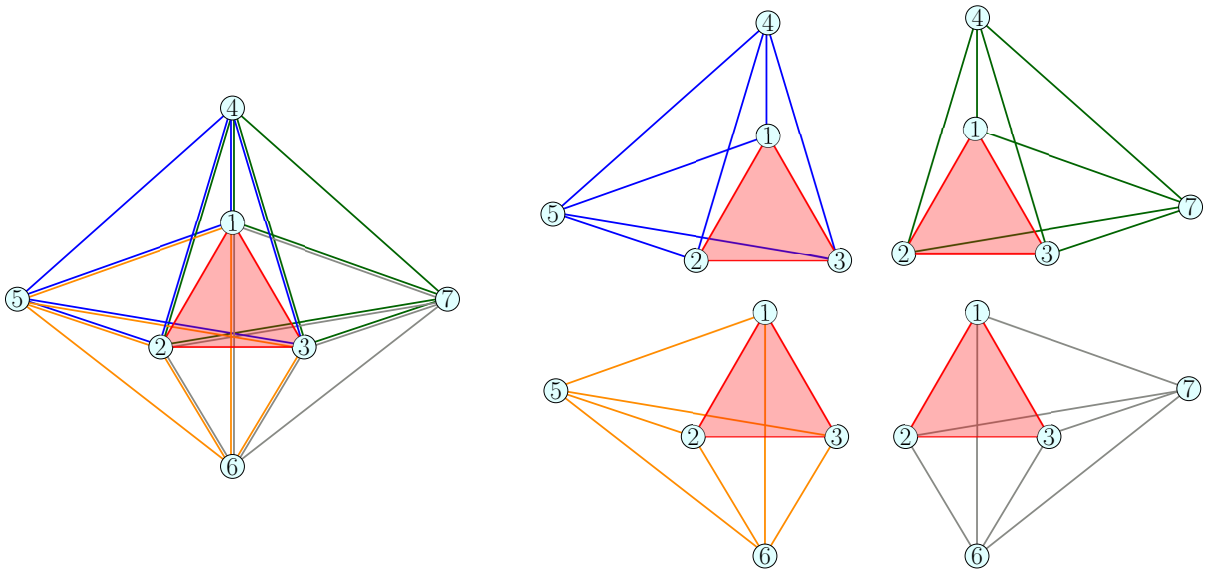


Figure 1.1: *The Δ_4 triangulation. The numbered circles correspond to points, while lines correspond to segments. Each color corresponds to a different 4-simplex. The bulk triangle* 123 *is highlighted in red. In the right panel, the 4-simplices are shown separately.*

The 2-complex of the Δ_4 triangulation has four vertices associated with a 4-simplex. It has four internal edges, each associated with a tetrahedron shared among two 4-simplices. There are also three external edges for each vertex. Each edge belongs to 4 faces, and each face is associated with a triangle of the Δ_4 triangulation. All triangles but one belong to the boundary of the triangulation. Therefore all the faces but one of the 2-complex are boundary faces. The bulk face is associated with the triangle shared by all 4-simplices. Thus it is crossing all four vertices. We label the 2-complex in the same way as the triangulation. See Figure 1.2 for a representation.

1.4 How-To write the EPRL spinfoam amplitude

For each wedge, we write a γ-simple unitary irreducible representation in the principal series of $SL(2,\mathbb{C})$ (see Appendix A.2 and reviews [84] and references therein for more mathematical details).

$$D^{\gamma j,j}_{jm,jn}(g_w)\,, \tag{1.2}$$

where $j \in \mathbb{N}/2$ is a spin, γ is the Immirzi parameter coming from the simplicity constraints, m,n are magnetic indices $m,n = -j,-j+1\ldots,j-1,j$, and $g_w \in SL(2,\mathbb{C})$ is a group element associated to the wedge. This restriction results from the weak quantum implementation of the simplicity constraints in the EPRL spinfoam model. The Y_γ map is responsible for this implementation and embeds the spin j $SU(2)$ representation in $SL(2,\mathbb{C})$ as in (1.2). The group element g_w represents the holonomy responsible for the parallel transport along the wedge from the source tetrahedron's reference system to the target tetrahedron's reference system. We conventionally associate the row of the representation matrix, the couple (j,m) in (1.2), to the target and the column, the couple (j,n) in (1.2), to the source. In this way, the $SL(2,\mathbb{C})$ γ-simple representation matrices inherit the orientation of the 2-complex.

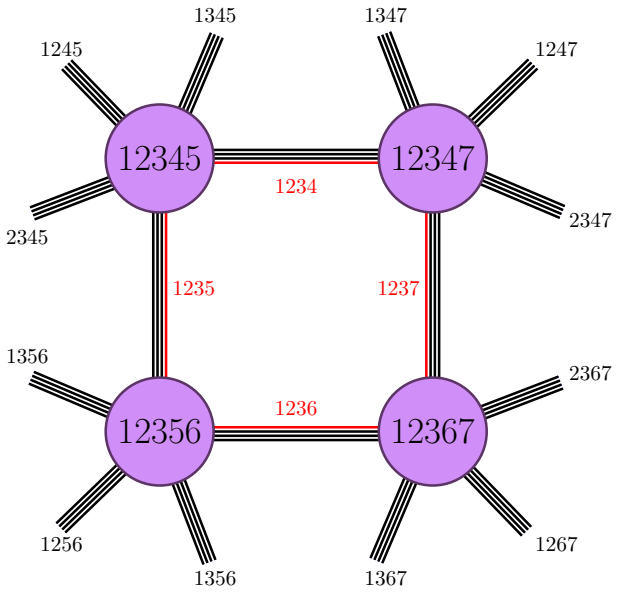


Figure 1.2: *The 2-complex of the Δ_4 triangulation. We named the vertices and the tetrahedra explicitly. We avoided naming the faces explicitly not to clutter the figure. Three numbers label the faces. We find a face's name by looking for the numbers in common to all the edges it belongs to. For example, the tetrahedra* 1234*,* 1235*,* 1236*, and* 1237 *all share the face* 123*.*

Instead of a group element for each wedge, we prefer a group element for each half-edge. We replace $g_w \to g_t^{-1} g_s$ where s and t are the source and target half-edges. This choice of fundamental variables guarantees that the parallel transport on a closed path in a vertex is trivial. In other words, the product of all the holonomies on the same closed path is the identity [2], or the holonomy is flat within a single vertex.

We set the spin j on each edge to be the same and contract the magnetic indices m, n. At the end of this procedure, the only non-contracted magnetic indices are on boundary half-edges. We prescribe them as part of the boundary data. A common choice to describe boundary data is to contract these magnetic indices with intertwiners in the recoupling basis or with coherent intertwiners if we want to represent some semi-classical geometry.

We sum over all the possible spins j_f associated with each closed face, and we weigh the contribution of the face with the dimensional factor $(2j_f + 1)$ [83]

$$\sum_{j_f} (2j_f + 1) \sum_{m_w, n_w} \left(\prod_{w \subset f} D^{\gamma j_f, j_f}_{j_f m_w j_f n_w}(g_w) \right) , \tag{1.4}$$

where the product is on all the wedges belonging to the face. We assign the spin as part of the boundary data on non-closed faces.

[2] Explicitly, if w_1, w_2 and w_3 are three wedges of the same vertex we have

$$g_{w_3} g_{w_2} g_{w_1} = g_{e_1}^{-1} g_{e_3} g_{e_3}^{-1} g_{e_2} g_{e_2}^{-1} g_{e_1} = \mathbb{1} , \tag{1.3}$$

where we have assumed that the wedges are oriented such that the target of w_1 is the source of w_2 and so on. If the orientation of one of the wedges w is the opposite, we replace g_w with its inverse.

We integrate over the group element associated with each half-edge using the Haar measure of $SL(2,\mathbb{C})$. One integration is redundant for each vertex, and we remove it to regularize the amplitude as prescribed in [85].

1.4.1 Graphical notation

Writing all EPRL spinfoam amplitude constituents can quickly get out of hand. We rely on a graphical notation to help us be precise and clear. We introduce the various elements as we need them. We represent a unitary irreducible representation as an oriented line in the principal series of $SL(2,\mathbb{C})$. The row labels correspond to the start of the line, and the column labels to the end of the line. We indicate the argument group element in a box and decorate the line with the needed representation labels

$$D^{\rho,k}_{jmln}(g) \quad = \quad \overset{\rho,k}{\underset{l,n}{\rightarrow}} \boxed{g} \rightarrow_{j,m} \,. \tag{1.5}$$

By connecting the two lines, we contract two representations summing over all the magnetic indices (both j and m are magnetic numbers from the perspective of the infinite-dimensional irreducible representations of $SL(2,\mathbb{C})$). For example, in graphical notation, the $SL(2,\mathbb{C})$ representation property reads

$$D^{\rho,k}_{jmln}(g_2 g_1) = \sum_{\substack{i\geq k \\ |p|\leq i}} D^{\rho,k}_{jmip}(g_2) D^{\rho,k}_{ipln}(g_1) \quad = \quad \overset{\rho,k}{\underset{l,n}{\rightarrow}} \boxed{g_1 g_2} \rightarrow_{j,m} \quad = \quad \overset{\rho,k}{\underset{l,n}{\rightarrow}} \boxed{g_1} \rightarrow \boxed{g_2} \rightarrow_{j,m} \,. \tag{1.6}$$

We denote the implementation of the Y_γ map (1.2) with a thick blue line that cuts across the representation line:

$$D^{\gamma j,j}_{jmjn}(g) \quad = \quad n \rightarrow \overset{j}{\boxed{g}} \rightarrow m \,. \tag{1.7}$$

If we apply the Y_γ map (1.7) to the product $g_1 g_2$ and use the decomposition (1.6), in the graphical notation, we have one blue line at both ends:

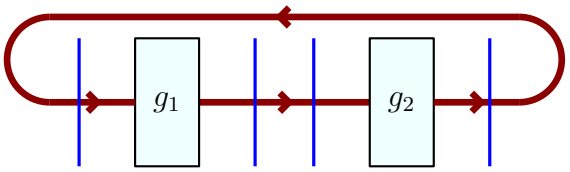
$$D^{\gamma j,j}_{jmjn}(g_1 g_2) \quad = \quad n \rightarrow \overset{j}{\boxed{g_1}} \rightarrow \boxed{g_2} \rightarrow m \,. \tag{1.8}$$

The (infinite) sum over two pairs of magnetic indices is implied in graphical notation, according to equation (1.6) . If we contract two representation lines with a Y_γ map, we only sum over one pair of magnetic indices:

$$\sum_{|p|\leq j} D^{\gamma j,j}_{jmjp}(g_2) D^{\gamma j,j}_{jpjn}(g_1) \quad = \quad n \rightarrow \boxed{g_1} \overset{j}{\rightarrow} \boxed{g_2} \rightarrow m \,. \tag{1.9}$$

We denote with a thicker red line the sum over the spin associated with that representation j weighted by a dimensional factor $(2j+1)$:

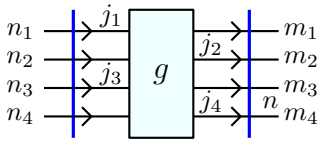
$$\sum_j (2j+1) \sum_{m,n} D^{\gamma j,j}_{jmjn}(g_1) D^{\gamma j,j}_{jnjm}(g_2) \quad = \quad$$



. (1.10)

The (tensor) product of two representations is represented as two lines. If the group element is the same, we use a single box. Similarly, for the Y-map, we use a single line. When we draw a box in amplitudes, we will always imply the integration with the Haar measure over the corresponding $SL(2,\mathbb{C})$ group element:

$$\int \mathrm{d}g D^{\gamma j_1,j_1}_{j_1 m_1 j_1 n_1}(g) D^{\gamma j_2,j_2}_{j_2 m_2 j_2 n_2}(g) D^{\gamma j_3,j_3}_{j_3 m_3 j_3 n_3}(g) D^{\gamma j_4,j_4}_{j_4 m_4 j_4 n_4}(g) \quad = \quad$$



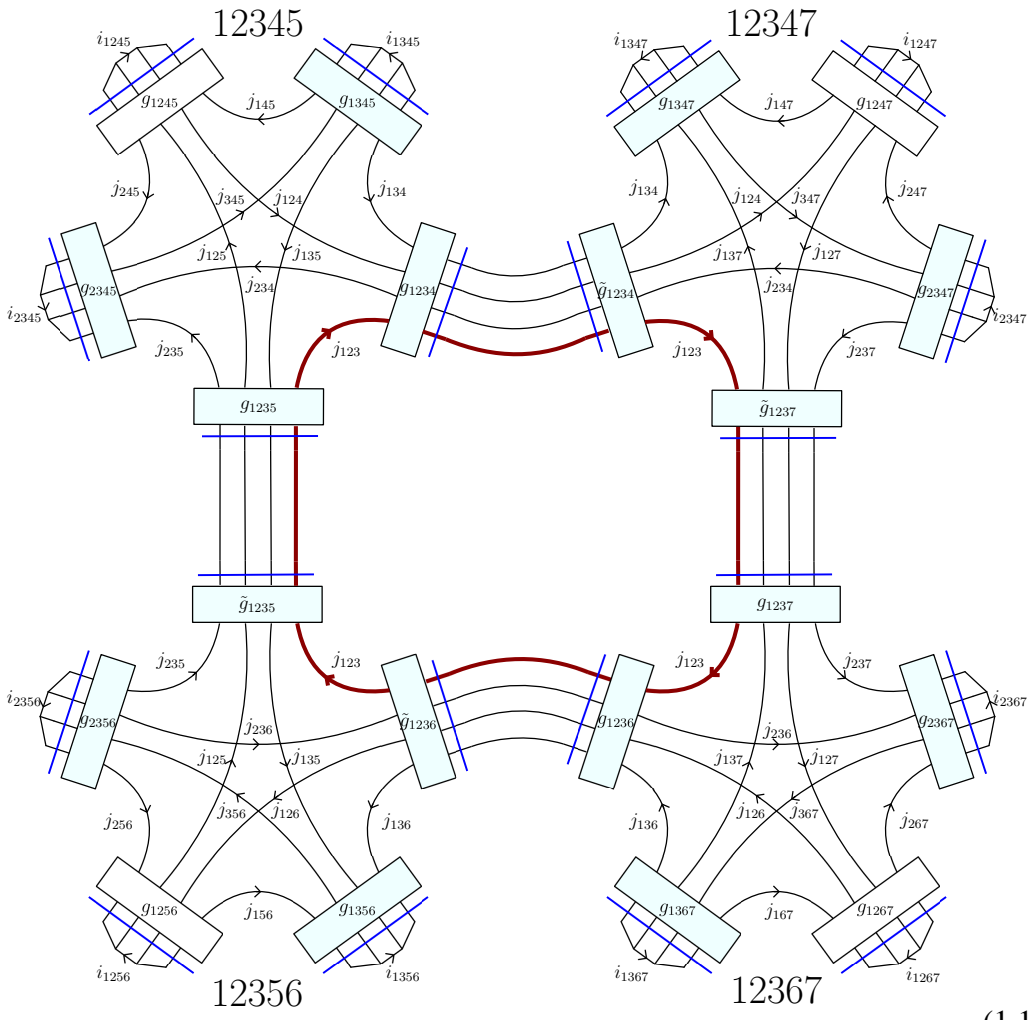
. (1.11)

1.4.2 An example: writing the Δ_4 amplitude

With the general recipe discussed in this Section and the corresponding graphical representation, we write the Δ_4 spinfoam amplitude associated with the 2-complex in Figure 1.2. We

also inherit the naming convention from the 2-complex.

A_{Δ_4} =



(1.12)

To assign a unique name to all the $SL(2,\mathbb{C})$ group elements, we denoted as g and $\tilde{g}$ the two group elements associated with the same (bulk) edge but belonging to different vertices. We used a small abuse of notation in writing (1.12). Some group elements appear as their inverse. To represent them as a single box, we opted not to distinguish them. However, following our conventions, the group element in the matrix element of a target half-edge always appears as its inverse. For example, the half-edge 1234 is the source of 234 and the target of 134. The group element g_{1234} appears as $D^{\gamma j_{234}, j_{234}}(g_{1234})$ and $D^{\gamma j_{134}, j_{134}}(g^{-1}_{1234})$.

As mentioned above , we contracted all the boundary magnetic indices with four valent intertwiners (12 in total) as part of the prescription of the boundary data. We chose the same recoupling basis on each and kept the label generic for the moment (i_e with e a quadruple identifying a boundary tetrahedron).

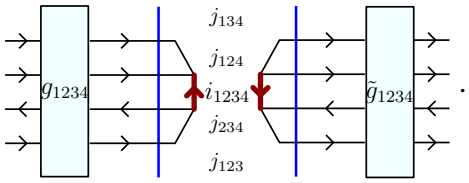
We highlighted in red the bulk face (123), dual to the triangle 123 in the Δ_4 triangulation 1.1. According to equation (1.10) , we are implying a summation over the spin j_{123} assigned to it weighted by a dimensional factor $2j_{123}+1$. As part of the boundary data, we also prescribed all the spins associated with the boundary faces. We keep them generic for the moment (j_f with f a triple identifying a boundary triangle).

We regularized the amplitude, removing one $SL(2,\mathbb{C})$ integration for each vertex as discussed above . In (1.12), we indicate the removed integrals with a white box. This choice is arbitrary, and the amplitude value is independent of this choice. However, we can use this arbitrariness to simplify the numerical computation (see Section 1.7) by making the symmetric choice. The integral removed is always opposite to the two bulk half edges and the bulk edge (123).

1.5 How-To divide the EPRL transition amplitudes

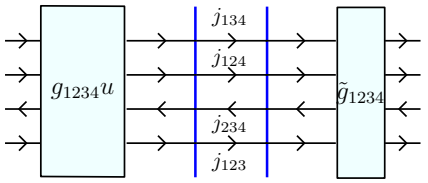
Approaching the calculation of the full amplitude is an arduous task. The group matrix elements in unitary representations are highly oscillating functions. The integrals are group integrals over many copies (sixteen in our example) of six-dimensional non-compact groups. We divide the transition amplitude into smaller, more manageable components and compute them serialized. This approach is the most advantageous if you aim to obtain a number from the computation of a transition amplitude. However, this could be suboptimal for semiclassical calculation due to the number of components.

Without any loss of generality, we insert a resolution of the identity over the intertwiner space between every two vertices of (1.12).



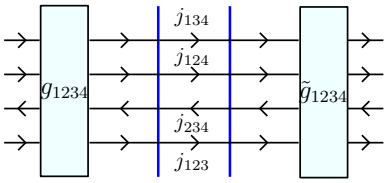
(1.13)

We rewrite the resolution of the identity over the intertwiner space as an integral over $SU(2)$ of four matrix elements (A.23). We commute the $SU(2)$ integral with the Y_γ map and bring the $SU(2)$ group element in the $SL(2,\mathbb{C})$ representation.



(1.14)

Finally, we use the invariance property of the $SL(2,\mathbb{C})$ Haar measure to reabsorb the $SU(2)$ group element with a change of variable, obtaining the original spinfoam edge.



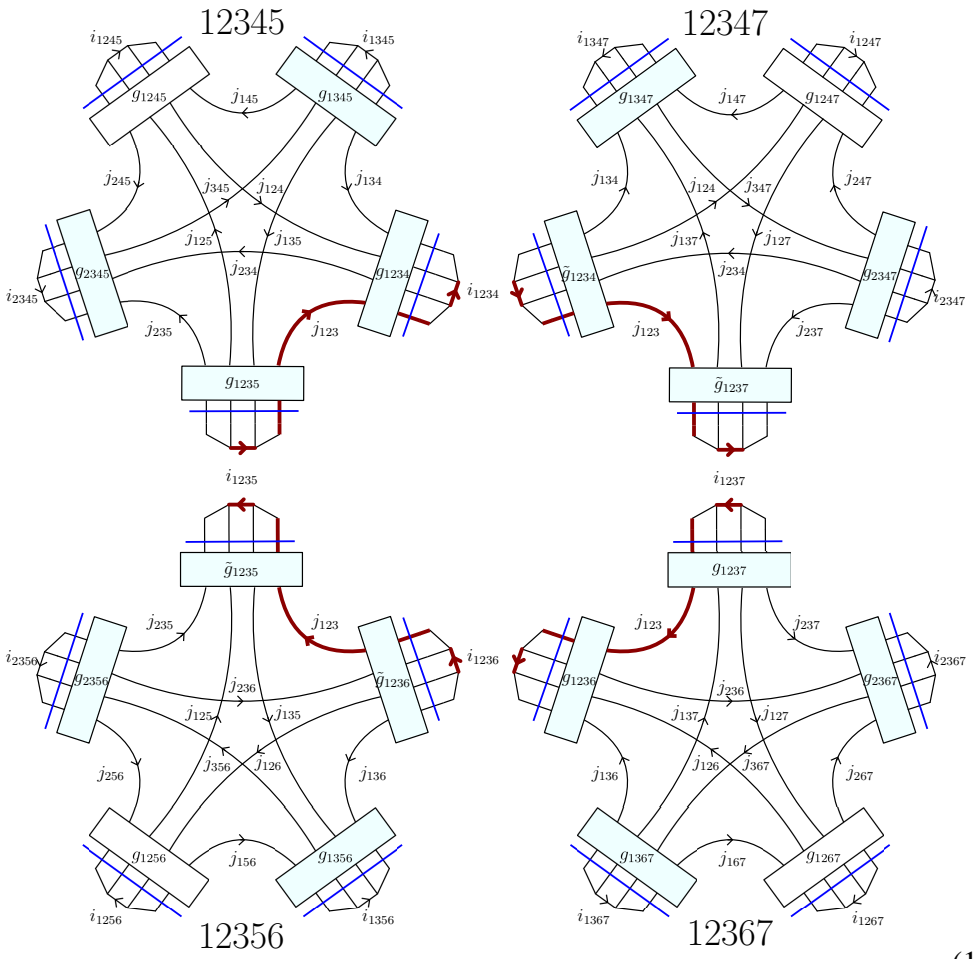
(1.15)

1.5.1 An example: decomposing the Δ_4 amplitude

If we divide the Δ_4 amplitude (1.12) inserting 4 resolutions of the identity (each one between two different vertices), the latter decomposes into a linear combination of the product

of four amplitudes. That is, one per vertex. These amplitudes are commonly known as *vertex amplitudes*. Using the graphical representation, we write the full Δ_4 transition amplitude as:

$$A_{\Delta_4} = \sum_{l_f}$$



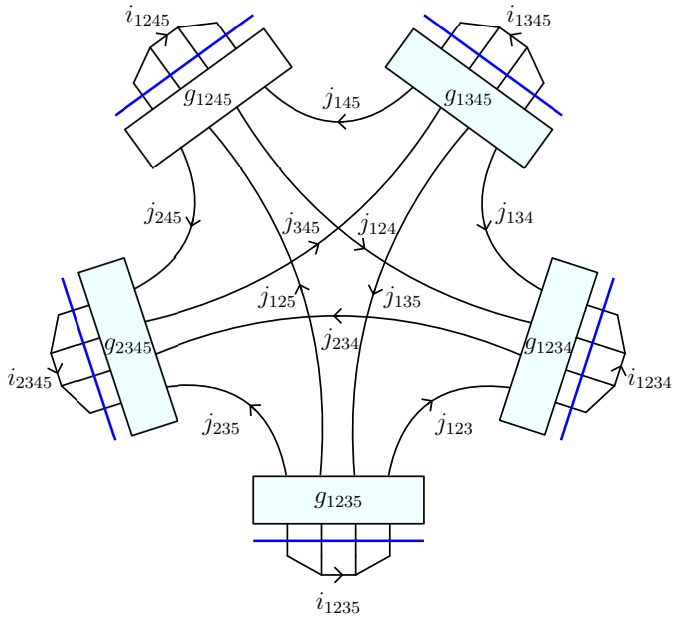
(1.16)

By separating the vertices as in (1.16), we have transformed the problem of calculating the full amplitude into the computation of the single building blocks: the vertex amplitudes.

1.6 How-To compute the EPRL vertex amplitudes

In this section, we will focus on contributions local at the vertices. For concreteness, we model the definition of the vertex amplitude on the (12345) vertex in the example (1.12).

$$A_{v_{12345}} = \quad . \tag{1.17}$$

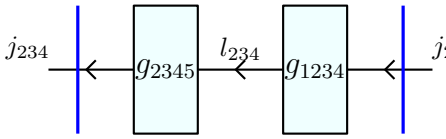


In (1.17) we contracted the magnetic indices of the bulk edges (1234) and (1235) with two intertwiners, labelled by i_{1234} and i_{1235}. We will see in the next section why this choice is natural, and we are not losing any generality. Remember that we regularized the amplitude by fixing the group element $g_{1245} = \mathbb{1}$, graphically denoting such element by leaving it blank.

Consider the contribution from the wedge (234). We use the representation property to separate the matrix elements corresponding to the two group elements.

$$D^{\gamma j_{234}, j_{234}}_{j_{234} m'_{234} j_{234} n_{234}}(g^{-1}_{2345} g_{1234}) = \sum_{|l_{234}| \geq j_{234}} \sum_{|n_{234}| \leq l_{234}} D^{\gamma j_{234}, j_{234}}_{j_{234} m'_{234} l_{234} n_{234}}(g^{-1}_{2345}) D^{\gamma j_{234}, j_{234}}_{l_{234} n_{234} j_{234} m_{234}}(g_{1234}) \tag{1.18}$$

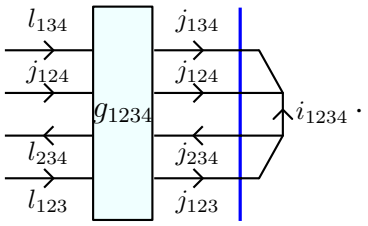
$$= \quad .$$



The inverse g_{2345} is due to the orientation of the wedge (234) and the conventions we are adopting.

To help the reader remember the extra summation introduced by the representation property, we wrote spin l_{234} even if we sum it over. This summation is bounded from below by j_{234} but is unbounded from above. It is a consequence of the non-compactness of the group (all unitary irreducible representations are infinite-dimensional). Each group element appears as

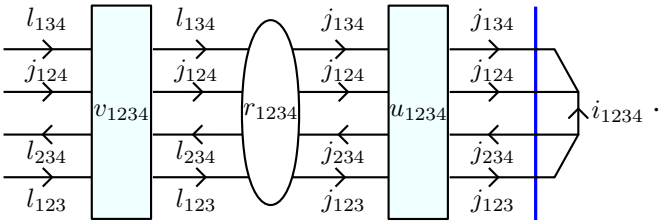
the argument of four matrix elements. For example, g_{1234} appears in the matrix elements



(1.19)

On the face (124) there is no sum over the spin l_{124}, as a consequence of the regularization choice $g_{1245} = \mathbb{1}$ and the presence of the Y_γ map on the half-edge (1245).

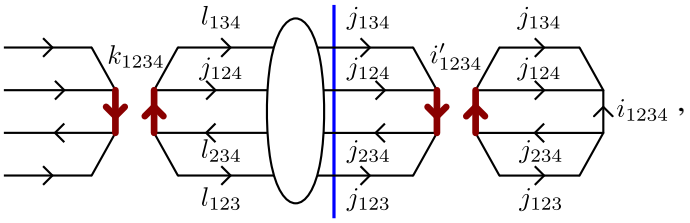
We parametrize each Lorentz transformation ($SL(2,\mathbb{C})$ group element) with an arbitrary rotation ($SU(2)$ group element) followed by a boost in a conventional direction (the 3 direction in our case) and another arbitrary rotation: the Cartan parametrization (A.35) of $SL(2,\mathbb{C})$. The representation matrices decompose as (A.35) and the Haar measure factorizes as in (A.36). We divide the contribution of the integral on the half-edge (1234) to the amplitude into



(1.20)

The matrix elements of u_{1234} and v_{1234} are $SU(2)$ matrix elements elements (A.38). We represent the integral over the rapidity r_{1234} of the product of four reduced matrix elements (A.40) as a white oval. We wrote the arguments explicitly to help the reader to visualize the parametrization. In the following, we will omit the name of redundant integration variables.

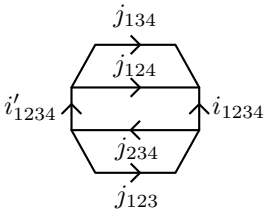
The Y map commutes with $SU(2)$ group elements. Therefore, we move it next to the rapidity integral. We perform the integrals over $SU(2)$ (A.25) in terms of $(4jm)$ symbols. The contribution to the amplitude from the half-edge (1.20) is



(1.21)

where the thicker red line represents a summation over the corresponding label weighted by a dimensional factor as in (1.10).

The contraction of two $(4jm)$ symbols obey the orthogonality condition (A.27) and allows us to remove the summation over i'_{1234}

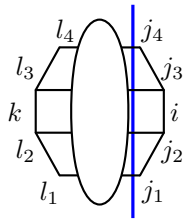


$$= (-1)^{2j_{234}} \frac{\delta_{i'_{1234} i_{1234}}}{2i_{1234}+1} \,. \tag{1.22}$$

where the phase $(-1)^{2j_{234}}$ is a consequence of the different orientation of the link (A.13).

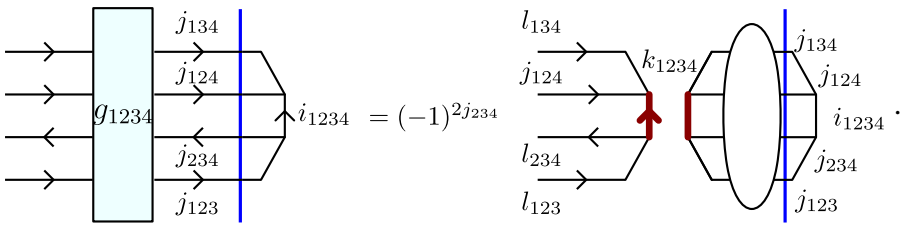
We define the booster functions B_4^γ as the result of the integral

$$B_4^\gamma(j_1,j_2,j_3,j_4,l_1,l_2,l_3,l_4;i,k) = \quad (1.23)$$



$$= \sum_{p_1,p_2,p_3,p_4} \begin{pmatrix} l_1 & l_2 & l_3 & l_4 \\ p_1 & p_2 & p_3 & p_4 \end{pmatrix}^{(k)} \left(\int_0^\infty \mathrm{d}r \frac{1}{4\pi} \sinh^2 r \bigotimes_{f=1}^{4} d^{\gamma j_f, j_f}_{l_f j_f p_f}(r) \right) \begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ p_1 & p_2 & p_3 & p_4 \end{pmatrix}^{(i)} .$$

The booster functions were first introduced in [42], numerically computed in [47, 48], analytically evaluated in terms of complex gamma functions [86, 87]. They have an interesting geometrical interpretation in terms of boosted tetrahedra [88]. The booster functions encode how the EPRL model imposes the quantum simplicity constraints and depend on the Immirzi parameter γ. Note that, in the definition (1.23), we dropped the information on the orientation of the faces. The orientation of the faces in the booster function is irrelevant. The effect of orientation change of the $(4jm)$ symbols cancels exactly the effect of orientation change of the reduced density matrices of $SL(2,\mathbb{C})$, as we discuss in Appendix A.2. Using this definition, we can write (1.19) in terms of the booster functions as:

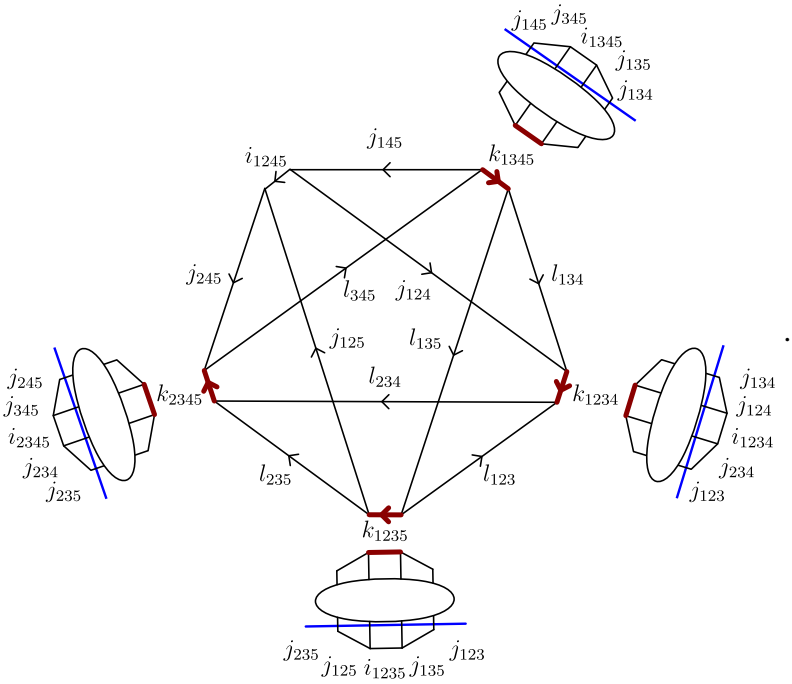


(1.24)

We compute the contribution to the amplitude from all other half edges of (12345) with the same prescription. The $(4jm)$ symbols in (1.24) contracts among themselves and form a $\{15j\}$

symbol of the first kind (A.30).

$$A_{v_{12345}} = (-1)^{2j_{135}+2j_{234}} \sum_{l_f}$$



(1.25)

The sum over spins l_f are only bounded from below (e.g. $l_{123} \geq j_{123}$) and the intertwiners k_e are bounded by the triangular inequalities of the $(4jm)$ symbols. To complete the calculation we recognize the $SU(2)$invariant as a canonical $\{15j\}$ symbol of the first kind (A.30). The vertex amplitude is

$$A_{v_{12345}} = (-1)^{2j_{135}+2j_{234}} \sum_{l_f} \begin{Bmatrix} i_{1245} & j_{124} & k_{1234} & l_{234} & k_{2345} \\ j_{145} & l_{134} & l_{123} & l_{235} & j_{245} \\ l_{345} & k_{1345} & l_{135} & k_{1235} & j_{125} \end{Bmatrix} \tag{1.26}$$
$$B_4^\gamma(j_{235}, j_{234}, j_{345}, j_{245}, l_{235}, l_{234}, l_{345}, j_{245}; i_{2345}, k_{2345})$$
$$B_4^\gamma(j_{123}, j_{135}, j_{125}, j_{235}, l_{123}, l_{135}, j_{125}, l_{235}; i_{1235}, k_{1235})$$
$$B_4^\gamma(j_{134}, j_{124}, j_{234}, j_{123}, l_{134}, j_{124}, l_{234}, l_{123}; i_{1234}, k_{1234})$$
$$B_4^\gamma(j_{145}, j_{345}, j_{135}, j_{134}, j_{145}, l_{345}, l_{135}, l_{134}; i_{1345}, k_{1345}) \ .$$

In general, one need to change the orientation of some links to obtain the canonical $\{15j\}$ symbol using (A.13) to compute the relative phase.

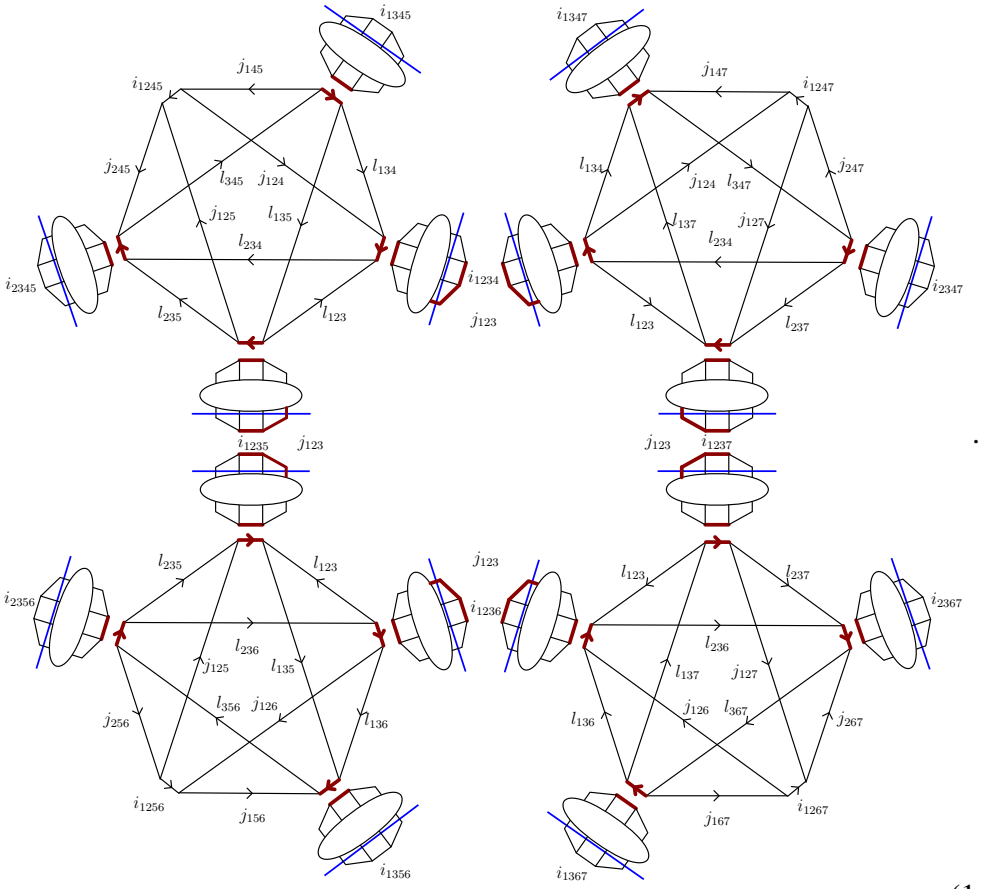
We rewrote the vertex amplitude (1.17) as a combination of a canonical $\{15j\}$ symbol weighted by four booster functions.

1.7 How-To calculate numbers

In the previous Section, we completed the formal evaluation of the amplitude. We can stop here if we are satisfied with the expression (1.16). A few more steps are needed to translate it

into a number. We decompose each vertex amplitude in (1.16) as in (1.25). By doing so, we finally write the Δ_4 transition amplitude in the appropriate form for a numerical evaluation:

$$A_{\Delta_4} = \sum_{l_f}$$



(1.27)

In this paper, we rely on the library `sl2cfoam-next` to evaluate the EPRL spinfoam amplitude numerically. The code discussed in this Section is available in the repository as notebooks [89].

1.7.1 Historical overview

The development of a library for the numerical computation of the Lorentzian EPRL 4-simplex vertex amplitude started with `sl2cfoam` [90]. The library is coded in `C` and is based on the decomposition of the vertex amplitude (1.26) in terms of booster functions. We refer to the original paper [47] for a detailed discussion of the library's performance, accuracy, and memory management.

The library computes the $SU(2)$ invariant symbols using `wigxjpf` [91]. The invariants are stored efficiently in custom hash tables based on `khash` [92] that consider their symmetry properties.

`sl2cfoam` computes the booster functions performing a numerical integration of the boost matrix elements (1.23). The integrand is rewritten as a finite sum of exponentials with complex coefficients to tame its highly oscillating behavior. One obtains the booster function from the interference of many exponential integrals with the trapezoidal rule. To reach enough

numerical precision, the authors employed arbitrary precision floating point computations with the GNU libraries `GMP` [93], `MPFR` [94] and `MPC` [95].

The library was used to explore the numerical properties of the EPRL vertex [31, 70, 46, 44]. The need for a much more efficient and accurate code immediately became clear, as the computational time for more complex amplitudes was out of reach.

Recently `sl2cfoam-next`[96], the evolution of `sl2cfoam`, has been released. Also, the new library is written in `C`, but it has an optional `julia` interface [97], which hugely simplifies its usage. Although `sl2cfoam-next` computes the Lorentzian EPRL vertex amplitude in the form (1.26), it introduces several ideas and techniques borrowed from High-Performance Computing and tensor networks. Therefore, with respect to the original version, it represents a significant improvement in performance and precision. We refer to [48] for its complete description and usage examples.

The numerical integration of the booster functions is performed with the Gauss-Kronrod quadrature method after a weighted sub-intervals decomposition of the integration range. Also, for technical reasons, the γ-simple unitary irreducible representations in the principal series of $SL(2,\mathbb{C})$ are slightly different from (1.2) as it uses $D^{\gamma(j+1)j}$ instead of $D^{\gamma jj}$.

The huge number of sums and products involved in the expression (1.17) is performed with optimized routines for multidimensional array multiplications (we will refer to them loosely as tensors in the informatics sense), such as BLAS [98] and MKL. For the description of the CPU parallelization scheme adopted, we refer to [48]. It has been recently introduced the possibility to offload tensor contractions to the GPU and parallelize them over the GPU cores with the `CUDA` platform [99] by using the `julia` package `CUDA.jl` [100, 101].

1.7.2 Introducing the cut-off

The vertex amplitude (1.26) comprises three distinct elements: the $\{15j\}$ symbol, the booster functions, and the combination of two. The formula (1.26) is exact, and `sl2cfoam-next` can compute its constituents to very high numerical precision. However, the sums over the spins l_f that appear in (1.26) due to the split of the representation matrix elements on the wedges are bounded from below but not from above. This means that to extract a number from (1.26), we must make an approximation and cut off the 6 unbounded sums in the vertex amplitude. While unbounded, the sums are convergent because the vertex amplitude is finite [85]. Therefore, in principle, it is possible to find a cut-off large enough to capture the value of the amplitude with the desired precision.

The library `sl2cfoam-next` implements a homogeneous cut-off Δl on all the unbounded summations. We replace the sums

$$\sum_{l_f=j_f}^{\infty} \quad \longrightarrow \quad \sum_{l_f=j_f}^{j_f+\Delta l} . \tag{1.28}$$

Unfortunately, we do not have a prescription to find the optimal Δl value. Numerical explorations show that it depends on the details of boundary data, such as the face spins j_f and the Barbero-Immirzi parameter γ. Currently, the best-consolidated strategy is to set Δl as large as possible and estimate the error by studying the value of the amplitude $A_{\Delta_4}(\Delta l)$ as a function of the cut-off.

Recently [71] introduced an extrapolation scheme to overcome the enormous computational cost of indefinitely increasing Δl. The extrapolation algorithm was used to calculate the self-energy spinfoam amplitude (see [102, 46]), which is a divergent amplitude since it contains a bubble. Furthermore, we mention that the implementation of Markov Chain Monte Carlo methods in the study of spinfoams based on the techniques discussed in this paper is in progress [78].

1.7.3 Using the `sl2cfoam-next`

Before any calculation, we need to import the `sl2cfoam-next` library and initialize it. In the blocks of code of this Section, we will imply that the library is correctly initialized first, and we omit the following code.

Listing 1.1: *Initialization of* `sl2cfoam-next`

```
1 using SL2Cfoam
2 Immirzi = 1.2
3 sl2c_data_folder = "$(path_to_library_data_folder)"
4 sl2c_configuration = SL2Cfoam.Config(VerbosityOff, VeryHighAccuracy, 100, 0)
5 SL2Cfoam.cinit(sl2c_data_folder, Immirzi, sl2c_configuration)
```

We set the value of the Immirzi parameter to 1.2 (for historical reasons, any value is equally possible). We define a data folder used both to look for the `fastwigxj` tables and to store the computed data optionally. In this way, we avoid recomputing the same vertex amplitude for a second time. We refer to the documentation of `sl2cfoam-next` and the accompanying paper [48] for a detailed description of all the setup options.

1.7.4 Computing one vertex

We find it very valuable to dedicate this paragraph to using the `julia` frontend of `sl2cfoam-next` to compute the EPRL vertex amplitude. We reference the amplitude (1.26). We provide some `jupyter` notebooks[3] in the Git repository [89], for interactive usage examples that the reader can compile and execute. In the Listing 1.2, we show an essential schematic representation of the code.

Listing 1.2: *Computation of a vertex amplitude with* `sl2cfoam-next`

```
1 Dl = 15
2 spins = j245, j125, j124, j145, j235, j234, j345, j123, j135, j134 = ones(10)
3 @time v = vertex_compute(spins, Dl);
```

We are omitting the initialization code in Listing 1.1. In lines 1-2, we specify the boundary data (all the spins equal to 1) and the cut-off $\Delta l = 15$. In line 3, we compute the amplitude. The function `vertex_compute` returns a tensor with five indices, one per intertwiner, computing the vertex amplitude (1.25) (without any phase) for all possible values of boundary intertwiners. In [89], we show how to compute a restricted range of boundary intertwiners.

[3] The code in [89] was tested with the kernel `julia 1.7.0`

The `@time` macro is used for logging purposes, tracking the computational time and memory usage. At fixed boundary spins and the Immirzi parameter, the computation time depends on several parameters, such as the value of the cut-off Δl and the accuracy level at which the library is set. With the parameters specified in Listing 1.1, the first time line 3 of Listing 1.2 is run takes 158 seconds. The computation time decreases exponentially by selecting a lower cut-off Δl. We tested this code on a laptop with Intel(R) Core(TM) i7-10750H 2.60GHz processor. The library distributes the workload on the available cores according to the parallelization scheme discussed in [48]. If we store the required booster functions during the first computation, the second time we run the script takes 3.2 seconds. It is the time to compute, sum, and contract all the required $\{15j\}$ symbols in the expression (1.25). Finally, If we store the full vertex amplitude, the computation time is negligible since nothing is calculated, and we retrieve the value from memory.

If we are interested in one single vertex amplitude, this is all we need to do.

1.7.5 An example: computing the Δ_4 amplitude with `sl2cfoam-next`

We split the computation of the amplitude (1.27) into two steps. First, we compute and save the value of all the necessary vertices. Then, we contract the required vertices to calculate the Δ_4 amplitude. We fix all boundary spins j equal to 1 for simplicity. The bulk spin j_{123} assumes values from 0 to $3j$, while bulk intertwiners i_{1234}, i_{1235}, i_{1236}, and i_{1237} assume values compatible with triangular inequalities. With the regularization choices we made, the vertex amplitudes are fully symmetric. The bulk spin and intertwiners always appear in the same position in each of the four vertices. Therefore, it is sufficient to compute only a single vertex amplitude for all the possible values of spins and intertwiners. To keep the computational time reasonable, we fix the cut-off Δl to 15. We analyze the dependence of the amplitude on this cut-off in the next step.

Listing 1.3: *Computation of all the vertex amplitudes needed in the computation of the transition amplitude* (1.27)

```
1  using JLD2
2  j = 1
3  Dl = 15
4  root_dir = pwd()
5  vertex_path = "$(root_dir)/vertex_ampls/Immirzi_$(Immirzi)/j_$(j)/Dl_$(Dl)"
6  mkpath(vertex_path)
7  j_bulk_min, j_bulk_max = 0, 3j
8  for j_bulk = j_bulk_min:j_bulk_max
9      spins = [j, j, j, j, j, j, j, j_bulk, j, j]
10     v = vertex_compute(spins, Dl)
11     vertex = v.a
12     @save "$(vertex_path)/j_bulk_$(j_bulk)_fulltensor.jld2" vertex
13 end
```

In lines 2-3, we set all boundary spins equal to 1, and the cut-off $\Delta l = 15$. In lines 4-6, we create the directory path to organize the files containing the computed amplitudes. In line 7, we define the range of the bulk spin, and from line 8, we loop over it. Finally, in lines 9-12, we assign the vertex amplitude's spins, compute the vertex amplitude, and save it for later use. Notice that we are computing the *fulltensor* vertex amplitude for all the possible values of boundary

intertwiners. This ensures that the Δ_4 amplitude can be calculated for any combination.

Finally, assembling all the vertices, we compute the whole amplitude (1.27). One of the main advantages of collecting the vertex amplitudes in multidimensional arrays is that there are very efficient methods to multiply (or "contract") the latter. For the application, we discuss in this work, improving upon a `for` loop is unnecessary, but `julia` offers the possibility to perform contractions wonderfully and, possibly, using the GPU. See, for example, the method `contract`, provided in `sl2cfoam-next` to contract vertices with coherent boundary states. Alternatively, there are packages such as `LoopVectorization.jl` (see [103] for an example) or libraries like `ITensor` [104]. In the following code block, we assume all the variables defined in Listing 1.3 are available.

Listing 1.4: *Computation of the transition amplitude* (1.27)*. All the vertex amplitudes are precomputed.*

```
1  i_b = 2
2  i = i_b + 1
3  D4_amp = 0.0
4  for j_bulk = j_bulk_min:j_bulk_max
5      fulltensor_to_load = "$(vertex_path)/j_bulk_$(j_bulk)_fulltensor.jld2"
6      @load "$(fulltensor_to_load)" vertex
7      D4_partial_amp = 0.0
8      D = size(vertex[i,:,:,i,i])[1]
9      for i_1234 in 1:D, i_1235 in 1:D, i_1236 in 1:D, i_1237 in 1:D
10         @inbounds D4_partial_amp += vertex[i,i_1234,i_1235,i,i]*vertex[i,i_1235,i_1236,i,i]*
11                                     vertex[i,i_1236,i_1237,i,i]*vertex[i,i_1237,i_1234,i,i]
12     end
13     D4_partial_amp *= (2j_bulk + 1)
14     D4_amp += D4_partial_amp
15 end
16 @show D4_amp
```

In lines 1-2, we define the boundary intertwiners. We pick them all equal to 2 for simplicity, but any other choice is also possible. Notice that in `julia`, the vector's index starts from 1. Therefore, we shift its value. In line 3, we initialize the variable that will contain the amplitude. From line 4, we loop over all the possible values of the bulk spin. In lines 5-6, we load the precomputed amplitude. In line 7, we initialize the variable to store the partial amplitude. The partial amplitude is the quantity in (1.27) at the bulk spin j_{123} fixed value. From lines 8 to 12, we sum the product of the four vertex amplitudes over the bulk intertwiners. Finally, in line 13, we add the dimensional factor to the full amplitude value we display in line 16.

1.7.6 Results and extrapolation

We summarize the result of our calculation in Table 1.1 and Figure 1.3.

Δl	0	1	2	3	4	5	6	7
$A_{\Delta_4}(\Delta l) \times 10^{36}$	0.202	1.09	2.03	2.59	2.90	3.09	3.21	3.29
Δl	8	9	10	11	12	13	14	15
$A_{\Delta_4}(\Delta l)$	3.36	3.40	3.44	3.47	3.50	3.51	3.53	3.54

Table 1.1: *Numerical values of the amplitude $A_{\Delta_4}(\Delta l)$ in function of the cut-off.*

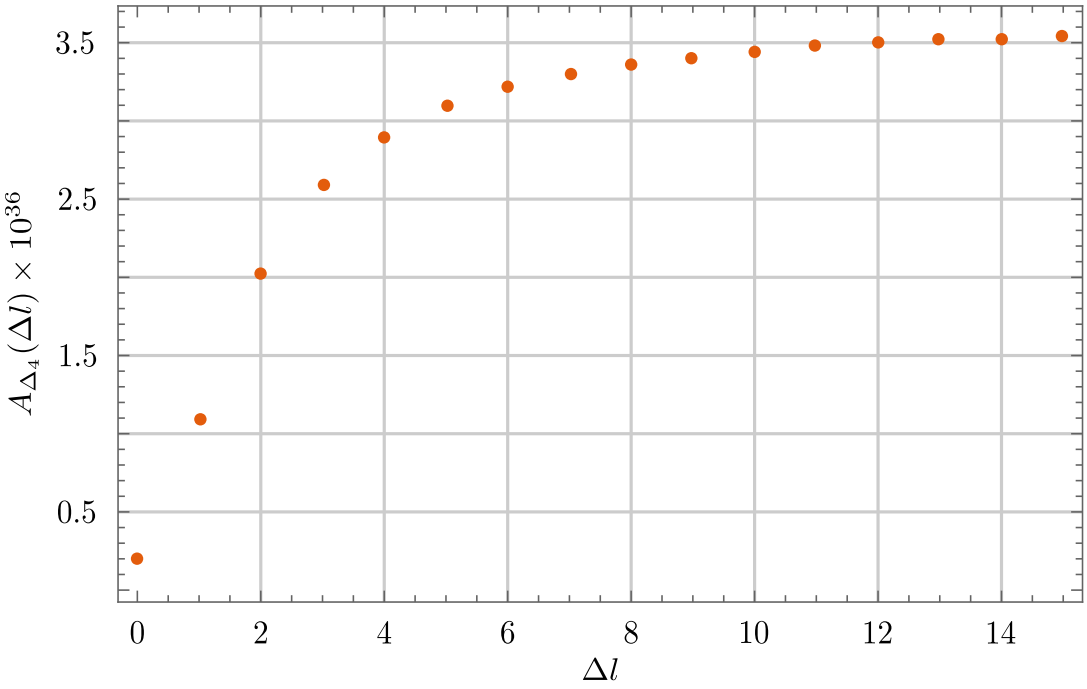


Figure 1.3: *Amplitude $A_{\Delta_4}(\Delta l)$ in function of the cut-off.*

Looking at the plot in Figure 1.3, it seems reasonable to deduce that by increasing the cut-off Δl, the value of the amplitude grows and (asymptotically) converges to the value of the amplitude. In the first numerical works based on `sl2cfoam` and `st2cfoam-next` [70, 31] the amplitude was approximated using the value with the largest available cut-off. However, we have more information (convergence, trends, speed). Is it possible to better estimate the amplitude with what we have?
We use series acceleration techniques. We reorganize the sums in the amplitude such that it takes the form

$$A_{\Delta_4}(\Delta l) = \sum_{n=0}^{\Delta l} a_n \,, \tag{1.29}$$

where a_0 is the amplitude with vanishing cut-off $\Delta l = 0$ (also called *simplified model* in [42]), a_1 encodes all the terms in the various sums of A_{Δ_4} that appears in the amplitude cut-off $\Delta l = 1$ but are not in a_0, and so on. The whole amplitude is recovered in the limit for an infinite cut-off of (1.29).

While recast in this form, we can apply techniques to estimate the value of numerical convergent series like the one in Appendix A.3. A similar approach was attempted in [72], and here we improve it and clarify it. The technique is analog to the Aitken delta-squared process [105] applied the the succession of the partial sum (1.29).

Since the amplitude is finite, the infinite cut-off limit of (1.29) exists, and the series defined in this way is convergent. We will assume that the ratios a_n/a_{n-1} are increasing (from a certain point onward). This assumption is backed up by numerical evidence (up to the available cut-

off). The lower bound estimate in (A.50) specialized for the series (1.29) is

$$A_{\Delta_4} \gtrapprox \frac{A_{\Delta_4}(\Delta l)A_{\Delta_4}(\Delta l-2)-A^2_{\Delta_4}(\Delta l-1)}{A_{\Delta_4}(\Delta l)-2A_{\Delta_4}(\Delta l-1)+A_{\Delta_4}(\Delta l-2)} = \frac{A_{\Delta_4}(15)A_{\Delta_4}(13)-A^2_{\Delta_4}(14)}{A_{\Delta_4}(15)-2A_{\Delta_4}(14)+A_{\Delta_4}(13)} \approx 3.61\cdot 10^{-36}\,, \tag{1.30}$$

where we specified the largest maximum value of the cut-off we computed, which is $\Delta l = 15$. The estimate (1.30) is significantly different from $A_{\Delta_4}(15)$ and does not require any additional calculation or resources. The lower bound (1.30) is analogous to the approximation we can obtain with Aitken's delta-squared process. With (A.50), we also obtain an upper bound to the amplitude.

$$A_{\Delta_4} \lessapprox \frac{A_{\Delta_4}(\Delta l)-A_{\Delta_4}(\Delta l-1)L}{1-L} = \frac{A_{\Delta_4}(15)-LA_{\Delta_4}(14)}{1-L} \approx 3.74\cdot 10^{-36}\,, \tag{1.31}$$

where $L = \lim_{\Delta l\to\infty} a_n/a_{n-1}$, which we estimate numerically with an inverse power law fit as in the example in Appendix A.3. We stress that the validity of this upper bound needs to be taken with a grain of salt since approximating the value of L can falsify the inequality in (1.31). Summarizing,

$$A_{\Delta_4} \in (3.61\cdot 10^{-36}, \approx 3.74\cdot 10^{-36})\,. \tag{1.32}$$

We can plot the amplitude together with the bound obtained from (1.30) and (1.31) to appreciate the improvement to the rough estimate $A_{\Delta_4}(15)$.

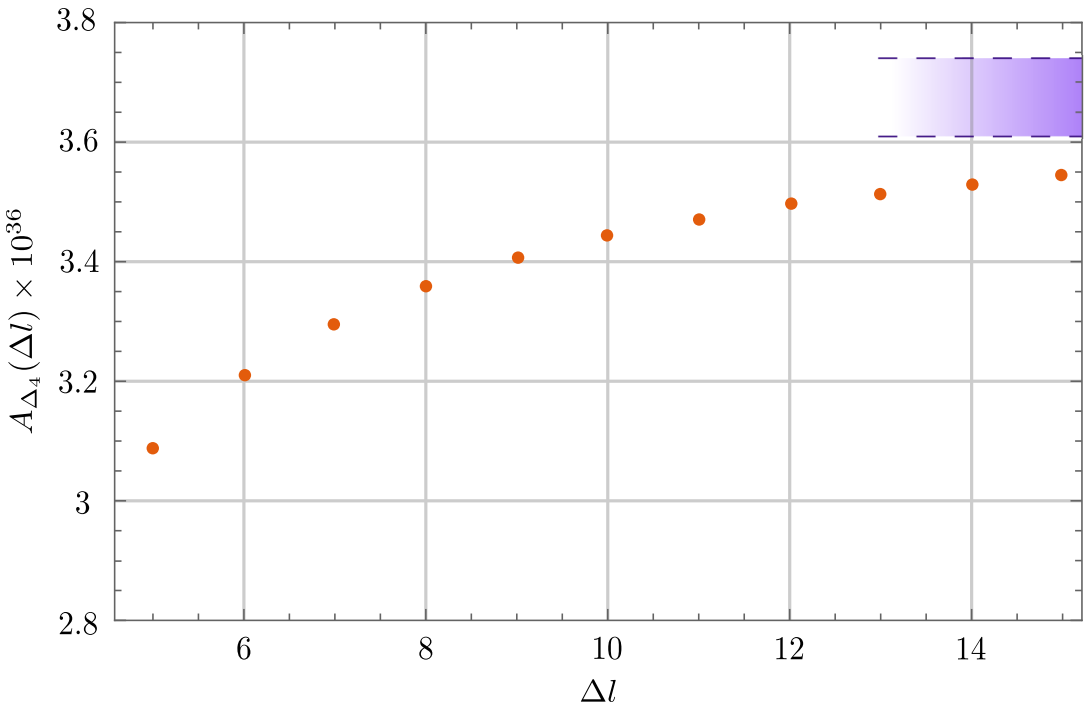


Figure 1.4: *Amplitude $A_{\Delta_4}(\Delta l)$ in function of the cut-off with the band* (1.32) *highlighted in blue. We excluded the points $\Delta l < 5$ for a better plot scale.*

Computational resources are precious. Up to this point, all our proposed calculations can be done on a standard laptop. How much can we improve the estimate by increasing the cut-off

using High-Performance Computing? Compute Canada's Narval cluster increased the cut-off from 15 to 25. The computation was distributed on 80 tasks with 10 CPUs per task, requiring about 8 minutes. The script used can be found in [89]. These were the resources we could employ in this project. However, there is still a big room for easy improvement.

Δl	16	17	18	19	20	21	22	23	24	25
$A_{\Delta_4}(\Delta l) \times 10^{36}$	3.55	3.56	3.57	3.58	3.58	3.59	3.59	3.60	3.60	3.61

Table 1.2: *Numerical values of the amplitude $A_{\Delta_4}(\Delta l)$ in function of the cut-off.*

Repeating the estimate process, we find new upper and lower bounds.

$$A_{\Delta_4} \in (3.63 \cdot 10^{-36}, \approx 3.69 \cdot 10^{-36}) \,. \tag{1.33}$$

The lower bound is marginally improved, as expected by comparing the numerical values in Table 1.2 to the ones in Table 1.1. However, the improvement on the upper bound is significant. Having more points to extrapolate the limit of the ratios L is essential.

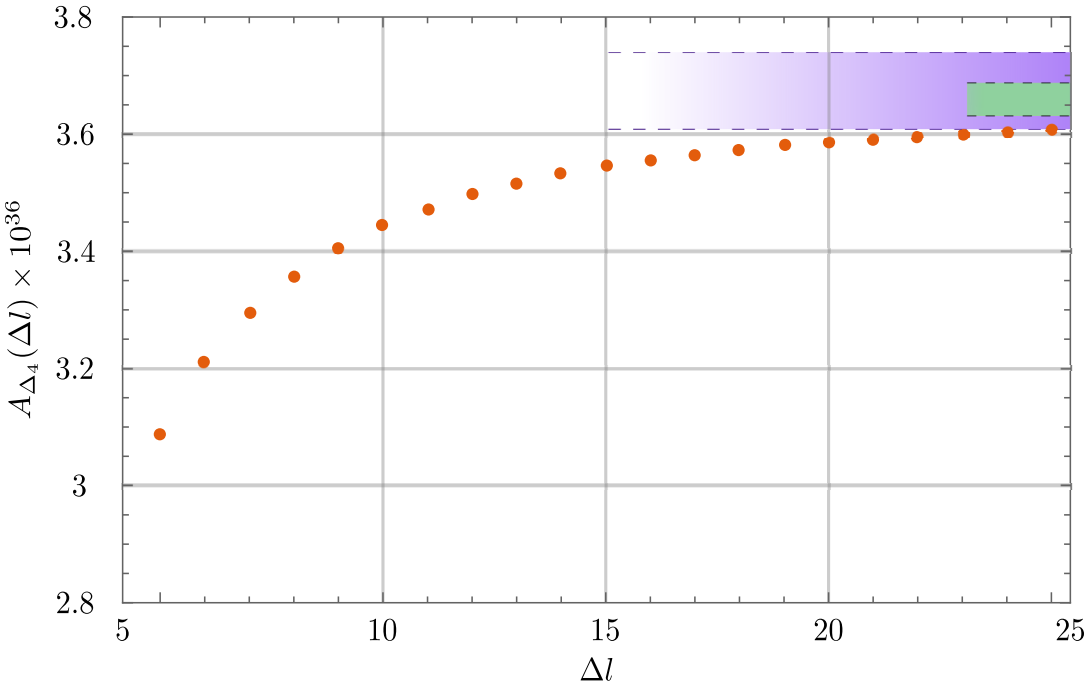


Figure 1.5: *Amplitude $A_{\Delta_4}(\Delta l)$ in function of the cut-off with the band* (1.32) *highlighted in purple and the band* (1.33) *highlighted in green.*

The calculation we proposed is not limited to our choice of boundary data. Using the same technique and adapting the code, we also compute the value of the A_{Δ_4} amplitude for boundary intertwiners $i_b = 1, 0$ and for other values of the Immirzi parameter $\gamma = 1, 0.1$. We summarize the results in Table 1.3

	$A_{\Delta_4}(25)$	$A_{\Delta_4}(25)$
$\gamma = 1.2, i_b = 1$	$5.06 \cdot 10^{-37}$	$(5.09 \cdot 10^{-37}, \quad 5.11 \cdot 10^{-37})$
$\gamma = 1.2, i_b = 0$	$1.90 \cdot 10^{-37}$	$(1.92 \cdot 10^{-37}, \quad 2.17 \cdot 10^{-37})$
$\gamma = 1.0, i_b = 2$	$9.44 \cdot 10^{-34}$	$(9.51 \cdot 10^{-34}, \quad 9.90 \cdot 10^{-34})$
$\gamma = 1.0, i_b = 1$	$1.49 \cdot 10^{-34}$	$(1.50 \cdot 10^{-34}, \quad 1.53 \cdot 10^{-34})$
$\gamma = 1.0, i_b = 0$	$4.95 \cdot 10^{-35}$	$(5.01 \cdot 10^{-35}, \quad 8.50 \cdot 10^{-35})$
$\gamma = 0.1, i_b = 2$	$4.28 \cdot 10^{-24}$	$(4.33 \cdot 10^{-24}, \quad 5.49 \cdot 10^{-24})$
$\gamma = 0.1, i_b = 1$	$1.24 \cdot 10^{-24}$	$(1.26 \cdot 10^{-24}, \quad 1.64 \cdot 10^{-24})$
$\gamma = 0.1, i_b = 0$	$2.33 \cdot 10^{-25}$	$(2.38 \cdot 10^{-25}, \quad 2.79 \cdot 10^{-25})$

Table 1.3: *Numerical calculation of the amplitude A_{Δ_4} with different boundary data. The boundary spins are all equal $j = 1$, and the cut-off $\Delta l = 25$.*

Chapter 2

Numerical analysis of the self-energy in covariant LQG

Statement of co-authorship

This Chapter discusses the paper [71], written by me, Francesco Gozzini, and Francesca Vidotto. It was published in *Physical Review D* on 20 May 2022. Furthermore, the proceeding paper [72] contains research material and results that converged into this paper. This was the first project I decided to invest and work on. The idea of using HPC to study the self-energy amplitude was born after I studied exhaustively the paper [46] and thanks to many exciting discussions with Francesco Gozzini, who had recently developed the `sl2cfoam-next` library. I took care of the numerical computations along with Francesco Gozzini. Furthermore, I contributed to drafting the sections and creating the cabling spinfoam diagrams and plots. It must be noted that the paper employs a peculiar extrapolation technique for convergent series, which is addressed as an "original numerical method". I found it accidentally by playing with the data, and it turned out to be crucial to state the conclusion of the work. Later on, an anonymous referee for a paper subsequent to this pointed out that this technique is equivalent to the Aitken extrapolation technique [105], which we kept utilizing on a standard basis ever since. I consider the estimate of the percentage of the total work conducted by me between 40% and 60%. This manuscript corresponds to the article's *Physical Review D*-published version, which agrees with the APS copyright policies.

2.1 Introduction

The spinfoam, or covariant, formulation of loop quantum gravity [74], is based on the definition of truncated amplitudes that define the background independent dynamics of quantum spacetime [106, 107] in a way similar to the Feynman graphs' definition of the dynamics of quantum field theory. In a Feynman graph, radiative corrections appear as loops (a set of propagators forming a circle) and may give rise to ultraviolet divergences. In a spinfoam, radiative corrections appear as bubbles (a set of faces forming a sphere). They may give rise to infrared

divergences when the cosmological constant is taken to be zero[1]. Here we investigate the first-order radiative correction to a single edge of the spinfoam, which is analog to the first-order contribution to the self-energy in quantum field theory. We use the version of the amplitude developed in [110, 111, 112, 28, 26, 113, 25, 82, 114] generally denoted as the EPRL amplitude. For reviews and notation, see [1, 115, 116].

The spinfoam infrared divergences have been studied analytically from several different perspectives [117, 52, 53, 118, 119, 120, 121, 122, 54, 55]. EPRL divergences have been studied in the Euclidean [123, 124] and Lorentzian theories [102, 46]. In particular, the last two references by Riello and Doná provide lower and upper bounds to the degree of divergence of the self-energy, resulting in a wide window. In [102], Riello gives a detailed analysis of the critical point that contributes to the amplitude in the large-spin scaling limit, based on the techniques introduced in [68], in the case of non-degenerate geometric configurations. He finds a logarithmic divergence in the cutoff K. He then estimates in a final appendix that the contribution of the degenerate sector might be linear in K, hence dominating the total divergence, postponing a detailed analysis of this sector to future studies. On the other hand, in [46], Doná derives an upper bound of the divergence by disregarding interference between various terms of the sum. His estimate leads to an upper bound on the divergence that scales as K^9. He also proves that the upper bounds derived with his algorithm provide excellent estimates of different kinds of infrared divergences restricted to the 3D model. This suggests that this might also be true in 4D, underlining the need for a complete numerical analysis for future studies.

These two results leave a window of possibilities that spans more than nine powers of the cutoff. In this work, we address the issue numerically. What we find is consistent with Riello's rough estimate: for small values of the Barbero-Immirzi parameter, the numerical analysis is compatible with a *linear* divergence in K.

The numerical calculation is possible thanks to the recently developed `sl2cfoam-next` library [48]. Computations were performed on Compute Canada's Cedar and Graham clusters (www.computecanada.ca) and on the Mesocentre of Aix-Marseille University (mesocentre.univ-amu.fr). The analysis and visualization of the computed data have been done with Julia [97] and Mathematica [125].

In section 2.2, we describe the structure of the self-energy amplitude. In section 2.3, we recall the calculation of the degree of divergence in the topological $SU(2)$ BF model, which can be found exactly. This allows us to compare the numerical results with the analytical ones in the literature. In section 2.4 we define the EPRL amplitude. In section 2.5, we describe the results of the numerical analysis of the EPRL self-energy amplitude. Finally, in the appendices, we report some further numerical studies and consistency checks.

2.2 Self-energy

Consider two four simplices joined by four tetrahedra. Two tetrahedra form the boundary of this cellular complex. In quantum gravity, this cellular complex can be associated with the

[1]The general definition of the spinfoam amplitudes includes a non-zero cosmological constant [108, 35, 109] that plays the role of an infrared cutoff and makes the amplitudes finite.

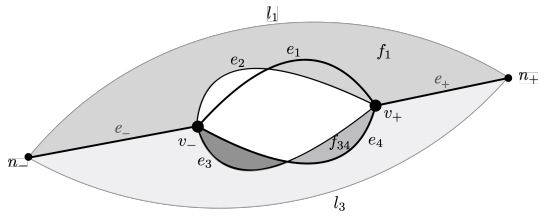


Figure 2.1: *The two-complex C of the self-energy. All vertices and edges, but only one internal and two external faces are depicted.*

transition from a single quantum of space to a single quantum of space via a splitting into four virtual quanta, which then recombine [106, 107].

The dual of this cellular complex is a two-complex C. It is formed by two pentavalent vertices $v_\pm$ joined by four internal edges e_a, with $a = 1, 2, 3, 4$. Each of the two vertices has one other edge $e_\pm$, ending on the boundary. The two-complex has four external faces f_a that end on the boundary, and six internal faces f_{ab}, dual to the surfaces that separate the four internal tetrahedra. See Figure 2.1. The six faces f_{ab}, taken together, form the bubble we are interested in.

The boundary of the two-complex C is the graph Γ formed by the two 4-valent nodes $n_\pm$ where the edges $e_\pm$ end (dual to the initial and final quanta of space), joined by four links l_a that bound the faces f_a (see Figure 2.2). The kinematical LQG boundary Hilbert space is therefore $\mathcal{H}_\Gamma = L_2\left[SU(2)^4/SU(2)^2\right]_\Gamma$. The graph Γ is represented in Figure 2.2.

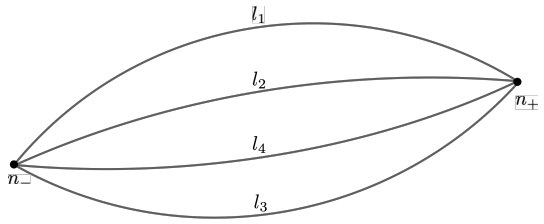


Figure 2.2: *The boundary graph.*

A basis in $\mathcal{H}_\Gamma$ is given by the spin network states $|j_a, i_\pm\rangle$, where j_a are the spins attached to the links l_a. In the two intertwiner spaces, we choose the recoupling basis $i_\pm$ defined by pairing the links j_1 and j_2 at the node $n_\pm$, i.e. the basis that diagonalizes the modulus square of the sum of the $SU(2)$ generators in the $SU(2)$ representation j_1 and j_2 [126, 127, 128]. The label i indicates the spin of the virtual link in this recoupling basis.

We focus on the subspaces $\mathcal{H}_j \subset \mathcal{H}_\Gamma$ where the four spins are equal $j_a = j$, and denote the basis states as

$$|j, i_\pm\rangle = |j, i_+\rangle \otimes |j, i_-\rangle \ . \tag{2.1}$$

We are interested in the EPRL amplitude [1]

$$W(j, i_\pm) = \langle W_C | j, i_\pm\rangle \tag{2.2}$$

relative to the two complex C. We need an infrared cutoff to compute this amplitude numerically to bound the sum over the spins associated with the internal faces. We denote this cutoff

K, and we define the cutoff amplitude as

$$W(j, i_\pm, K) = \langle W_{C,K} | j, i_\pm \rangle. \tag{2.3}$$

We are interested in studying the dependence of this amplitude on K.

2.3 BF amplitude

As a first step, we consider the amplitude

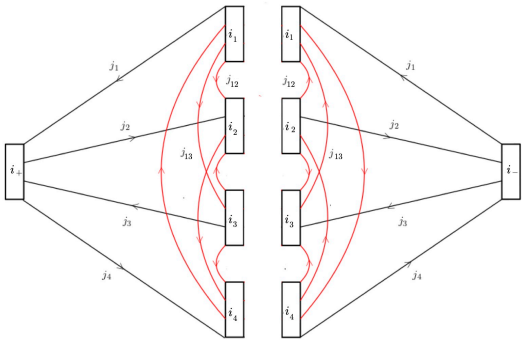
$$W_{BF}(j, i_\pm) = \langle W_C^{BF} | j, i_\pm \rangle \tag{2.4}$$

in the topological $SU(2)$ BF theory, where the amplitude can be computed analytically [46]. The BF self-energy amplitude can be written as

$$W_{BF}(j, i_\pm) = \sum_{j_{ab}, i_a} d_{\{j_{ab}\}} d_{\{i_a\}} \prod_{\pm} \{15j\}_{j_a, j_{ab}, i_\pm, i_a}, \tag{2.5}$$

where $d_{\{j_{ab}\}} = \prod_{(a,b)} (2j_{ab} + 1)$ and $d_{\{i_a\}} = \prod_a (2i_a + 1)$, with $a, b = 1, ...4$, $a \neq b$. The BF vertex amplitude $\{15j\}_{j_{AB}, i_A}$ is the 15-j Wigner symbol defined contracting the five intertwiners i_A with the ten spins J_{AB}, with $A, B = 1, ...5$, $A \neq B$. See Appendix A.1 for notation and conventions. A graphical representation of this amplitude that will be useful later on is the following:

$$W_{BF}(j, i_\pm) = \sum_{j_{ab}, i_a} d_{\{j_{ab}\}} d_{\{i_a\}} \times \tag{2.6}$$

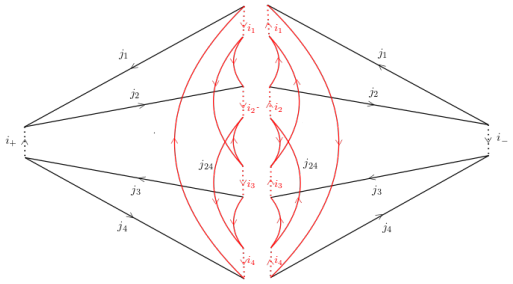


.

Here, for completeness, we should include in the expression for the amplitude the dimensional factor of the boundary faces and intertwiners, but this is a constant overall factor that can be ignored in the analysis since it does not affect the scaling of the divergence, that is what interests us. In (2.6), the six internal faces j_{ab} are highlighted in red. If we choose a recoupling

basis for the internal intertwiners as follows

$$W_{BF}(j, i_\pm) = \sum_{j_{ab}, i_a} d_{\{j_{ab}\}} d_{\{i_a\}} \times \tag{2.7}$$



,

then, this picture directly expresses the two BF vertex amplitudes in terms of Wigner 3-j symbols: each intersection of three lines is one such symbol, and the lines specify the contractions' scheme.

The multiple sums over the spins j_{ab} and the virtual spins i_a of the intertwiners are constrained by the Mandelstam identities that must be satisfied for the Wigner symbols not to vanish. But it is clear from the last picture that the fixed values of the external spins j_a do not prevent the internal spins j_{ab} from growing arbitrarily large. This is the origin of the divergence. To study it numerically, we introduce a cutoff $j_{max} = K$ on the six internal faces j_{ab}, bounding all the sums by K. That is, we define

$$W_{BF}(j, i_\pm; K) = \sum_{\substack{j_{ab} \le K \\ i_a}} d_{\{j_{ab}\}} d_{\{i_a\}} \prod_{\pm} \{15j\}_{j_a, j_{ab}, i_\pm, i_a}.$$

We have computed this quantity numerically for a fixed value of the j_a and $i_\pm$ for different values of K. In the BF case, it is known analytically that the degree of divergence does not depend on j_a and $i_\pm$. The result is shown in Figure 2.3, where the continuous curve is $W_{BF} = 4.8 \cdot K^9$. This result is consistent with the analytical result [46], which gives

$$W_{BF}(j, i_\pm; K) \propto K^9. \tag{2.8}$$

We briefly recall how to derive this result. Consider the simple case where all external spins are zero. The BF amplitude can be expressed in Fourier transform as multiple integrals over $SU(2)$, given by one integration per link and one delta function per face. In this case, we have

$$W_{BF} = \int \mathrm{d}g_a \prod_{a>b} \delta\left(g_a g_b^{-1}\right).$$

It is easy to see that one integration is redundant, and the other three can be used to eliminate three delta functions, leaving

$$W_{BF} = \delta(\mathbb{1})^3. \tag{2.9}$$

To regularize this expression, one can write the delta function as a sum over characters χ^j of irreducible representations of $SU(2)$ and introduce an infrared cutoff K, as follows:

$$\delta(U) = \lim_K \delta_K(U) = \lim_K \sum_j^K (2j+1)\chi^j(U) \ .$$

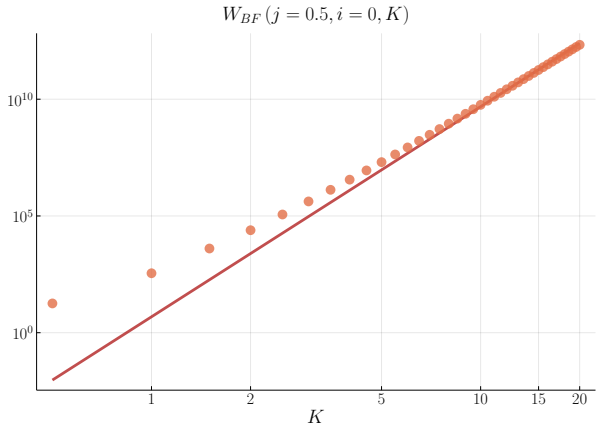


Figure 2.3: *Log-log plot of the BF self-energy amplitude as a function of the cutoff K on the internal spins j_{ab}. The continuous curve represents the $4.8 \cdot K^9$ function.*

Then, easily, $\delta_K(\mathbb{1}) \approx K^3$ and (2.8) follows from (2.9).

The numerical analysis in Figure 2.3 shows that the asymptotic scaling of the self-energy BF divergence (2.8) is reached for very low values of the infrared cutoff K. The computation involved in producing Figure 2.3 is nowadays trivial and can be easily performed on a laptop thanks to very efficient routines for computing Wigner (3,6,9)-j symbols [129, 130]. However, we note that the computational complexity grows for a large power of the cutoff index K, approximately equal to the number of internal faces of the foam. In Figure 2.4 we plot the total number of combinations of internal spins that must be computed for each cutoff value K till $K = 20$. We can see that the triangular inequalities reduce the complexity from K^6 to effectively $\sim K^{5.6}$. This still gives many configurations to consider and provides a considerable numerical challenge in the EPRL case, whose vertex amplitude is much more computationally complex than the BF one, as shown in the next section.

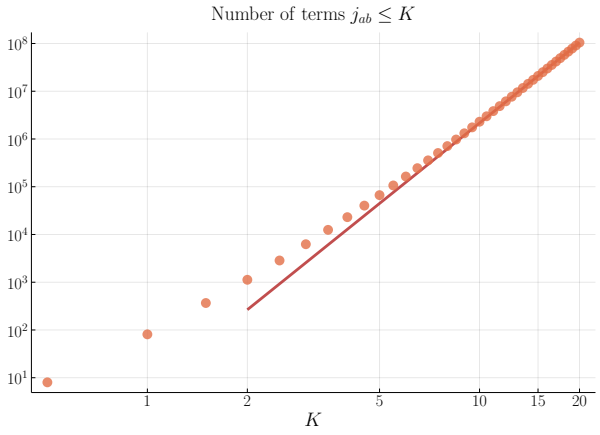


Figure 2.4: *Log-log plot of the total number of different configurations of internal spins j_{ab} to be summed over for increasing cutoff values K. The power-law fit is $\sim 1.7 \cdot K^{5.6}$.*

2.4 EPRL

Let us now come to the quantum gravity theory. The EPRL vertex amplitude is defined using a restriction of the unitary irreducible representations in the principal series of $SL(2,\mathbb{C})$ [25, 113]. To evaluate it numerically, a more tractable formulation has been derived in [42] and discussed in [47, 45]. In this formulation, the vertex is expressed as a sum over $SU(2)$ 15-j symbols weighted by one "booster function" $B4$ per edge (see Appendix A.2 for explicit formulas). The amplitude of a vertex bounded by spins j_f and intertwiners i_e can thus be written as a sum

$$V\left(j_f,\, i_e,\, \gamma\right) = \sum_{\substack{l_f \geq j_f \\ k_e}} d_{\{k_e\}} \left(\prod_e B_4(j_f, l_f; i_e, k_e; \gamma) \right) \{15j\}_{l_f, k_e} \tag{2.10}$$

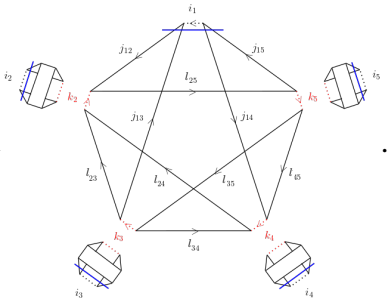
with virtual intertwiners k_e and virtual spins $l_f \geq j_f$, $e = 2...5$. One booster function must be replaced by the identity, corresponding to the suppression of a redundant $SL(2,C)$ integration[2]. Notice that we keep the dependence on the Barbero-Immirzi parameter explicit. In this expression, the infinite sum over l_f, called the "shell expansion", can be shown analytically to converge, but given the numerical calculation, it is convenient to truncate it at a finite value defining

$$V\left(j_f,\, i_e, \gamma\right) = \lim_{\Delta l \to \infty} V\left(j_f,\, i_e, \gamma; \Delta l\right) \tag{2.11}$$

where

$$V\left(j_f,\, i_e, \gamma; \Delta l\right) = \sum_{\substack{j_f \leq l_f \leq j_f + \Delta \\ k_e}} d_{\{k_e\}} \left(\prod_e B_4(j_f, l_f; i_e, k_e; \gamma) \right) \{15j\}_{l_f, k_e}$$

$$= \sum_{\substack{j_f \leq l_f \leq j_f + \Delta \\ k_e}} d_{\{k_e\}}$$



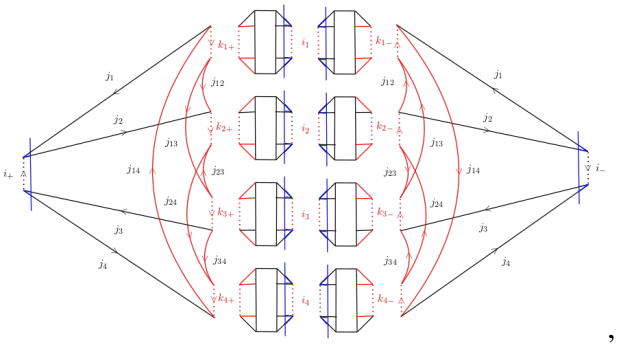
.

As before, we have neglected dimensional factors attached to the boundary intertwiners i_e and the boundary spins j_f. The following will refer to the cutoff Δl as the number of *shells*. Even if the convergence of the amplitudes a function of Δl is assured [47, 45], it is not possible to have a unique prescription to set the optimal Δl to get an acceptable convergence, since it depends on the details of data such as the face spins j_f and the Barbero-Immirzi parameter. Furthermore, the convergence strongly depends on the structure of the 2-complex, and there is no known general procedure to estimate the error made in truncating the sum over the auxiliary spins. In the next Section, we describe a technique to extrapolate the limit $\Delta l \to \infty$ for the self-energy.

[2]The 15-j symbol in (2.10) also depends on the four spins which label the gauge fixed edge, even if in the analytic notation it is not emphasized

With the form of the vertex described above, we can write the self-energy amplitude explicitly in a form suitable for numerical analysis. This gives

$$W(j, i_\pm, \gamma; K, \Delta l) = \sum_{\substack{j_{ab}\leq K \\ j_{ab}\leq l_{ab}\leq j_{ab}+\Delta l \\ i_a, k_{q\pm}}} d_{\{j_{ab}\}} d_{\{i_a\}} d_{\{k_{q\pm}\}} \times$$



, (2.12)

with $q = 1...4$. Compared to the diagrammatic expressions used for the EPRL vertex (2.12), we have rearranged some links to emphasize the structure of the contractions between the vertices, which defines the spinfoam associated with the triangulation described in Section 2.2.

The EPRL self-energy amplitude is the main object of the analysis performed in this paper. The interested reader can derive the complete analytical expressions by comparing the graphical counterparts with the definitions in Appendices A.1 and A.2.

When the two boundary intertwiners i_+ and i_- have the same value i, the EPRL self-energy amplitude simplifies as

$$W(j, i, \gamma; K, \Delta l) = \sum_{\substack{j_{ab}\leq K \\ i_a}} d_{\{j_{ab}\}}\, d_{\{i_a\}} V(j, j_{ab}; i, i_a; \gamma; \Delta l)^2 \tag{2.13}$$

since the two vertex amplitudes entering the sum over the internal faces j_{ab} are identical. Except for the coherent state and the boundary observable analysis (Appendix B.1), we shall always consider the two boundary intertwiners $i_\pm$ to have the same value. In the following, we use $i = (0, 0)$. The result with different values is substantially identical, differing only by a slight overall shift of the points.

Numerically, the vertex amplitude $V(j, j_{ab}; i, i_a; \gamma; \Delta l)$ in (2.13) can be efficiently computed using the recently developed library `sl2cfoam-next` [48] given the list of boundary spins ($j_i = j, j_{ab}$) and the number of shells Δl. The squared sum over the internal intertwiner indices i_a can be implemented by contracting the resulting *vertex tensor* with itself. The library automatically distributes all the configurations of internal spins across the available cluster nodes, and each node parallelizes the computation of the various shells over the available local CPUs. The particular code employed for this work is available at the repository [131].

From (2.12) it is clear that the computation of a single EPRL vertex amplitude is much more complex than the corresponding BF vertex amplitude with the same boundary spins and intertwiner indices. In fact, a single EPRL amplitude requires the computation of roughly $(\Delta l + 1)^6$ different BF amplitudes and $(\Delta l + 1)^3$ booster functions, *for each* different set of intertwiners i_a that bound the vertex (whose number is $\sim (2j + 1)^5$ if the boundary spins are of

order $\sim j$). In the present case, a different EPRL vertex tensor (i.e., a bundle of amplitudes at fixed boundary spins with running boundary intertwiner indices) must be computed for each configuration of the internal spins j_{ab}, whose number as a function of the cutoff K has been estimated in the previous Section (see Figure 2.4). From these simple estimates, we see that the numerical complexity of computing (2.13), using our methods, proliferates when any of the parameters K, Δl or j increases, roughly with a power law between 5 to 6 powers of the increasing parameter. We provide in the next Section some data for the resources employed in the computation of (2.13) up to $K = 10$ and $\Delta l = 20$.

2.5 Divergence Analysis

We have studied the amplitude (2.13) numerically and considered the following questions:

1. What is the asymptotic scaling of the EPRL divergence?

2. What is the dependence on the Barbero-Immirzi parameter?

3. At fixed boundary spins and Barbero-Immirzi parameter, does the scaling change using boundary spin network states or intrinsic coherent states?

4. What configurations contribute most to the divergence itself?

The first two questions are addressed in the main text, the third in the paper [71], and the fourth in Appendix B.2. The main difficulty in answering these questions is the limit $\Delta l \to \infty$ that cannot be taken numerically. To address this problem, we have relied on a property of the dependence of the amplitude on Δl at fixed K, that has emerged from the numerical analysis itself.

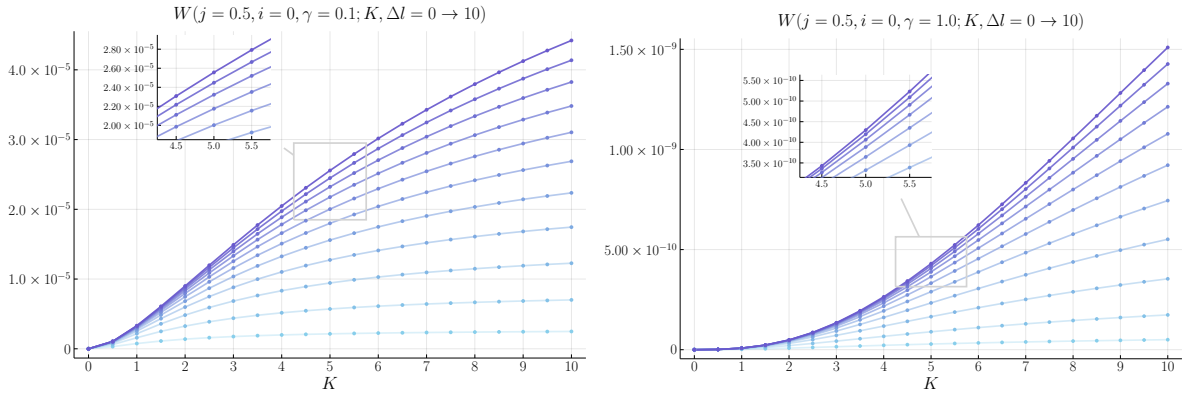


Figure 2.5: *The EPRL divergence* (2.13) *as a function of the cutoff parameter K, for some shells* $\Delta l = 0$ *to* 10. *Darker colors correspond to larger* Δl. *A portion of the plot around* $K = 5$ *is highlighted to show the convergence properties of the shell approximation.* Left: $\gamma = 0.1$. Right: $\gamma = 1.0$.

Extrapolation scheme

We show in Figure 2.5 the amplitude (2.13) as a function of the cutoff K, for various values of the number of shells Δl and $\gamma = 0.1$ and $\gamma = 1.0$. From a simple qualitative analysis, we infer that the convergence of the shell approximation is faster for a reduced bulk spins cutoff, while it becomes slower as K increases. The highest curve is a lower bound to the amplitude as a function of K, which is recovered in the limit $\Delta l \to \infty$. Convergence in the number of shells appears to depend on γ, with the case $\gamma = 1.0$ converging faster.

The convergence of (2.13) in the parameter Δl can be extrapolated from the data reported in Figure 2.5. To this aim, we have computed the ratios of the differences between adjacent curves of Figure 2.5, at fixed K, as a function of Δl. That is, we have studied the function:

$$f(\gamma, K, \Delta l) \equiv \frac{W(\gamma, K, \Delta l + 2) - W(\gamma, K, \Delta l + 1)}{W(\gamma, K, \Delta l + 1) - W(\gamma, K, \Delta l)} , \tag{2.14}$$

for each fixed value of j and i. The result is shown in Figure 2.6 for $K \geq 5$ and $\gamma = 1.0$. It appears from Figure 2.6 that the function $f(\gamma, K, \Delta l)$ tends to a constant for larger enough

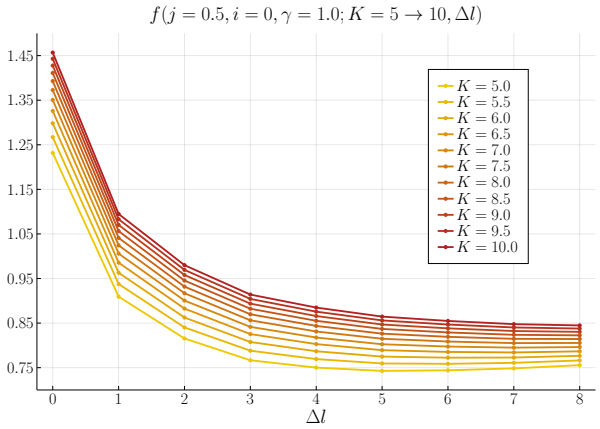


Figure 2.6: *Function* (2.14) *for* $\gamma = 1.0$. *The curves tend to a constant value* $c_{K,\gamma}$.

values of K, as a function of Δl (however, see also the more complete analysis for the case $\gamma = 0.1$, that is studied in a following section). Since (2.14) is the ratio between successive terms of a series, this means that the sum over the parameter Δl can be well approximated by a geometric series when K is large. Defining

$$c_{K,\gamma} \equiv f\,(\gamma, K, N - 2) \tag{2.15}$$

where N is the largest Δl parameter that has been reached numerically, our assumption implies for $\Delta l \geq N$

$$W(\gamma, K, \Delta l + 1) - W(\gamma, K, \Delta l) \approx \left(c_{K,\gamma}\right)^{\Delta l} . \tag{2.16}$$

We thus approximate the divergence in the limit $\Delta l \to \infty$ as

$$\begin{aligned} W(\gamma, K) &\equiv \lim_{\Delta l \to \infty} W(\gamma, K, \Delta l) \\ &\approx W(\gamma, K, N - 1) + \frac{W(\gamma, K, N) - W(\gamma, K, N - 1)}{1 - c_{K,\gamma}}, \end{aligned} \tag{2.17}$$

where we used the elementary limit of the geometric series. We note that the approximation improves when *(i)* more shells can be computed numerically, i.e., the index N increases, and *(ii)* the ratios (2.14) are closer to true constants in Δl. We applied this extrapolation scheme to the amplitudes computed numerically for different values of γ. In particular, in the case, $\gamma = 0.1$, where the convergence in Δl is the slowest, hence the approximation is the least accurate, we pushed the Δl parameter to very large values to study the accuracy of our extrapolation scheme and obtain very precise results for a physically relevant case.

Case $\gamma = 0.1$

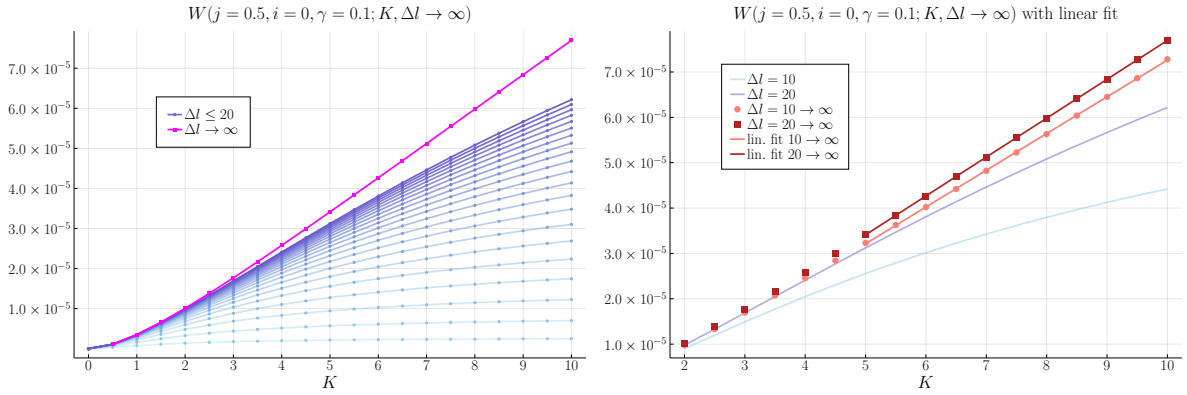


Figure 2.7: *The complete analysis for the case* $\gamma = 0.1$ *till* $N = \Delta l_{\max} = 20$. Left: *All the curves at various* Δl *with the extrapolation from* $\Delta l = 20$. Right: *The extrapolations from* $\Delta l = 10$ *and* $\Delta l = 20$ *with the corresponding linear fits. The slopes of the fits are* $8.1 \cdot 10^{-6}$ *and* $8.6 \cdot 10^{-6}$*, respectively. The standard errors of the fits, obtained with Julia's* `LsqFit` *package, are* $2.3 \cdot 10^{-8}$ *and* $4.6 \cdot 10^{-9}$*, respectively.*

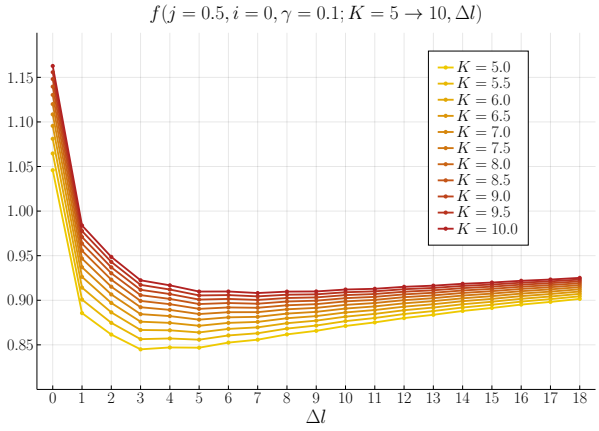


Figure 2.8: *Function* (2.14) *for* $\gamma = 0.1$ *plotted till* $N = \Delta l_{\max} = 20$. *It is evident that the approximation* $f \approx \text{const}$ *for* $\Delta l > N$ *provides a lower bound to the exact limit* $\Delta l \to \infty$.

In Figure 2.7, we show the amplitude (2.13) for the case $\gamma = 0.1$, with the computed data for $\Delta l \le 20$ and the extrapolated curves from $N = 10$ and from $N = 20$. From these plots, we can

verify that the approximation gives results that are already accurate, starting with low values of N since the difference between the two extrapolations is small, around 5% for $K = 10$. Plotting the ratios (2.14) till very large values of Δl, as we did in Figure 2.8, we can see that for the case $\gamma = 0.1$ the approximation $f(\gamma, K, \Delta l) \approx \text{const}$ is more accurate for larger values of K and less accurate for smaller cutoffs. This is not an issue since convergence in Δl is reached sooner for tiny cutoffs (since the ratios are smaller, the series converges faster). Also, Figure 2.8 shows that the extrapolation is a lower bound to the actual amplitude, although very close to the exact result.

Notably, the qualitative behavior of the divergence, either extrapolated from $N = 10$ or $N = 20$, is linear. In Figure 2.7, we show the fit of the extrapolated curves with straight lines. The extrapolated points fall precisely on a straight line for $K \gtrsim 5$ in both cases, with the better approximation at $N = 20$ being also closer to linear behavior than the one at $N = 10$. We conclude in this case that the overall divergence is very close to linear in the cutoff parameter K.

A few comments are in order. First, our results are in accordance with the rough analytical estimate of Riello [102]. It is essential to notice that his primary analysis focuses on what Riello calls the *non-degenerate* sector of the divergence and disregards the contributions from *degenerate* configurations when the 4-simplices dual to the two spinfoam vertices $v_\pm$ have non-maximal dimension. He found that this sector contributes with a leading factor of $\log K$ to the divergence. However, in Appendix C of [102], the remaining *degenerate* sector is estimated to contribute with a dominant factor of K to the overall divergence, which our numerical analysis shows. Our result also clearly indicates that the interference effects neglected in [46] strongly suppress the divergence of approximately eight powers of K.

Second, our numerical analysis does not entirely exclude the possibility that the behavior for very large values of K is different from what is inferred in the regime $K \lesssim 10$. There are however multiple arguments supporting our conclusion: *(i)* our findings are compatible with past analytical estimates [102]; *(ii)* from Figure 2.7 we see that improving the approximation also improves the matching with the linear fit, which appears perfect for as much as about 12 data points; *(iii)* we can compare the EPRL divergence with the BF case, where the asymptotic power-law divergence is already manifest for spins of order ~ 10 (see Figure 2.3), and analytically it is seen to be independent of the details of the chosen boundary data; *(iv)* recent numerical investigations [48, 132] strongly support the hypothesis that the semiclassical (i.e., asymptotic) regime of theory is reached for relatively small spins if the Barbero-Immirzi parameter γ is small (e.g., around 0.1). This appears to be related to the slower frequency of asymptotic amplitude oscillations whenever γ is small [132]. Hence, in this case, we can assume that we are looking at the asymptotic divergence for $K \gtrsim 5$. This is also supported by our analysis for larger values of γ, which do not appear to converge to a fixed (γ-independent) asymptotic behavior in the considered range of the cutoff, as we show in the next paragraph.

As a final comment for the $\gamma = 0.1$ case, we report technical details about the hardware resources employed. The computation of the divergence at $K \leq 10$ and $\Delta l = 20$ ran on 640 CPUs for ~ 60 hours, for a total of ~ 40000 CPU-hours. The computation at $\gamma = 0.1$ from $\Delta l = 0$ to 20 took about 125000 CPU-hours. Recall that the computational complexity of the EPRL model scales as $(\Delta l + 1)^6$. Thus the plots with the largest number of shells took most of the allocated CPU time. Symmetries in the internal spins j_{ab} peculiar to this case ($j = 1/2, i_\pm = 0$) have been implemented in the code to reduce the computational time of about

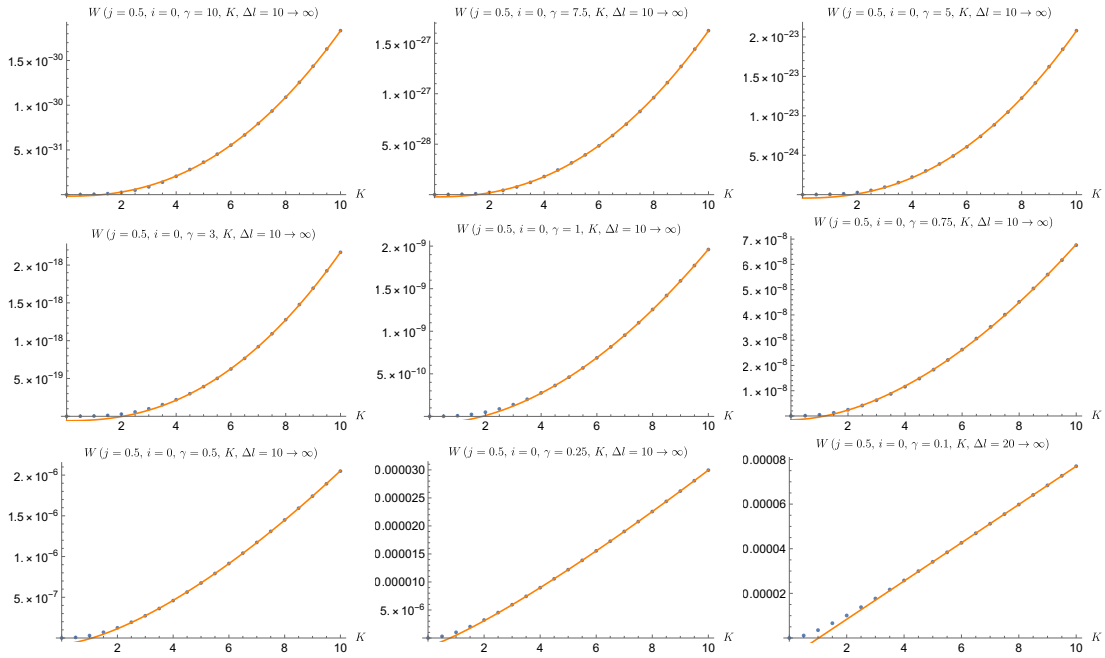


Figure 2.9: *Fit of the amplitude* $W(\gamma, K)$ *extrapolated via equation* (2.17) *with* $N = 10$ *for decreasing values of the Barbero-Immirzi parameter* $\gamma \neq 0.1$ *and with* $N = 20$ *for* $\gamma = 0.1$. *The fit is made by cutting the values of* $W(\gamma, K)$ *for* $K \leq 4$ *and superimposing the resulting curve with all the extrapolated points.*

a factor ~ 6.

Case $\gamma \gtrsim 0.1$

The extrapolation method outlined above allowed us to investigate also larger values of the Barbero-Immirzi parameter γ without replicating the already substantial computational effort required for the case $\gamma = 0.1$. We have studied the scaling of the asymptotic divergence for 9 values of γ, ranging between 0.1 and 10. The result is shown in Figure 2.9, for which we used $N = 10$ in equation (2.17). Notice that the extrapolation scheme is more effective in the cases $\gamma > 0.1$ than in the case $\gamma = 0.1$, as the curves of the ratios given by the function (2.14) become closer to constants as γ increases (compare Figure 2.6 with Figure 2.8). The data of Figure 2.9 are fitted with a function:

$$W(\gamma, K) = a + bK^c \tag{2.18}$$

where a, b, c are real coefficients. The best fit for the values of these parameters is shown in Table 2.1.

Table 2.1 shows that the amplitude is suppressed for large γ. The divergence is increasingly well fitted by an approximately linear scaling as the Barbero-Immirzi parameter decreases, while for $\gamma \gtrsim 1$, the curve is approximated by a quadratic function in the range $K \in [0, 10]$. We interpret this apparent dependence of the scaling on γ as a sign that the asymptotic or semi-classical regime is not reached for $K \leq 10$ when $\gamma \gtrsim 1$. A possible educated guess is that the

Table 2.1: *Fit coefficients table*

$W(\gamma, K) = a + bK^c \quad (j = 0.5, i = 0),\ K \in [0, 10]$			
γ	a	b	c
10	$-1.7 \cdot 10^{-32}$	$9.2 \cdot 10^{-33}$	2.3
7.5	$-2.2 \cdot 10^{-29}$	$8.2 \cdot 10^{-30}$	2.3
5	$-4.3 \cdot 10^{-25}$	$1.1 \cdot 10^{-25}$	2.3
3	$-5.7 \cdot 10^{-20}$	$1.1 \cdot 10^{-20}$	2.3
1	$-8.6 \cdot 10^{-11}$	$2.6 \cdot 10^{-11}$	1.9
0.75	$-1.7 \cdot 10^{-9}$	$1.1 \cdot 10^{-9}$	1.8
0.5	$-7.6 \cdot 10^{-8}$	$6.7 \cdot 10^{-8}$	1.5
0.25	$-1.5 \cdot 10^{-6}$	$2.0 \cdot 10^{-6}$	1.2
0.1	$-8.8 \cdot 10^{-6}$	$8.6 \cdot 10^{-6}$	1

non-linear behavior for large γ could correspond to the initial non-linear part of the curve at small γ, namely that the dependence could become linear for any γ at sufficiently high K. This intriguing hypothesis could have interesting consequences for a renormalization procedure. We leave the accurate testing of the large-γ sector for future works. As the computational complexity dramatically increases when K increases, it is likely that different numerical methods or approximations would be needed to test this sector, which is interesting from a theoretical point of view but does not appear attractive physically, given the current status of the theory.

2.6 Conclusions

In this work, we applied new computational techniques to the study of the infrared divergence of the spinfoam self-energy graph in the EPRL model. We have computed the divergent amplitudes with a running cutoff under an approximation (the "shells approximation"), which we then lifted by introducing an extrapolation scheme suited for the case considered. We tested our assumptions by refining the approximation up to many shells, using considerable computational resources for the most interesting case with the Barbero-Immirzi parameter $\gamma = 0.1$. We also explored the divergence at other values of γ and investigated the dependence and reliability of our results by using different boundary data and computing geometrical boundary observables.

Our findings considerably refine the upper and lower bounds in the previous literature [102, 46]. Furthermore, we have shown that the numerical evidence indicates a *linear* divergence in the infrared cutoff parameter. On the one hand, our result shows that the divergence scaling is much less severe than what is expected by approximate power-counting arguments of BF and Euclidean theories [123, 124] or upper-bound estimates to the Lorentzian theory [46]. However, on the other hand, our result also shows that the total divergence is of a higher order than the analytical estimate of the so-called "non-degenerate" sector [102] and thus provides the first strong evidence that virtual "degenerate" (i.e., of non-maximal dimension) configurations effectively dominate the first order quantum corrections to the bare spinfoam propagator. Previous works hinted at this point (see Appendix C of [102]) but were left unverified.

As in ordinary quantum field theories, understanding quantum corrections to the bare spin-

foam propagator is tightly related to the issue of renormalization, which is still an active area of research in the context of spinfoam and group field theories. Our work fills one of the first steps toward this program by precisely estimating the degree of divergence. Our results about the divergence at various values of γ also suggest that a possible renormalization scheme might involve running the Barbero-Immirzi parameter. The numerical testing of this hypothesis is left for future work, as the possible generalization of our extrapolation scheme to different divergent graphs.

This is one of the first works in the covariant loop quantum gravity field to exploit high-performance codes, specifically on a computer cluster. The computations accomplished in this paper would not have been possible using a single or a few machines. We believe that complex computational projects will prove to be more and more fundamental to advancing the field in the coming years, and we hope that our work and codes will provide a useful and encouraging ground for progressing in this direction. In this regard, the application of MCMC methods in covariant LQG to study the quantum regime in spinfoams composed of multiple vertex amplitudes glued together is described in [78].

Chapter 3

Radiative corrections to the Lorentzian EPRL spinfoam propagator

Statement of co-authorship

This Chapter illustrates the paper [133], written by me, Pietro Dona, and Edward-Wilson Ewing. It was published in *Physical Review D* on 26 September 2022. The idea of writing such a paper was born after an exciting discussion with Pietro Dona, who had previously given a presentation in which Ed Wilson was part of the audience. I took care of the numerical computations. Furthermore, I contributed to the analytical calculations, drafted sections, and realized plots and pictures. I estimate the percentage of the total work conducted by me between 30% and 50%. This manuscript corresponds to the article's *Physical Review D*-published version, which agrees with the APS copyright policies.

3.1 Introduction

The main goal of spinfoam theory is to define the dynamics of loop quantum gravity in a background-independent and Lorentz covariant way, providing transition amplitudes between spin network states [1, 74]. The state of the art is the EPRL and FK spinfoam models [25, 28]; this paper will focus on the Lorentzian EPRL spinfoam model. These theories have a compelling connection with discrete general relativity in the double limit of finer discretization and vanishing $\hbar$ [29, 30, 31, 32, 27].

The theory is ultraviolet finite; however, the unbounded summation over the bulk degrees of freedom can cause large-volume infrared divergences (although the infrared divergences are not present in the extension of the theory that includes a cosmological constant using quantum groups [35]). A complete spinfoam theory requires a renormalization procedure to remove these low-energy divergences, and renormalization will be essential to define the continuum limit properly. Various renormalization procedures have been proposed in the context of 2-complex refinement [52, 53], or in group field theory [54, 55, 56], but despite this work, there remain many critical open questions.

An essential step in developing a renormalization procedure in spinfoam models is to con-

sider 'self-energy' corrections to the propagator. These radiative corrections to the EPRL model have been studied analytically [102], numerically [71, 72], and using hybrid techniques [46]. So far, the main object of study has been a particular Feynman diagram believed to give the leading-order radiative correction; this Feynman diagram (which is often called the 'melonic' self-energy diagram) is a particular two-vertex spinfoam diagram associated with a space-time bubble. To calculate the contribution from this diagram, the common strategy is to introduce a homogeneous cutoff on all the spinfoam bulk face summations and estimate the degree of divergence of the amplitude, studying it as a function of the cutoff. There are strong numerical indications that the divergence for the melonic self-energy graph is linear in the cutoff [72, 71]. There are also some general (although weaker) bounds on the degree of divergence for any spinfoam diagram [134].

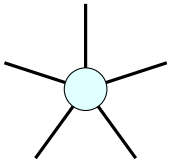
Different spinfoam amplitudes are generally associated with triangulations sharing the same fixed boundary. We order the spinfoam diagrams having the same boundary by the number of vertices. The melonic self-energy diagram is not the only two vertex diagram contributing to the radiative corrections to the Lorentzian EPRL propagator. Still, most of the other diagrams have fewer bulk faces. Therefore, we expect most of them to contribute to the self-energy calculation with subdominant divergences. This is indeed the case in topological $SU(2)$ BF theory, where the calculation can be done analytically (see Appendix C.1). Still, even if there are some clear structural analogies between the two theories (in particular, they share the same propagator), some significant differences remain. An explicit computation of the divergence of all diagrams is needed for the EPRL spinfoam model.

This paper continues the numerical study of the two-vertex diagrams contributing to the Lorentzian EPRL propagator started in [71]. We focus on the subclass of diagrams with two vertices and two boundary edges, each contained by the four boundary faces. The diagrams differ in the connectivity of the internal edges and faces; in particular, the number of internal faces varies from one diagram to another. Analytical calculations of transition amplitudes with the Lorentzian EPRL spinfoam theory are challenging and limited to the large spin regime. Recently, there has been significant progress in numerical techniques for spinfoam models, and we use these new developments for our numerical analysis. Specifically, we use the numerical techniques from [71], including the approximation scheme clarified in [9]. The numerical calculations are done using the library `sl2cfoam-next` [48] (available in the repository [96]), the state-of-the-art code to compute EPRL spinfoam amplitudes on a computer.

In Section 3.2, we review the Lorentzian EPRL transition amplitude and point out the origin of infrared divergences. We also list a class of diagrams contributing to the two-vertex correction to the propagator. Then, in Section 3.3, we overview the main ingredients of our numerical recipe. We conclude with Section 3.4 where we analyze the amplitudes and estimate the divergence of the spinfoam diagrams we study. For completeness, we also include a detailed expression of the Lorentzian EPRL vertex amplitude in the Appendices, the analytic calculation of the divergent two-vertex diagrams for the topological $SU(2)$ BF theory in Appendix C.1, and a discussion concerning other self-energy diagrams (that should be included or not in the calculation depending on the symmetries of the spinfoam model) in Appendix C.2.

3.2 EPRL radiative corrections

Spinfoam diagrams are constructed by contracting interaction vertices with spinfoam propagators along the edges. The Lorentzian EPRL spinfoam model prescribes a simplicial[1] interaction vertex and the associated vertex amplitude is

$A_v\left(j_f,\ i_e\right) =$  $= \sum_{l_f=j_f}^{\infty} \sum_{k_e} \{15j\}(j_f, l_f) \prod_{e=2}^{5} B_4^{\gamma}\left(l_f, j_f; i_e, k_e\right)(2k_e+1)\,,$ (3.1)

The amplitude (3.1) has fifteen arguments (ten spins j_f and five intertwiners i_e). The form of the amplitude in (3.1) is known as the booster decomposition of the vertex amplitude. It rewrites the vertex amplitude as a superposition of $15j$ symbols weighted by booster functions B_4^{γ} [42]. The booster functions enforce the simplicity constraints of the EPRL spinfoam models, depending on the Immirzi parameter γ, and have a compelling geometrical interpretation in terms of boosted tetrahedra [30]. We refer to the original paper [42] for an explicit expression in terms of intertwiners and $SL(2,\mathbb{C})$ γ-simple unitary irreducible representations. The EPRL propagator is simple and is given by

$A_e\,(i_e) =$ $= (2i_e+1)\,\delta_{i_e,i'_e}\,,$ (3.2)

forcing the intertwiners in two vertices corresponding to the same edge to be the same.

A spinfoam transition amplitude associated with the 2-complex of a triangulation Δ is given by the contraction of vertices and propagators dual to the 2-complex, spins and intertwiners associated with bulk faces and edges are summed over. Each face is weighted with a dimensional factor $A_f(j_f) = 2j_f + 1$ and the whole amplitude is

$$A_\Delta = \sum_{j_f, i_e} \prod_f A_f(j_f) \prod_e A_e(i_e) \prod_v A_v\left(j_f,\ i_e\right)\,. \tag{3.3}$$

We focus on diagrams contributing radiative corrections that satisfy three requirements to avoid an excessive proliferation of diagrams to study. Each diagram we consider has

1. Two vertices
2. Two boundary edges
3. Four boundary faces and each one of these faces contains both boundary edges.
4. Only trivial propagators, preserving the order of the faces.

[1] A general form of the spinfoam vertex exists for an arbitrary number of edges [82]. Still, we do not consider it here since, in this case, there is an infinite number of diagrams contributing to the radiative corrections to the theory. We restrict to the simplicial vertex to avoid this uncontrolled proliferation of diagrams.

We give some examples of diagrams excluded by these conditions in Appendix C.2. There are four classes of diagrams that differ by the combinatorics of the bulk edges shared by the two vertices, as shown in Figure 3.1.

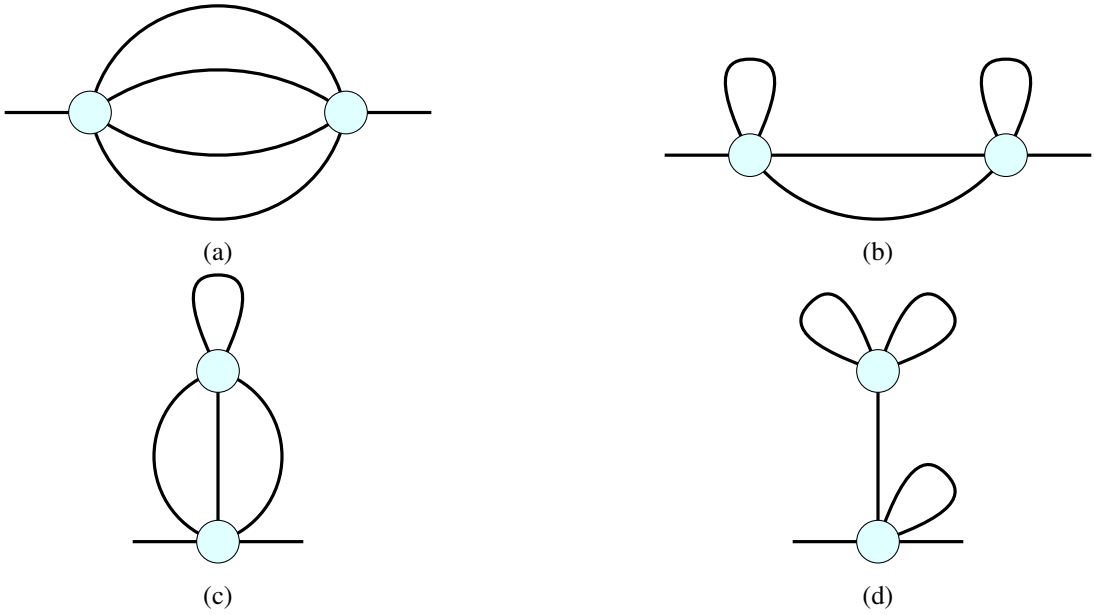


Figure 3.1: *Classes of spinfoam diagrams with two vertices and two boundary edges.*

The diagrams in each class differ by the combinatorics of the faces in the bulk. For the diagrams satisfying the 4 conditions listed above, we find: (a) two diagrams with 6 unbounded faces, one in class 3.1a and one in class 3.1d, (b) in each of class b and d there are two diagrams with four unbounded faces, and there is another one in class c, (c) finally, there are two diagrams in class a with two unbounded faces. To complete the analysis, we also mention that there are other diagrams across all four classes where the sums on the internal spins and intertwiners are fully bounded due to $SU(2)$ triangular inequalities, in which case the corresponding amplitude is a sum of a limited number of terms and the summation is trivially finite.

3.3 Numerical calculations

We study the divergences of the Lorentzian EPRL spinfoam diagrams of Figure 3.1 using `sl2cfoam-next`. The library is an open-source library written in `C` to calculate EPRL transition amplitudes numerically, with an optional user-friendly `Julia` interface. It is based on the booster decomposition of the vertex amplitude (3.1) and implements the calculation of the vertex amplitude with a homogeneous truncation of the unbounded sums over the virtual spins l_f parameterized by Δl

$$\sum_{l_f=j_f}^{\infty} \to \sum_{l_f=j_f}^{j_f+\Delta l} . \tag{3.4}$$

We compute the diagrams following the strategy described in [72, 71] where the diagram in

class a with 6 internal faces is studied in great detail. We refer to those articles for an updated analysis of the divergence of the melonic self-energy EPRL spinfoam diagram.

We fix the boundary spins j_b and intertwiners i_b and choose a truncation parameter Δl. The sums over the spin of the bulk faces j_f are unbounded and potentially divergent. We introduce a homogeneous cutoff K on these sums while we sum over all the possible values of the bulk intertwiners i_e. We take this chance to stress the difference between Δl and K. At first glance, they look similar, as they are both introduced as an upper bound of a previously unbounded summation but they are profoundly different. The parameter Δl represents the unavoidable truncation to approximate the *convergent* series in the vertex amplitude. On the other hand, K is the cutoff on the divergent sums on the bulk degrees of freedom responsible for the large volume divergences of the diagrams.

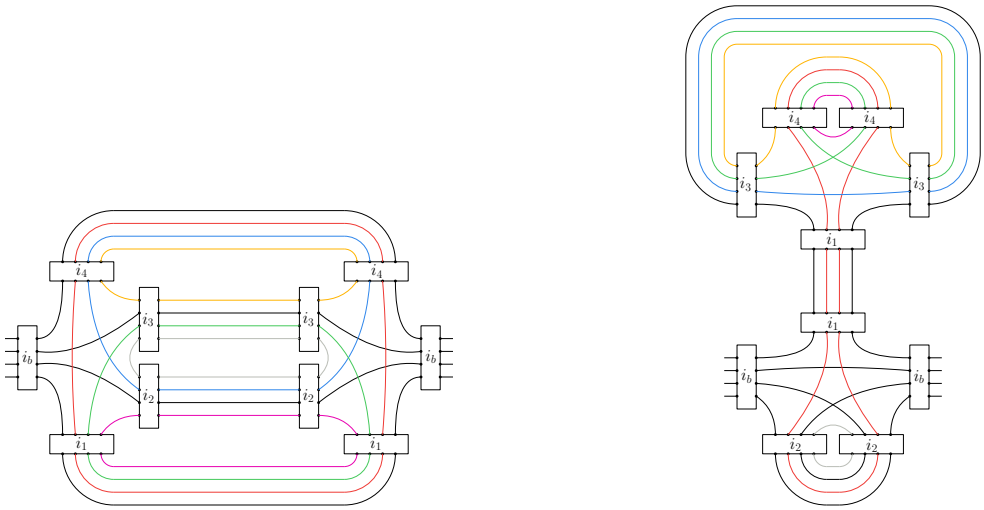
The parameters Δl and K are independent. Nevertheless, we have some technical constraints. We want a value for the cutoff K large enough to estimate the degree of divergence of the amplitude but compatible with the computational resources at our disposal. We also want a truncation Δl large enough to obtain a good numerical approximation of the amplitude. Increasing the truncation will improve the numerical estimate of the amplitude; however, although the resources necessary for the calculation will grow considerably with the truncation, the numerical accuracy will improve very little. It is essential to choose a balanced value of the parameters. Previous work suggests that choosing similar values for Δl and K gives good numerical results [71]. Given the computational resources at our disposal, we choose $\Delta l = 10$ and consider K running from 0 to 10 in half-integer steps.

In the following, we neglect the dimensional factors related to boundary faces and intertwiners, as these correspond to a constant multiplicative factor that does not affect the functional dependence of the amplitude on K.

To minimize the dependence on the parameter Δl, we use the convergence acceleration technique described in [9]. Using the value of the amplitude at fixed K obtained with the three largest truncations available Δl, $\Delta l-1$, and $\Delta l-2$, we approximate the amplitude (corresponding to the limit $\Delta l \to \infty$) with

$$A(K) \approx \frac{A(K,\Delta l)A(K,\Delta l-2) - A^2(K,\Delta l-1)}{A(K,\Delta l) - 2A(K,\Delta l-1) + A(K,\Delta l-2)} \,. \tag{3.5}$$

The `Julia` scripts used to compute the diagrams, the Wolfram's Mathematica notebooks in which we analyze the amplitudes, and the numerical values of the amplitudes are available in the public repository [135]. The scripts can be parallelized on multiple CPU cores to optimize the numerical calculations. For each spinfoam diagram, the sums over the spins of the bulk faces shared by the two vertex amplitudes (3.1) are distributed on the available tasks, using the distributed memory parallel computing implementation of `Julia`. Then, the sums over the spins of the bulk faces on every single vertex are parallelized on the threads of each task. Finally, the contraction over bulk intertwiners uses the loop vectorization technique. The code we used is scalable on a supercomputer. We provide the script used for calculating the Compute Canada clusters in the repository [135]. The total CPU usage for the computations described in this paper consisted of ~ 200 CPU hours.



(a) (b)

Figure 3.2: *6F. Wiring of the spinfoam diagrams with 6 bulk faces. We highlight the internal faces in different colors.*

3.4 Results

We focus on uniform boundary configurations to simplify the numerical calculation and optimize the computational resources at our disposal. We take all the boundary spins j_b to be the same, fixing them to the lowest possible non-trivial value $j_b = \frac{1}{2}$. We look at the amplitudes with equal boundary intertwiners and with different Immirzi parameters to minimize the dependence on the boundary data choices we have to make. We compute the amplitudes using a cutoff ranging from $K = 0$ to 10 in half-integer steps and a truncation $\Delta l = 10$. We approximate the amplitudes using (3.5).

3.4.1 spinfoam diagrams with 6 bulk faces

We briefly discuss the two diagrams with the most unbounded bulk faces. We show in Figure 3.2 the wiring diagrams highlighting the combinatorics of their internal faces.

Studying the divergence of the melonic spinfoam diagram 6F(a) required tremendous numerical effort. For a detailed presentation of the results, see [72, 71]. Similarly, we expect that the numerical analysis of the spinfoam diagram 6F(b) will also require optimization and significant computational time. For this reason, we leave a detailed study for future work. Nonetheless, it is useful to highlight a few key points concerning these diagrams before moving on to other diagrams with fewer internal faces.

The EPRL amplitude associated with the diagram 6F(a) is

$$A_{6F(A)}(j_b,\ i_b;\ K) = \sum_{j_f=0}^{K} \sum_{i_e} A_f(j_{f_1})A_f(j_{f_2})A_f(j_{f_3})A_f(j_{f_4})A_f(j_{f_5})A_f(j_{f_6})\, A_e(i_1)A_e(i_2)A_e(i_3)A_e(i_4)\times$$
$$A_v\left(j_b, j_b, j_b, j_b, j_{f_1}, j_{f_2}, j_{f_3}, j_{f_4}, j_{f_5}, j_{f_6};\ i_b, i_1, i_2, i_3, i_4\right)\times$$
$$A_v\left(j_b, j_b, j_b, j_b, j_{f_1}, j_{f_2}, j_{f_3}, j_{f_4}, j_{f_5}, j_{f_6};\ i_b, i_4, i_3, i_2, i_1\right)\ . \tag{3.6}$$

In [72], it is shown that the amplitude (3.6) with boundary intertwiners $i_b = 0$, boundary spins $j_b = \frac{1}{2}$ and Immirzi parameter $\gamma = 0.1$ diverges linearly in the cutoff K[2]. This result is obtained by fitting the amplitude as a function of the homogeneous cutoff up to $K = 10$ with truncation $\Delta l = 20$ and using a convergence acceleration technique like (3.5). Crucially, one of the observations of [71] is that the value of the amplitude estimated using (3.5) with truncation $\Delta l = 20$ and $\Delta l = 10$ are essentially identical. This justifies our choice of using a truncation $\Delta l = 10$ for the other diagrams to calculate the amplitude accurately while limiting the need for time-intensive computational resources.

Next, the EPRL amplitude associated with the diagram 6F(b) is

$$A_{6F(B)}(j_b,\ i_b;\ K) = \sum_{j_f=0}^{K} \sum_{i_e} A_f(j_{f_1})A_f(j_{f_2})A_f(j_{f_3})A_f(j_{f_4})A_f(j_{f_5})A_f(j_{f_6})\, A_e(i_1)A_e(i_2)A_e(i_3)A_e(i_4)\times$$
$$A_v\left(j_b, j_{f_1}, j_{f_1}, j_b, j_b, j_b, j_b, j_{f_2}, j_b, j_b;\ i_1, i_b, i_2, i_2, i_b\right)\times$$
$$A_v\left(j_b, j_{f_1}, j_{f_1}, j_b, j_{f_3}, j_{f_4}, j_{f_5}, j_{f_6}, j_{f_4}, j_{f_3};\ i_1, i_3, i_4, i_4, i_3\right)\ . \tag{3.7}$$

The number of terms of the amplitude 6F(b) for increasing cutoff values K grows approximately as $\sim 30 \cdot K^4$. Although this trend is slower than it is for the melonic self-energy diagram [71], there is an important difference that makes the computation of amplitude 6F(b) far more resource-demanding. In the melonic self-energy diagram 6F(a), there is a boundary face on each edge, and triangular inequalities constrain the growth of the spin of the bulk faces at each of the two vertices. Note that this does not happen in the diagram 6F(b), and as a result, we expect that the calculation of the vertices requires considerable computational resources. As K increases, eight spins out of ten in one vertex of 6F(b) are of the order of K, compared to the six in the diagram 6F(a). For the same reason, the computation time as a function of the truncation parameter Δl is also considerably higher. As a rough estimate, calculating the amplitude 6F(b) up to $K = 10$ and $\Delta l = 2$, distributed on 32 CPUs, took about 16 hours. The calculation with $\Delta l = 10$ could take weeks. Calculating the amplitude 6F(b) with sufficient precision requires an enormous investment of time and computational resources that we do not currently have access to. Therefore we leave a detailed study of this diagram for future work.

[2]It is worth mentioning that the result looks slightly different for larger values of γ. However, there are solid arguments to explain why it is a numerical artifact and that the divergence of the melonic self-energy diagram is linear in the cutoff.

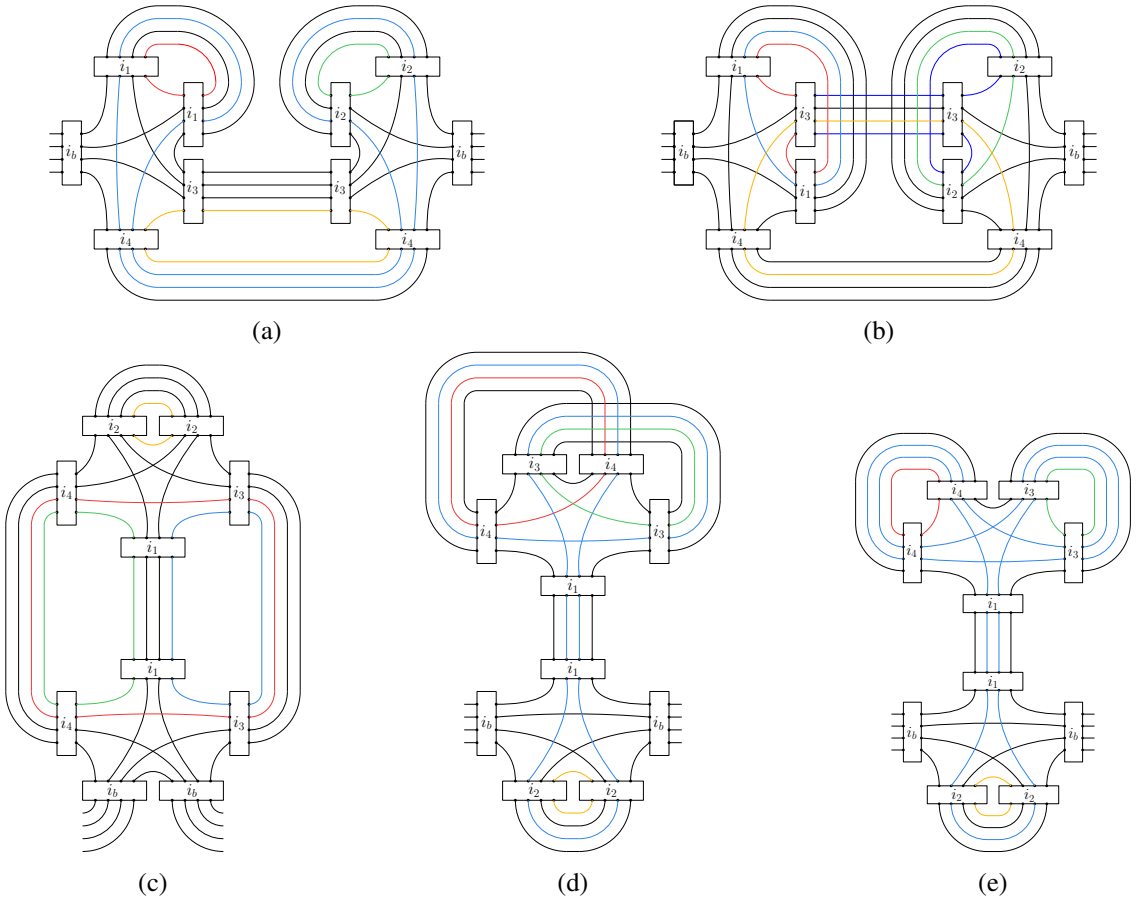


Figure 3.3: *4F. Wiring of the spinfoam diagrams with 4 bulk faces. We highlight the internal faces in different colors.*

3.4.2 spinfoam diagrams with 4 bulk faces

Five diagrams (satisfying the four conditions above) with four unbounded bulk faces. Their wiring diagrams are given in Figure 3.3, showing the combinatorics of the internal faces.

We illustrate the numerical analysis in detail for one diagram; the others are very similar, and we report the results for the other diagrams. The EPRL amplitude associated with the spinfoam diagram 4F(a) is

$$\begin{aligned} A_{4F(A)}(j_b,\ i_b;\ K) = \sum_{j_f=0}^{K} \sum_{i_e} & A_f(j_{f_1})A_f(j_{f_2})A_f(j_{f_3})A_f(j_{f_4})\, A_e(i_1)A_e(i_2)A_e(i_3)A_e(i_4)\times \\ & A_v\left(j_b, j_b, j_b, j_b, j_{f_1}, j_{f_2}, j_{f_2}, j_b, j_b, j_{f_3};\ i_b, i_4, i_3, i_1, i_1\right)\times \\ & A_v\left(j_b, j_b, j_b, j_b, j_{f_4}, j_b, j_{f_2}, j_b, j_{f_2}, j_{f_1};\ i_b, i_2, i_2, i_3, i_4\right)\ . \end{aligned} \tag{3.8}$$

We start by looking at the amplitude with both boundary intertwiners $i_b = 0$. We report the numerical values of the amplitude for different cutoffs K in Figure 3.4.

In topological BF theory, the amplitude of this diagram diverges less rapidly than the melonic self-energy diagram (see Appendix C.1), and it seems reasonable to expect that the same

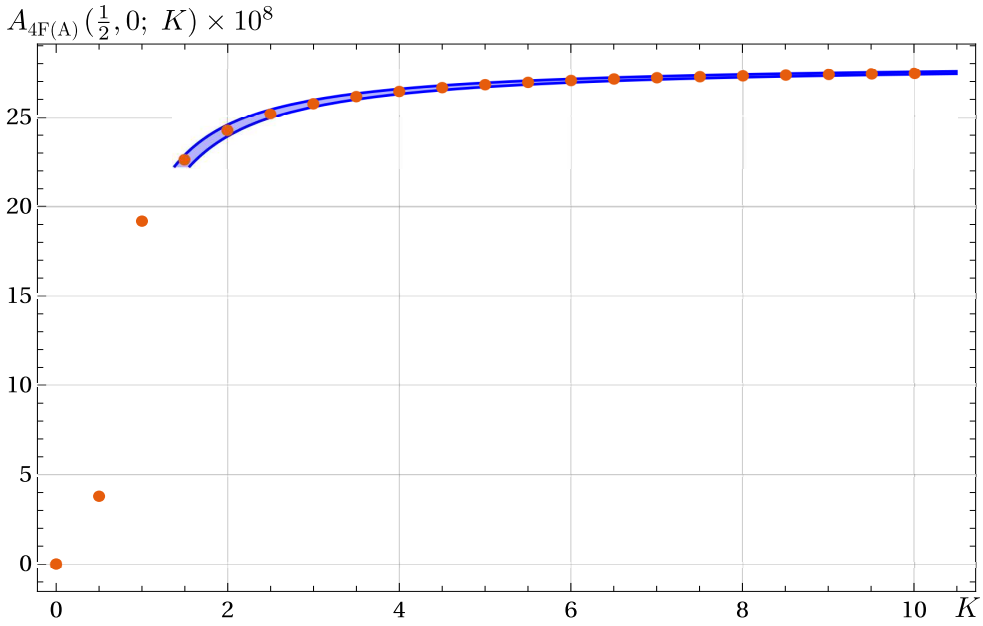


Figure 3.4: *Plot of the numerical values of the EPRL amplitude of the spinfoam* (3.1b) *with boundary spins* $j_b = \frac{1}{2}$*, boundary intertwiners* $i_b = 0$ *and Immirzi parameter* $\gamma = 0.1$*. We plot with a blue band the fit* (3.11)*.*

will be true for the EPRL spinfoam model. Since there is strong numerical evidence that the EPRL melonic self-energy diagram is linearly divergent in the cutoff, we expect the diagram 4F(a) will have a degree of divergence that is at most $\log K$.

We make a baseline fit with a four-parameter function using the `NonlinearModelFit` routine in Mathematica, with the result

$$A^{prelim}_{4F(A)}(\tfrac{1}{2}, 0; K) = -4.11 \cdot 10^{-9} \log K + 2.93 \cdot 10^{-7} - 8.61 \cdot 10^{-8} K^{-1} - 1.77 \cdot 10^{-8} K^{-2} \,. \quad (3.9)$$

The coefficient of $\log K$ is orders of magnitude smaller than the others, indicating that the amplitude may be convergent. It is generally difficult to numerically distinguish between a logarithmic and a convergent behavior at these scales since K is at most 10 in the numerical results. Increasing the cutoff by a factor of 2 results in a $\log(2) \approx 0.69$ increase of the logarithmic term but costs several more hours of computational time. The amount of resources needed for the calculation does not scale linearly in the cutoff (a rough estimate of the computational time would be K to the power of the number of unbounded summations). As a result, increasing the cutoff is not a good strategy to answer this question.

There is a more efficient way to verify the convergence of the diagram. We amplify the divergence of the diagram by replacing the face amplitude in (3.3)

$$A_f(j_f) = 2j_f + 1 \quad \longrightarrow \quad A^{amp}_f(j_f) = (2j_f + 1)^2 \,. \quad (3.10)$$

We can verify that of the four unbounded sums in (3.8), only three are unbounded since $SU(2)$ triangular inequalities limit the spin on one of the internal faces (specifically, the yellow internal face which shares an edge with three fixed boundary faces). Therefore, the degree of divergence of the amplified amplitude increases by K^3. We evaluate the amplified amplitude

numerically using the same parameters and boundary data as the original one. We fit the amplified amplitude with a cubic polynomial in the cutoff and find that the coefficient of K^3 is negligible by two orders of magnitude with respect to the others[3]. The amplified amplitude diverges quadratically as K^2 in the cutoff, confirming that the original amplitude had a negative degree of divergence. To avoid any doubt, we also repeat the calculation with an amplified face amplitude $A_f^{amp}(j_f) = (2j_f + 1)^3$, with the expected result that the amplitude is again amplified by a factor of K^3, now diverging as the fifth power K^5 of the cutoff, confirming that the original amplitude (3.8) is convergent[4].

Given the strong indications that the leading order dependence on the cutoff scales as K^{-1}, we fit the initial amplitude again, omitting the $\log K$ term, and find with a 95% confidence interval

$$A_{4F(A)}(\tfrac{1}{2}, 0; K) = (2.799, 2.806) \cdot 10^{-7} - (5.387, 4.790) \cdot 10^{-8}\, K^{-1} - (5.552, 4.501) \cdot 10^{-8}\, K^{-2}\,. \tag{3.11}$$

To remove the cutoff, it is easy to take the limit $K \to \infty$ with the result $A_{4F(A)} = (2.799, 2.806) \cdot 10^{-7}$.

The result concerning the degree of divergence of this diagram seems to be robust. We also studied the amplitude for different boundary intertwiners $i_b = 1$ and found a similar outcome. We fit the amplitude and see with a 95% confidence interval

$$A_{4F(A)}(\tfrac{1}{2}, 1; K) = (5.969, 5.983) \cdot 10^{-8} - (2.123, 2.006) \cdot 10^{-8}\, K^{-1} - (0.910, 1.117) \cdot 10^{-8}\, K^{-2}\,. \tag{3.12}$$

Likewise, we study the amplitude with boundary intertwiners $i_b = 0$ and a larger Immirzi parameter $\gamma = 1$. Again, we find that the amplitude is convergent,

$$A_{4F(A)}(\tfrac{1}{2}, 0; K)^{\gamma=1} = (1.080, 1.084) \cdot 10^{-12} - (1.680, 1.640) \cdot 10^{-12}\, K^{-1} - (1.146, 1.229) \cdot 10^{-12}\, K^{-2}\,. \tag{3.13}$$

We summarize these results in Figure 3.5.

We perform a similar analysis to calculate the amplitudes of the other diagrams with 4 internal faces in Figure 3.3. In this case, we report the data, fits, and plots only for $\gamma = 0.1$ and boundary intertwiners $i_b = 0$, to keep the presentation concise, but we repeated the calculation for $\gamma = 1$ and $i_b = 1$ finding qualitatively similar results. The amplitude of the spinfoam diagram 4F(b) is

$$\begin{aligned} A_{4F(B)}\,(j_b,\ i_b;\ K) = \sum_{j_f=0}^{K} \sum_{i_e} & A_f(j_{f_1}) A_f(j_{f_2}) A_f(j_{f_3}) A_f(j_{f_4})\, A_e(i_1) A_e(i_2) A_e(i_3) A_e(i_4) \times \\ & A_v\left(j_b, j_b, j_b, j_b, j_b, j_{f_1}, j_b, j_{f_2}, j_{f_3}, j_{f_2};\ i_b, i_4, i_1, i_3, i_1\right) \times \\ & A_v\left(j_b, j_b, j_b, j_b, j_{f_2}, j_{f_4}, j_b, j_{f_2}, j_{f_1}, j_b;\ i_b, i_2, i_3, i_2, i_4\right)\,, \end{aligned} \tag{3.14}$$

[3] Fitting with a cubic polynomial, we find a ratio between the coefficients of K^3 and K^2 of 0.01, indicating that the coefficient of K^3 is irrelevant.

[4] Fitting with a degree six polynomial, we find a ratio of 0.05 between the coefficients of K^6 and K^5, suggesting that the coefficient of K^6 is irrelevant.

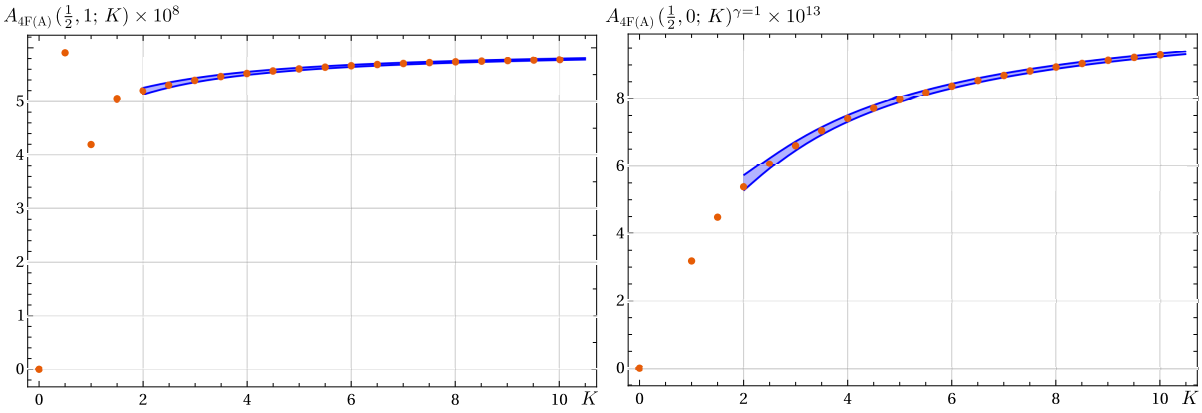


Figure 3.5: *Plot of the numerical values of the EPRL amplitude of the spinfoam diagram 4F(a) with boundary spins $j_b = \frac{1}{2}$ (orange dots) and the corresponding fit (blue band)* (3.12) *and* (3.13). *Left panel: boundary intertwiners $i_b = 1$ and Immirzi parameter $\gamma = 0.1$. Right panel: boundary intertwiners $i_b = 0$ and Immirzi parameter $\gamma = 1$. The strange amplitude behavior with $i_b = 1$ and $K = \frac{1}{2}$ is a numerical artifact of this diagram because the cutoff value is smaller than the intertwiners.*

and it appears to be convergent. The numerical results are shown in Figure 3.6, together with the best-fit curve with a 95% confidence interval given by

$$A_{4F(B)}(\tfrac{1}{2}, 0; K) = (1.147, 1.150) \cdot 10^{-7} - (2.933, 2.696) \cdot 10^{-8}\, K^{-1} - (7.095, 2.929) \cdot 10^{-9}\, K^{-2}\,. \tag{3.15}$$

The amplitude of the spinfoam diagram 4F(c) is

$$\begin{aligned} A_{4F(C)}(j_b,\ i_b;\ K) = \sum_{j_f=0}^{K} \sum_{i_e} & A_f(j_{f_1}) A_f(j_{f_2}) A_f(j_{f_3}) A_f(j_{f_4})\, A_e(i_1) A_e(i_2) A_e(i_3) A_e(i_4) \times \\ & A_v\left(j_{f_1}, j_b, j_b, j_{f_2}, j_b, j_b, j_{f_3}, j_b, j_b, j_b;\ i_1, i_4, i_b, i_b, i_3\right) \times \\ & A_v\left(j_{f_2}, j_b, j_b, j_{f_1}, j_b, j_b, j_{f_3}, j_{f_4}, j_b, j_b;\ i_1, i_3, i_2, i_2, i_4\right)\,, \end{aligned} \tag{3.16}$$

and again, we find a clear indication that this amplitude is convergent. We perform a fit and find with a 95% confidence interval

$$A_{4F(C)}(\tfrac{1}{2}, 0; K) = (1.0269, 1.0273) \cdot 10^{-7} - (7.595, 7.566) \cdot 10^{-8}\, K^{-1} + (1.199, 1.250) \cdot 10^{-8}\, K^{-2}\,. \tag{3.17}$$

Figure 3.6 represents the data and the fit.

Finally, we look at the amplitudes 4F(d) and 4F(e). The amplitudes associated with these diagrams are

$$\begin{aligned} A_{4F(D)}(j_b,\ i_b;\ K) = \sum_{j_f=0}^{K} \sum_{i_e} & (-1)^{\chi_D} A_f(j_{f_1}) A_f(j_{f_2}) A_f(j_{f_3}) A_f(j_{f_4})\, A_e(i_1) A_e(i_2) A_e(i_3) A_e(i_4) \times \\ & A_v\left(j_b, j_{f_1}, j_{f_1}, j_b, j_b, j_b, j_b, j_{f_2}, j_b, j_b;\ i_1, i_b, i_2, i_2, i_b\right) \times \\ & A_v\left(j_b, j_{f_1}, j_{f_1}, j_b, j_b, j_{f_3}, j_{f_1}, j_b, j_{f_4}, j_b;\ i_1, i_3, i_4, i_3, i_4\right)\,, \end{aligned} \tag{3.18}$$

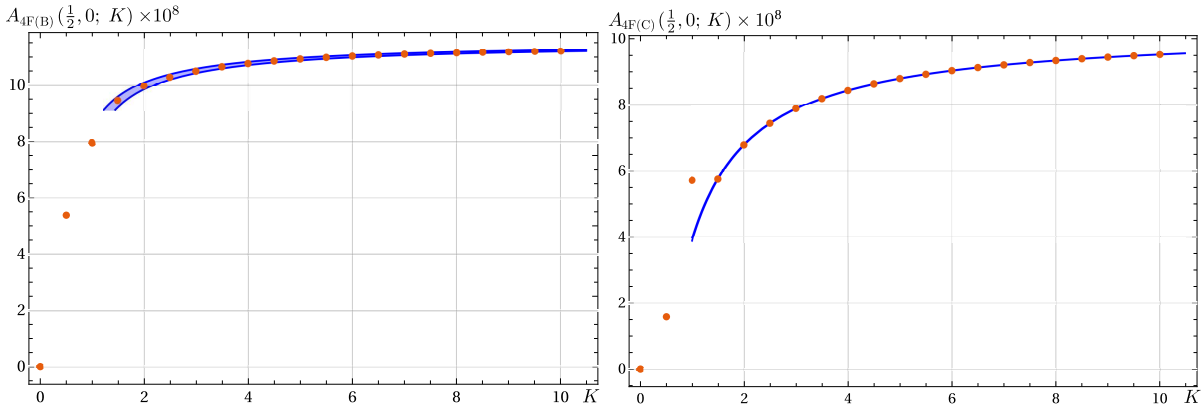


Figure 3.6: *The plot of the numerical values of the EPRL amplitudes with boundary spins* $j_b = \frac{1}{2}$*, boundary intertwiners* $i_b = 0$ *and Immirzi parameter* $\gamma = 0.1$*. We plot the corresponding fits with a blue band. Left panel: spinfoam diagram 4F(b) and fit* (3.15)*. Right panel: spinfoam diagram 4F(c) and fit* (3.17)*.*

and

$$A_{4F(E)}\,(j_b,\ i_b;\ K) = \sum_{j_f=0}^{K}\sum_{i_e}(-1)^{\chi_E}A_f(j_{f_1})A_f(j_{f_2})A_f(j_{f_3})A_f(j_{f_4})\,A_e(i_1)A_e(i_2)A_e(i_3)A_e(i_4)\times$$
$$A_v\left(j_b, j_{f_1}, j_{f_1}, j_b, j_b, j_b, j_b, j_{f_2}, j_b, j_b;\ i_1, i_b, i_2, i_2, i_b\right)\times$$
$$A_v\left(j_b, j_{f_1}, j_{f_1}, j_b, j_{f_3}, j_{f_1}, j_{f_1}, j_b, j_{f_1}, j_4;\ i_1, i_3, i_3, i_4, i_4\right)\ , \tag{3.19}$$

where $\chi_D = 2j_{f_3}$ and $\chi_E = 2j_b + 2j_{f_1} + j_{f_3} + j_{f_4}$ are phases coming from the decomposition of the amplitude. Once again, the numerical evaluation of the amplitudes (3.18) suggests the amplitudes are convergent. We perform a fit with a 95% confidence interval and find

$$A_{4F(D)}(\tfrac{1}{2},0;K) = -(1.778, 1.776)\cdot 10^{-7} + (5.097, 6.681)\cdot 10^{-9}\ K^{-1} - (2.999, 2.720)\cdot 10^{-8}\ K^{-2}\ , \tag{3.20}$$

$$A_{4F(E)}(\tfrac{1}{2},0;K) = (3.586, 3.592)\cdot 10^{-7} + (8.172, 13.486)\cdot 10^{-9}\ K^{-1} - (4.360, 3.425)\cdot 10^{-8}\ K^{-2}\ . \tag{3.21}$$

We report these fits and compare them with the data points in Figure 3.7.

It is somewhat surprising that the degree of divergence of these last two diagrams is at most the same as for the first 3 diagrams with four internal faces (and possibly even with a smaller degree of divergence given by K^{-2}). Naively, one might expect the degree of divergence to be directly related to the number of unbounded summations. If one looks carefully, diagrams 4F(d) and 4F(e) are the only ones among the diagrams in Figure 3.3 with four unbounded internal faces—the other three diagrams always have one face bounded by triangular inequalities (specifically, the yellow internal face in each diagram). (On the other hand, all of the diagrams in Figure 3.3 in the $SU(2)$ BF theory have the same degree of divergence.) This counterintuitive result indicates that the degree of divergence depends in a complicated way on the components of the diagram.

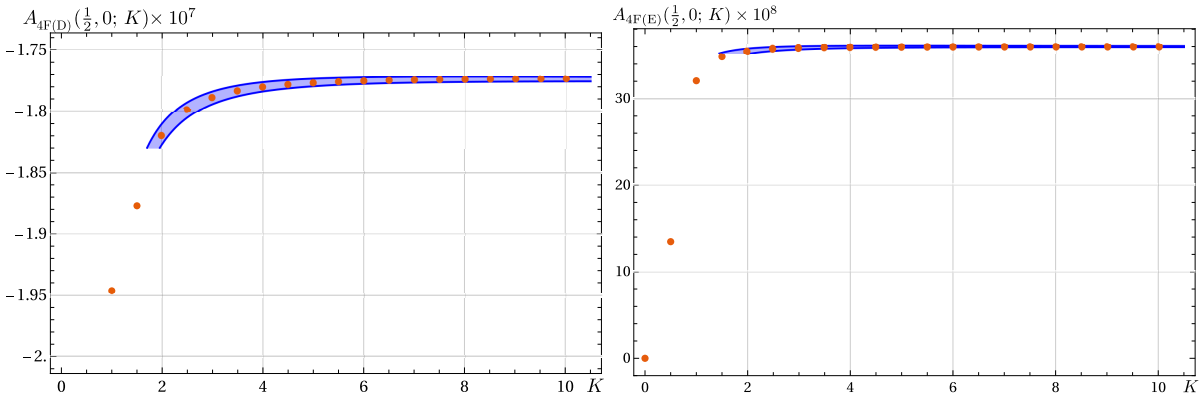
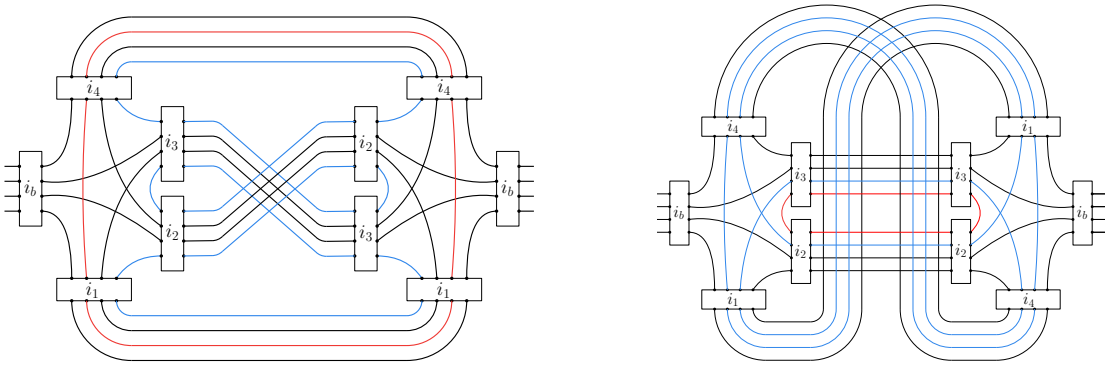


Figure 3.7: *Plot of the numerical values of the EPRL amplitudes with boundary spins* $j_b = \frac{1}{2}$*, boundary intertwiners* $i_b = 0$ *and Immirzi parameter* $\gamma = 0.1$*. We plot with a blue band the corresponding fits. Left panel: spinfoam diagram 4F(d) and fit* (3.20)*. Right panel: spinfoam diagram 4F(e) and fit* (3.21)*.*

3.4.3 spinfoam diagrams with 2 bulk faces

Only two diagrams satisfy the four above conditions with two unbounded bulk faces. These are shown in Figure 3.8, where we highlight the combinatorics of the internal faces. Note that other diagrams exist with two internal faces satisfying the four conditions. Still, in all other such diagrams, the spin of the internal faces is bounded due to $SU(2)$ triangular inequalities. Also, in this case, we report only the calculation with $\gamma = 0.1$ and boundary intertwiners $i_b = 0$.



(a) (b)

Figure 3.8: *2F. Wiring of the spinfoam diagrams with 2 bulk faces. We highlight the internal faces in different colors.*

The amplitudes of the diagram 2F(a) and 2F(b) are

$$\begin{aligned} A_{2F(A)}(j_b,\ i_b;\ K) = \sum_{j_f=0}^{K} \sum_{i_e} & A_f(j_{f_1}) A_f(j_{f_2})\, A_e(i_1) A_e(i_2) A_e(i_3) A_e(i_4) \times \\ & A_v\left(j_b, j_b, j_b, j_b, j_{f_1}, j_b, j_{f_2}, j_{f_1}, j_b, j_{f_1};\ i_b, i_1, i_2, i_3, i_4\right) \times \\ & A_v\left(j_b, j_b, j_b, j_b, j_{f_1}, j_b, j_{f_2}, j_{f_1}, j_b, j_{f_1};\ i_b, i_4, i_2, i_3, i_1\right), \end{aligned} \tag{3.22}$$

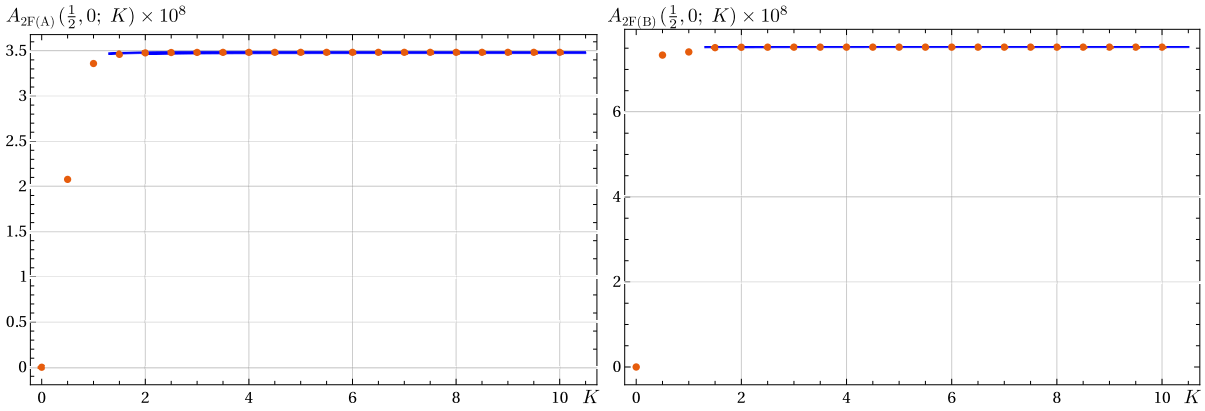


Figure 3.9: *The plot of the numerical values of the EPRL amplitudes with boundary spins* $j_b = \frac{1}{2}$*, boundary intertwiners* $i_b = 0$ *and Immirzi parameter* $\gamma = 0.1$*. We plot with a blue band the corresponding fits. Left panel: spinfoam diagram 2F(a) and fit* (3.24)*. Right panel: spinfoam diagram 2F(b) and fit* (3.25)*.*

and

$$A_{2F(B)}(j_b,\ i_b;\ K) = \sum_{j_f=0}^{K}\sum_{i_e} A_f(j_{f_1})A_f(j_{f_2})\,A_e(i_1)A_e(i_2)A_e(i_3)A_e(i_4)\times \\ A_v\left(j_b,j_b,j_b,j_b,j_b,j_{f_1},j_{f_1},j_{f_2},j_{f_1},j_b;\ i_b,i_1,i_2,i_3,i_4\right)\times \\ A_v\left(j_b,j_b,j_b,j_b,j_b,j_{f_1},j_{f_1},j_{f_2},j_{f_1},j_b;\ i_b,i_1,i_3,i_2,i_4\right)\ . \tag{3.23}$$

Both amplitudes are convergent. With a fit with a 95% confidence interval, we find

$$A_{2F(A)}(\tfrac{1}{2},0;K) = (3.481, 3.482)\cdot 10^{-8} + (1.643, 2.689)\cdot 10^{-10}\ K^{-1} - (6.924, 5.082)\cdot 10^{-10}\ K^{-2}\ , \tag{3.24}$$

and

$$A_{2F(B)}(\tfrac{1}{2},0;K) = (7.5237, 7.5243)\cdot 10^{-8} + (7.584, 12.386)\cdot 10^{-11}\ K^{-1} - (3.245, 2.400)\cdot 10^{-10}\ K^{-2}\ , \tag{3.25}$$

and we compare them with the numerical values of the amplitudes in Figure 3.9.

Since these diagrams contain fewer faces than diagrams in Figure 3.3, we expect them to converge more rapidly, and the plots showing the numerical results indeed suggest this is the case. In an attempt to quantify the rate of convergence of these amplitudes, we amplify the divergence of the amplitudes by replacing the face amplitude in (3.3) as

$$A_f(j_f) = 2j_f + 1 \quad \longrightarrow \quad A_f^{amp}(j_f) = (2j_f+1)^4\ . \tag{3.26}$$

With this amplification, we find a linearly divergent amplitude, suggesting that the original amplitude has a degree of divergence of -5.

3.5 Conclusions

Without a cosmological constant, spinfoam models are affected by infrared divergences. They are associated with significant volume divergences and originate from the unbounded sums

over the spinfoam's bulk degrees of freedom.

Calculating the radiative corrections to the Lorentzian EPRL spinfoam propagator is necessary to study the theory's renormalization and explore the continuum limit. Two-vertex diagrams provide the first contributions to a vertex expansion. Previous studies focused on the 'melonic' self-energy diagram 6F(a), with numerical evidence suggesting it diverges linearly with respect to the spin cutoff [72, 71], but there exist other diagrams with two vertices that also need to be computed to include all contributions at this order in the vertex expansion.

In this paper, we studied two-vertex diagrams with two boundary edges, four boundary faces containing both boundary edges, and whose propagators preserve the order of the faces. We find eight additional diagrams that we classify based on the number of internal faces.

We focus on the diagrams with four and two bulk faces (see Figure 3.3 and Figure 3.8). These diagrams are expected to be subdominant because they have fewer internal faces with respect to the diagram 6F(a), and we find numerical evidence that the diagrams with four or two internal faces appear to be convergent. We also point out the existence of a second diagram with six internal faces 6F(b), but leave the numerical calculation of this diagram for future work since it requires computational resources we currently do not have access to.

For simplicity, in the numerical calculations, we fix the boundary spins to $\frac{1}{2}$, the first non-trivial option. We consider both possible boundary intertwiners (0 or 1), and we repeat the calculations, setting the Immirzi parameter to $\gamma = 0.1$ and also $\gamma = 1$ to minimize the dependence of our analysis on the specific choice of boundary data. We introduce a homogenous cutoff K on the summations over the spins of the bulk faces, and we compute the EPRL spinfoam amplitudes using the `sl2cfoam-next` framework as a function of K. For the diagrams with four or two internal faces, all amplitudes have a degree of divergence (in K) smaller than the melonic self-energy diagram, confirming expectations. Somewhat surprisingly, these diagrams all appear convergent, with numerics suggesting a degree of divergence of K^{-1} for diagrams with four internal faces and even faster convergence for diagrams with two internal faces. If this is the case, then within the class of graphs we consider, seven of the nine diagrams are finite (and of the two remaining diagrams, both have six internal faces, and one is known to diverge while the other has not been computed yet).

This result may have some significant ramifications for renormalization. While we have only considered a specific class of diagrams, the fact that many are finite provides some encouragement that the theory might be renormalizable. On the other hand, there seems to be a rapid growth in the number of diagrams, significantly as the number of vertices increases. Whether the spinfoam model is renormalizable may hinge on which of these two effects dominates—of course, it will be necessary to consider diagrams with more vertices to answer this question, and we leave this task for future work.

Related to this point is the question of which spinfoam diagrams should be considered in a given calculation. In this paper, we imposed some conditions that reduced the number of diagrams we considered, although it may be necessary to relax these conditions (for more on this point, see Appendix C.2). This question is closely related to the symmetries of the spinfoam model: what exactly are the symmetries that should be imposed on a spinfoam model, and do they force specific contributions to vanish?

It may be fruitful to consider the questions of renormalization and symmetries from the

perspective of group field theory, whose partition function can be expanded in the usual way with the result giving a spinfoam model [136, 137]. Imposing symmetries directly in the group field theory action may be more accessible. In turn, this can be used to determine which diagrams will contribute simply by seeing whether they appear when calculating the Feynman expansion for the partition function. A further advantage is that this approach can also be used to calculate symmetry factors for different spinfoam diagrams, which may not be as easy to determine from the spinfoam model.

Finally, an important goal for future research would be to find a simple formula giving the degree of divergence of any diagram given its number of vertices, number of bulk edges, and number of bulk faces (and perhaps other relevant topological or combinatorial information); this could be useful for example to determine which interaction terms (in a group field theory language) are applicable/irrelevant. Our work suggests a clear dependence of the degree of divergence on the number of bulk faces, with more internal faces producing greater divergence. For example, the melonic self-energy diagram appears to diverge linearly and has six bulk faces. In comparison, diagrams with four internal faces seem to have a degree of divergence K^{-1}, and diagrams with two internal faces converge even faster still. Nevertheless, it is essential to compute a broader range of diagrams to infer a general formula for the EPRL degree of divergence; we leave this for future work.

Chapter 4

Summing bulk quantum numbers with Monte Carlo in spinfoam theories

Statement of co-authorship

This Chapter presents the paper [138], written by me and Pietro Dona. It was published in *Physical Review D* on 9 May 2023. The idea of applying Monte Carlo methods to investigate infrared divergences came after studying such a technique to apply it to the multidimensional discrete summations appearing in spinfoams. This method was first applied in [78] (to which Chapter 5 is dedicated). After several weeks of attempts, I could finally reproduce the results of [71] using Monte Carlo methods. I noticed that it required a small fraction of computational time. After sharing such findings with Pietro Dona, we decided to apply such a technique to compute the vertex renormalization spinfoam amplitude. The latter was precisely one of the spinfoam diagrams I have wanted to investigate since I started my Ph.D. I took care of the numerical computations, contributing to drafting sections. I also realized some of the plots and images. I estimate the percentage of the total work conducted by me between 50% and 70%. This manuscript corresponds to the article's *Physical Review D*-published version, which agrees with the APS copyright policies.

4.1 Introduction

spinfoam theory is the Lorentz covariant version of loop quantum gravity (LQG) and provides a tentative background independent path integral for gravity. It gives dynamics to LQG kinematical states defining transition amplitudes between spin network states [1, 74]. The most promising spinfoam theory is the EPRL-FK model [25, 28]. Various generalizations include the extension to general triangulations [82], the inclusion of a cosmological constant [35, 36], and boundary with different signature [37]. These theories promisingly connect with discrete general relativity in the double limit of finer discretization and large areas [29, 30, 31, 32, 27, 33].

The field has recently undergone an explosion of numerical methods, providing new tools to address many open questions of the theory. We can compute expectation values and fluctuations of operators in the large spins regime using the complex saddle point analysis and

the integration on a Lefschetz thimble using Markov chain Monte Carlo [32, 57]. It is possible to verify that Regge geometries emerge in the large-scale and small Immirzi parameter regimes. A similar result can also be obtained from effective spinfoam models [139, 51]. Finally, `sl2cfoam` (and its latest iteration `sl2cfoam-next`) is an open-source framework to compute EPRL spinfoam amplitudes [47, 48]. It is based on a divide-and-conquer strategy and a booster decomposition of the vertex amplitude [42]. The library was already employed to explore the large quantum numbers regime [48, 140, 9], the infrared divergences of the theory [71, 133], the black-to-white hole transition [141], and correlations in the early universe [78]. Very recently, a hybrid approach taking advantage of all the available techniques was also proposed [49].

This work overcomes one of the principal limitations of `sl2cfoam-next`. The library provides an optimized and efficient framework to compute all the constituents of a spinfoam transition amplitude. Nevertheless, numerically computing spinfoam amplitudes with many internal faces is prohibitively taxing. There are too many objects to compute as their number scales exponentially with the number of internal faces. We overcome this problem by evaluating the sums over the internal quantum numbers using statistical frameworks. We find the possibility of using uniform sampling in Monte Carlo surprisingly effective. We can compute amplitudes with slightly better than 1% precision by considering a sampling five orders of magnitude smaller than the total amount of terms of the sum.

We apply this novel technique to compute the melonic self-energy and the vertex renormalization amplitude in the $SU(2)$ BF and EPRL theories. These two amplitudes are the perfect laboratory to test the effectiveness of Monte Carlo as they possess many internal faces. These spinfoam amplitudes are believed to be divergent, and their renormalization is crucial to define the continuum limit of the EPRL theory. In the case of the melonic self-energy, we find excellent agreement with the numerical results in the literature [72, 71], which do not use stochastic methods. This work contains the first computation of the Lorentzian EPRL vertex renormalization amplitude. We surprisingly find numerical evidence for its convergence.

The scripts we use to compute the amplitudes and the notebooks to analyze the data are publicly available at the repository [142]. We perform most of our calculations on the *Narval* cluster of the Digital Research Alliance of Canada.

4.2 spinfoam transition amplitudes

We write a spinfoam transition amplitude starting from a 2-complex Δ simplicial triangulation of the space-time manifold decorated with LQG quantum numbers. Each face is colored with a spin j_f and each edge with an intertwiner i_e.

The spinfoam transition amplitude A_Δ is the product of local fundamental amplitudes: a face amplitude $A_f(j_f)$, an edge amplitude $A_e(i_e)$, and a vertex amplitude $A_v\left(j_f,\ i_e\right)$. Finally, we sum over all the possible quantum numbers associated with the bulk of the 2-complex

$$A_\Delta = \sum_{j_f=0}^{\infty} \sum_{i_e} \prod_f A_f(j_f) \prod_e A_e(i_e) \prod_v A_v\left(j_f,\ i_e\right) . \tag{4.1}$$

This work focuses on two spinfoam theories: the topological BF $SU(2)$ model and the Lorentzian EPRL model. We introduce them here schematically and report their detailed definition in Appendix D.1. We refer to reviews [74] or books [1] for a more complete and pedagogical introduction. We use the same notation for the vertex amplitudes in the two models. It is convenient to avoid overburdening the notation and not repeat the same equations twice. We will stress the difference between the two models if necessary. The vertex amplitude for the topological model is defined as

$$A_v\left(j_f,\ i_e\right) = \{15j\}(j_f; i_e)\ , \tag{4.2}$$

where the $\{15j\}(j_f; i_e)$ is a $SU(2)$ invariant depending on the ten spins and five intertwiners coloring the spinfoam vertex. We work with the booster functions decomposition of the Lorentzian EPRL spinfoam model introduced in [42]. In this form, the EPRL vertex amplitude is a superposition of $\{15j\}$ symbols weighted by booster functions B_4^γ.

$$A_v\left(j_f,\ i_e\right) = \sum_{l_f=j_f}^{\infty} \sum_{k_e} \{15j\}(j_f, l_f) \prod_{e=2}^{5} (2k_e+1) B_4^\gamma\left(l_f, j_f; i_e, k_e\right)\ . \tag{4.3}$$

The presence of the booster functions is the striking difference between the amplitudes (4.2) and (4.3). They encode the imposition of the simplicity constraints and the explicit dependence of the theory from the Immirzi parameter γ. They possess a compelling geometrical interpretation of boosted tetrahedra [88]. The edge and face amplitudes are fixed, requiring the correct composition of spinfoam amplitudes [83].

$$A_e\left(i_e\right) = 2i_e + 1\ , \qquad \text{and} \qquad A_f\left(j_f\right) = 2j_f + 1\ . \tag{4.4}$$

Depending on the details of the 2-complex could be necessary to also multiply by some extra phase and edge-related $SU(2)$ invariants depending on the spin and intertwiners quantum numbers. They result from our decision to work with a specific recoupling scheme in the vertex amplitudes. We refer to the review [9] for a step-by-step guide on computing them.

We perform all the numerical calculations using `sl2cfoam-next`, the state-of-the-art code, to compute spinfoam amplitudes with a computer. The library is open source and written in `C`. It is based on the booster decomposition of the EPRL vertex amplitude, optimizing the available computational resources. We refer to the original paper [48], the review [9], or the book chapter [143] for a detailed description.

One of the main ingredients we mention here is the introduction of a homogeneous truncation parameter Δl to approximate the unbounded convergent sums over the virtual spins l_f in (4.3).

$$\sum_{l_f=j_f}^{\infty} \quad \longrightarrow \quad \sum_{l_f=j_f}^{j_f+\Delta l}\ . \tag{4.5}$$

Despite the notation, we emphasize that the truncation parameter Δl is independent of the 2-complex Δ.

4.3 Summing bulk degrees of freedom with Monte Carlo

The library `sl2cfoam-next` [48] computes EPRL vertex amplitudes (4.3) very fast and efficiently[1]. Unfortunately, it is insufficient to compute a general spinfoam amplitude with many vertices and internal faces. The number of vertex amplitudes we have to calculate, assemble, and sum grows exponentially with the number of bulk faces. We can convince ourselves this is a severe problem with a back-of-the-envelope calculation. Imagine you want to compute an amplitude with F internal faces, and all the spins associated with the inner faces have some characteristic value J. We must loop through all $(2J+1)^F$ possible values that the internal spins can assume and compute all the vertex amplitudes to calculate the amplitude. Let us assume, optimistically, that we need just $1\mu s$ of CPU time to obtain them (the actual time is orders of magnitude larger). Suppose we want to calculate an amplitude with 10 internal faces and average spins of order 10. To perform this calculation, we need approximately $21^{10}\mu s \approx 6$ months of CPU time, which is a lot of time considering our modest requirements and optimistic hypothesis.

We overcome this problem using Monte Carlo to perform the sum over the bulk spins. We rewrite the spinfoam amplitude (4.1) in terms of partial amplitudes

$$A_\Delta = \sum_{j_f=0}^{\infty} a(j_f) \qquad \text{with} \qquad a(j_f) = \sum_{i_e} \prod_f A_f(j_f) \prod_e A_e(i_e) \prod_v A_v\left(j_f,\ i_e\right) . \tag{4.6}$$

We include the sum over the bulk intertwiners in the partial amplitudes $a(j_f)$. Those sums are always finite for fixed bulk spins, and we perform them leveraging the tensorial structure of `sl2cfoam-next` without resorting to Monte Carlo methods. We omit the evident dependence from the 2-complex Δ of the partial amplitude.

The partial amplitude vanishes if some spins j_f do not satisfy triangular inequalities[2]. We restrict the sum over the bulk spins in (4.6) to the set of spins satisfying triangular inequalities that we indicate as $\mathcal{I}_\Delta$. In this way, we eliminate the majority of trivially vanishing partial amplitudes. This step is convenient to obtain an efficient Monte Carlo amplitude estimate.

Generally, $\mathcal{I}_\Delta$ is unbounded. Therefore, applying Monte Carlo to estimate the amplitude is impossible. We circumvent this limitation subdividing $\mathcal{I}_\Delta$ into layers $\mathcal{J}_k$.

$$\mathcal{J}_k = \left\{ j_f \in \mathcal{I}_\Delta | \max j_f = k \right\} . \tag{4.7}$$

By definition, each layer is finite. Different layers do not overlap $\mathcal{J}_k \cap \mathcal{J}_{k'} = \emptyset$, and the union of all of them forms the original set $\mathcal{I}_\Delta = \bigcup_{k=0}^{\infty} \mathcal{J}_k$. We reorganize the spinfoam amplitude as a sum over layers of layer amplitudes

$$A_\Delta = \sum_{k=0}^{\infty} S_k \qquad \text{with} \qquad S_k = \sum_{j_f \in \mathcal{J}_k} a(j_f) . \tag{4.8}$$

[1] It can evaluate the topological BF $SU(2)$ vertex amplitude too.

[2] For example, if j_1, j_2, j_3, and j_4 are the four spins associated with the faces contained in an edge of Δ and

$$\text{Max}(|j_1 - j_2|, |j_3 - j_4|) > \text{Min}(j_1 + j_2, j_3 + j_4) \Rightarrow a_\Delta(j_f) = 0$$

then the set of intertwiners associated with that edge is empty, and the partial amplitude trivially vanishes.

Each layer amplitude S_k is defined as a sum with a finite number of terms (that, however, proliferates with k). We can approximate them with Monte Carlo S_k^{mc} using the procedure described in Appendix D.2 with a fixed amount of samples N_{mc}. In general, the number of layers is infinite[3]. We cut off the sum over the layer to a top layer K. This prescription is equivalent to introducing a homogeneous cutoff K to all the bulk spins. The Monte Carlo approximation of a spinfoam amplitude with a cutoff K is given by

$$A_\Delta^{mc}(K) = \sum_{k=0}^{K} S_k^{mc} \ . \tag{4.9}$$

The S_k^{mc} calculation requires a discrete random uniform probability distribution over the layer $\mathcal{J}_k$. In principle, we could map the layer in an interval of integers, define a uniform distribution there and map it back to $\mathcal{J}_k$. However, this prescription is very unpractical. Therefore, we prefer to define the uniform distribution in an alternative way.

We extract one real number from a continuous uniform distribution in $[0, k + 0.5]$ for each bulk face. Then, we floor them to half-integers and check if they belong to the layer $\mathcal{J}_k$. If they do, we accept them as a random layer sample. If they do not, we discard them and repeat the procedure. We summarize this procedure in the flowchart 1.

Algorithm 1 Random sampling of spins in $\mathcal{J}_k$

procedure RANDOMSAMPLE(k,$\mathcal{J}_k$)
  while true do ▷ repeat until we find a good candidate
    $j_f \leftarrow$ extract a real number from a uniform distribution in $[0, k + 0.5]$ for each face
    $j_f \leftarrow$ floor them to half-integers and interpret them as spins
    if all spins j_f are smaller than k **then**
      continue
    if any spin j_f do not satisfy triangular inequalities of $\mathcal{J}_k$ **then**
      continue
    return j_f ▷ the spins belong to the layer $\mathcal{J}_k$

In the application we present in Section 4.5, we perform an explicit test to show that the samples extracted with this algorithm are uniformly distributed in $\mathcal{J}_k$.

We acknowledge that Algorithm 1 is not optimal. To scale it up to more complicated amplitudes, we must improve it considerably. We leave this task to future work. Since they share the edge structure, the sampling Algorithm 1 is the same for both spinfoam models we study.

We conclude this section by showing a pseudocode representation (Algorithm 2) of the `Julia` scripts that implement the Monte Carlo estimate of the spinfoam amplitude (4.9). The complete Julia scripts are available in the repository [142].

[3]In some exceptional cases, the layers are all empty from a particular value of k forward because of the triangular inequalities involving both bulk and boundary spins. It is the case of the Δ_3, and the Δ_4 triangulations studied numerically in [31, 9]

Algorithm 2 Scheme of Monte Carlo estimate of the spinfoam amplitude

for $k = 0.5, 1 \dots K$ **do** ▷ for each layer
    extract N_{mc} samples of the layer using RandomSample(k,$\mathcal{J}_k$)
    store them in memory

for $k = 0.5, 1 \dots K$ **do** ▷ for each layer
    load the bulk spins samples from memory
    for $n = 1 \dots N_{mc}$ **do** ▷ for each sample
        compute the partial amplitude $a(j_f)$
    sum the partial amplitudes and save the layer amplitude S_k
    compute the amplitude $A^{mc}_{\Delta}(k)$ summing the layer k to the previous ones
    store the amplitude

In calculating an EPRL spinfoam amplitude, the partial amplitude depends on the truncation parameter Δl. Therefore, we fix the truncation parameter once and for all and store the amplitude for every Δl.

4.4 Applications to the melonic self-energy and vertex renormalization diagrams

We test the effectiveness of the Monte Carlo framework described in Section 4.3 by computing four spinfoam amplitudes. We focus on the melonic self-energy amplitude and the vertex renormalization (or $5-1$ Pachner move) amplitude with the topological BF $SU(2)$ and the EPRL model. These diagrams are essential for studying the infrared divergences of spinfoam theories and their continuum limit.

The melonic self-energy diagram comprises two vertices, two boundary edges (one for each vertex), and four bulk ones connecting the two vertices, four boundary faces, and six bulk faces. We report in Figure 4.1 a schematic representation of the 2-complex, and we refer to Appendix D.4 for the detailed routing diagram.

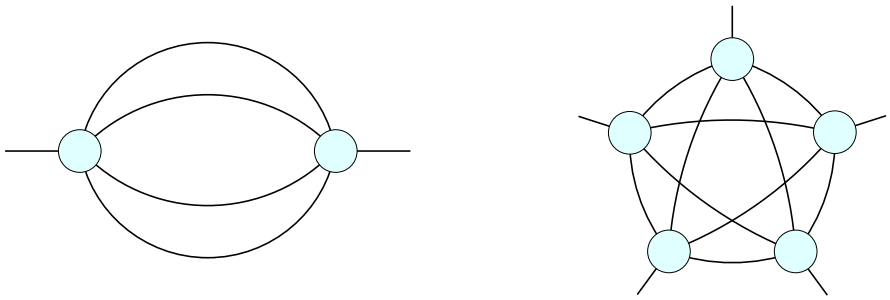


Figure 4.1: *Schematic representation of the 2-complexes of the self-energy spinfoam diagram (left) and the vertex renormalization spinfoam diagram (right).*

To simplify the numerical calculation, we consider very symmetric boundary data where the boundary spins j_b are all the same and all the boundary intertwiners i_b are all the same. The

spinfoam transition amplitude with a homogeneous cutoff on all the spins associated with bulk faces is

$$\begin{aligned} A_{se}\left(j_b,\ i_b;\ K\right) = \sum_{j_f=0}^{K}\left(\prod_{f=1}^{6}(2j_f+1)\right)\cdot\sum_{i_e}\left(\prod_{e=1}^{4}(2i_e+1)\right)\cdot \\ A_v\left(j_b,j_b,j_b,j_b,j_1,j_2,j_3,j_4,j_5,j_6;\ i_b,i_1,i_2,i_3,i_4\right)\cdot \\ A_v\left(j_b,j_b,j_b,j_b,j_1,j_2,j_3,j_4,j_5,j_6;\ i_b,i_4,i_3,i_2,i_1\right)\ . \end{aligned} \tag{4.10}$$

The amplitude (4.10) is the same for both spinfoam models that differ by the vertex amplitudes A_v: (4.2) for BF SU(2) and (4.3) for EPRL. The melonic self-energy diagram with the BF $SU(2)$ topological model can be evaluated analytically and numerically [71, 46, 72]. We know it is divergent. The divergence is due to redundant $SU(2)$ delta functions that indicate some residual gauge freedom in the path integral [144] and can be dealt with by gauge fixing appropriately. The same amplitude with the EPRL model has been studied analytically [102], with a hybrid calculation [46], and, recently, numerically [71, 72]. The amplitude is divergent, and there are strong indications that it diverges linearly in the cutoff. We are revisiting this amplitude as a control for the Monte Carlo technique we introduce. In fact, the self-energy has a relatively small number of internal faces, so the computation is still possible even without using Monte Carlo. Reproducing known results allows us to evaluate the choices of the framework.

The vertex renormalization diagram contains five vertices, five boundary edges (one for each vertex), and ten bulk ones connecting all couples of vertices, ten boundary faces, and ten bulk faces. We report in Figure 4.1 a schematic representation of the 2-complex, and we refer to Appendix D.4 for the detailed routing diagram.

Also in this diagram, we simplify the numerical calculation by taking symmetric boundary data with all equal boundary spins j_b and boundary intertwiners i_b. If we put a homogeneous cutoff K on the sums over the bulk spins, the amplitude reads

$$\begin{aligned} A_{vr}\left(j_b,\ i_b;\ K\right) = \sum_{j_f=0}^{K}\left(\prod_{f=1}^{10}(2j_f+1)\right)\cdot\sum_{i_e}\left(\prod_{e=1}^{15}(2i_e+1)\right)\cdot \\ A_v(j_b,j_b,j_b,j_b,j_1,j_2,j_3,j_4,j_5,j_6;\ i_b,i_4,i_{11},i_{12},i_2)\cdot \\ A_v(j_b,j_b,j_b,j_b,j_7,j_8,j_1,j_9,j_2,j_3;\ i_b,i_6,i_{13},i_{14},i_4)\cdot \\ A_v(j_b,j_b,j_b,j_b,j_{10},j_4,j_7,j_5,j_8,j_1;\ i_b,i_8,i_{15},i_{11},i_6)\cdot \\ A_v(j_b,j_b,j_b,j_b,j_6,j_9,j_{10},j_2,j_4,j_7;\ i_b,i_{10},i_{12},i_{14},i_{15})\cdot \\ A_v(j_b,j_b,j_b,j_b,j_3,j_5,j_6,j_8,j_9,j_{10};\ i_b,i_2,i_{13},i_{15},i_{10})\cdot \\ \{6j\}(j_b,j_3,i_1,j_5,j_6,i_2)\cdot\{6j\}(j_b,j_1,i_3,j_2,j_3,i_4)\cdot \\ \{6j\}(j_b,j_7,i_5,j_8,j_1,i_6)\cdot\{6j\}(j_b,j_{10},i_7,j_4,j_7,i_8)\cdot \\ \{6j\}(j_b,j_6,i_8,j_9,j_{10},i_{10})(-1)^{\chi}\ , \end{aligned} \tag{4.11}$$

where $\{6j\}$ are $SU(2)$ invariants that we define in Appendix D.1. The phase in (4.11) reduces to:

$$\chi = \sum_{k=1}^{10} j_k + \sum_{k=11}^{15} i_k\ . \tag{4.12}$$

The form of the amplitude is convoluted because we want to use the same intertwiner recoupling scheme in all the vertices. This is necessary to perform the numerical calculations efficiently, as `sl2cfoam-next` implements only a specific vertex amplitude (2.12). Again, the amplitude (4.11) is the same for both spinfoam models that differ by the vertex amplitudes. The vertex renormalization diagram can be analytically evaluated analytically with the BF $SU(2)$ topological model, explicitly integrating the group functions in the holonomy representation of the amplitude [46]. This amplitude was already studied with the Euclidean EPRL model in [123], finding a logarithmic divergence. A numerical calculation of the amplitude for the cutoff values greater than 4 is extremely challenging if we do not use Monte Carlo. The degree of divergence of the same diagram with the EPRL spinfoam model is entirely unknown. Any calculation with known techniques is too complicated. In Section 4.8 we study it using Monte Carlo. Computing this amplitude is a stress test for the Monte Carlo framework and a novel result for studying EPRL spinfoam infrared divergences.

In Figure 4.2, we show the number of bulk spin configurations j_f as a function of the cutoff K for the vertex renormalization diagram. We only consider spin configurations that satisfy triangular inequalities. It is evident that the number of configurations to be summed increases as a power law with the cutoff K. Hence the convenience of using Monte Carlo. A simple numerical fit for $K \in [5, 10]$ shows that the number of configurations qualitatively scales as $\sim 39.3 \cdot K^{8.5}$ for the vertex renormalization. For the self-energy diagram, in [72], the same fit for $K \in [5, 20]$ showed that the number of configurations scales as $\sim 17.1 \cdot K^{5.6}$.

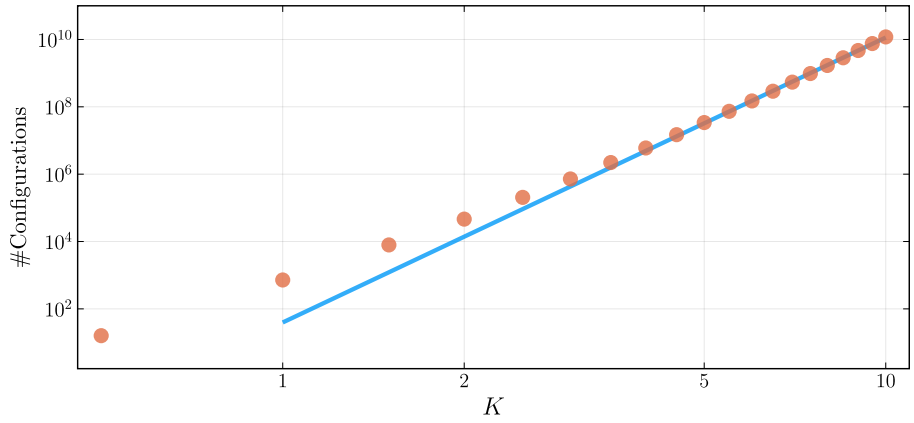


Figure 4.2: *Number of bulk spins configurations j_f as a function of the cutoff K of the vertex renormalization diagram.*

In all the amplitudes, to perform the numerical calculations with a modest amount of resources, we restrict the numerical calculation to the simplest non-trivial case of $j_b = \frac{1}{2}$ for both instances of boundary intertwiners $i_b = 0$ and $i_b = 1$. In the following sections, we will explicitly discuss the calculation's result only in the case of boundary intertwiner $i_b = 0$. However, we performed the same analyses also with boundary intertwiners $i_b = 1$. We find qualitatively identical results. Interested readers can find them in the detailed notebook in our public repository [142].

In the case of the EPRL model, we need to specify two more parameters to perform a numerical calculation. We fix the Immirzi parameter to $\gamma = 0.1$. We choose this value to partially

compare our results with the literature on the numerical evaluation of EPRL spinfoam amplitudes [71]. We also choose the truncation parameter $\Delta l = 10$ for similar reasons. This choice is also motivated by keeping the numerical task practical. The cost of resources increases rapidly with Δl, and literature [48, 9, 71] suggests that for maximal spins of order 10, the truncation $\Delta l = 10$ is a good compromise between costs and precision.

In the following sections, we use the same name for the amplitudes A_{se} and A_{vr} with both the BF $SU(2)$ and the EPRL model to keep the notation as clean as possible. The reader can uniquely identify which model the amplitude is computed with from the section. Finally, we use the term “exact amplitude" referring to (4.8) with a finite cutoff K computed without resorting to Monte Carlo methods.

4.5 The melonic amplitude in the topological theory

In this section, we use the Monte Carlo framework described in Section 4.3 on the melonic self-energy transition amplitude in the topological $SU(2)$ BF model. This calculation aims to fine-tune and validate our choices of Monte Carlo parameters.

First, we test if the algorithm we use to sample the layers is equivalent to a uniform discrete probability distribution over the amplitude layers. We list *all* the sets of bulk spins in the layer, and we map them in an interval of integers. Each element of the list is associated with its positions (we choose the order of the list arbitrarily but only once). We produce many samples using the Algorithm 1. We compute the samples' mean, variance, and skewness and check if they are compatible with the corresponding quantities of a discrete uniform probability distribution. We tested every layer of this amplitude and found excellent agreement. For brevity, we report the analysis with a sample of 100 000 configurations of the amplitude layer with $k = 10$ that contain $V_{\mathcal{J}_k} = 549\,406$ possible configurations. We report them in Table 4.1 and Figure 4.3. The probability distribution generated with Algorithm 1 is equivalent to a uniform discrete probability distribution.

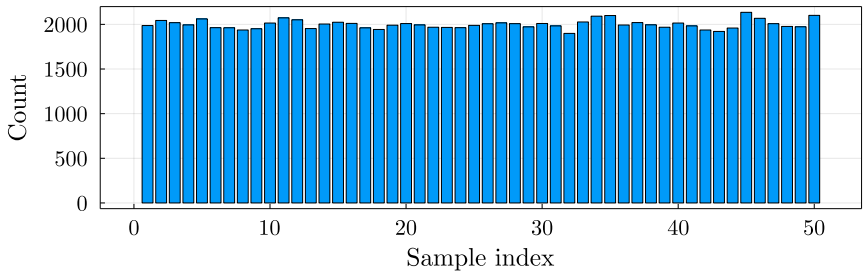


Figure 4.3: *Histogram of the sample with* 100 000 *configurations from the layer* $k = 10$ *with* 50 *bins. Each bin contains approximately* 2 000 *elements.*

Quantity	Expected	Sample	Difference (%)
Mean	274 703.5	274 864.7	0.05 %
Standard Deviation	1 587 371.04	1 585 998.51	0.09 %
Skewness	0	−0.00545	-

Table 4.1: *We compare the first three momenta of the sample with the corresponding exact quantities of a discrete uniform probability distribution. We find excellent agreement.*

We estimate the value and error of each amplitude layer with Monte Carlo repeating it $T = 20$ times and computing the mean and standard deviation. We choose the number of trials after a simple test. We fix the size of the Monte Carlo sampling to $N_{mc} = 1\,000$ to efficiently iterate and improve the analysis. We compute the average over $T = 10$, 20, and 50 trials. We repeat it 100 times to study the distribution of the average. The law of large numbers states that the distribution of the averages is normal with standard deviation given by the average standard deviation. We can visualize it using a box plot we report in Figure 4.4.

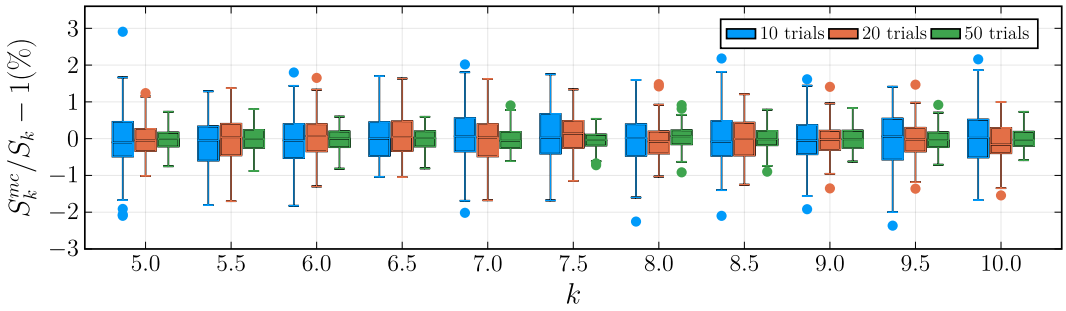


Figure 4.4: *Box plot of the average of the Monte Carlo evaluation of different amplitude layers. We consider the layers from* $k = 5$ *to* 10*. We repeat the estimate* 100 *times. We look at* 10 *trials (in blue),* 20 *trials (in orange), and* 50 *trials (in green). To ease the comparison, we plot the estimated value of the layer relative to the exact one.*

This qualitative analysis shows that the tails of the distribution with 10 trials are very long. The standard deviation with just 10 trials is not a reasonable estimate of the error of the Monte Carlo estimate of the amplitude's layer. This observation is independent of the layer. With 20 trials, the first and third quartiles are reduced to half, making it a better option. A similar observation is valid for 50 trials, albeit more resource intensive. We use 20 trials as a good compromise between precision and simplicity. Nevertheless, the error between the Monte Carlo estimate relative to the exact quantity is always a few percent with $N_{mc} = 1\,000$ and using 20 trials instead of 10 improve its estimate from 1% to 2%. The gain in the error estimate using 50 trials is marginal and does not justify the requirement of extra resources.

We study the Monte Carlo estimate of the amplitude as a function of the cutoff K for three different sample size choices of $N_{mc} = 1\,000$, $N_{mc} = 10\,000$, and $N_{mc} = 100\,000$. We average the calculation of $T = 20$ trials for each layer and sum them to get the amplitude. We compute the error on the amplitude from the standard deviation of each layer. We compare the relative error on the amplitude as a function of the cutoff for different sizes of Monte Carlo samples (see Figure 4.5).

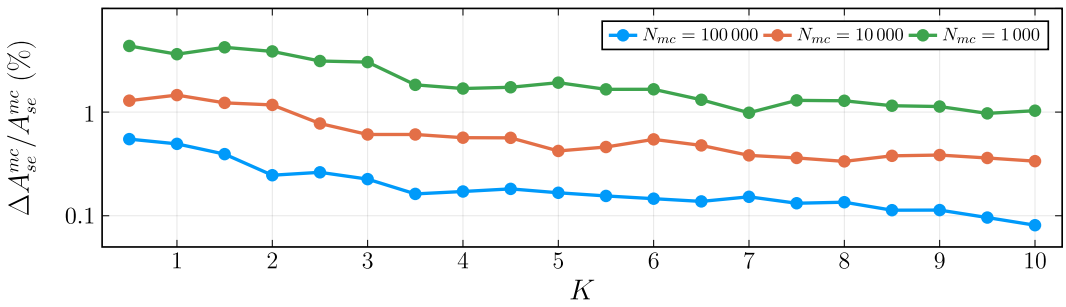


Figure 4.5: *Relative error of the $SU(2)$ BF melonic amplitude as a function of the cutoff computed with* 20 *trials. We compare different Monte Carlo sampling sizes* $N_{mc} = 1\,000$ *(green),* $N_{mc} = 10\,000$ *(orange), and* $N_{mc} = 100\,000$ *(blue) samples.*

For all three sample sizes, the relative error on the amplitude is smaller than 1%. As expected, the error decreases for larger values of N_{mc}. The relative error for $N_{mc} = 100\,000$ is smaller than 0.1%. We decide to use $N_{mc} = 100\,000$ for all the other calculations we present in this section. We also plot the estimated value of the amplitude with $N_{mc} = 100\,000$ relative to the exact value with the estimated errors. We plot in Figure 4.6 the Monte Carlo estimate of the amplitude in relation to the exact value, with error bars corresponding to the standard deviation.

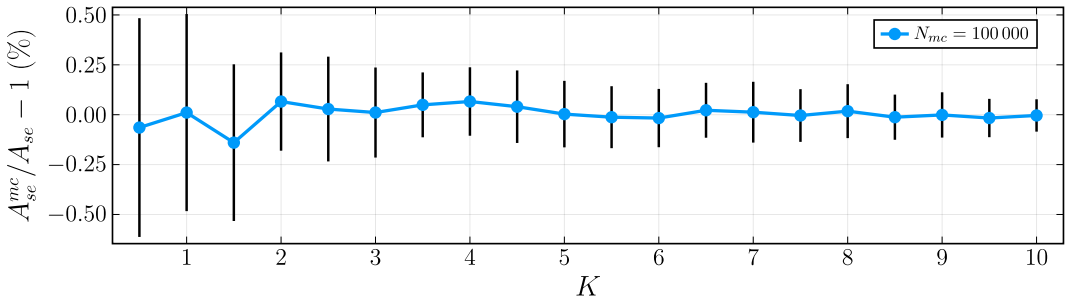


Figure 4.6: *Monte Carlo estimates of the $SU(2)$ BF melonic amplitude relative to the exact value with* $N_{mc} = 100\,000$ *and* $T = 20$ *trials in each layer.*

The exact value of the amplitude is compatible with the Monte Carlo estimate within the errors. At first sight, one could be confused by the trend of the errors decreasing with the cutoff. The observation that the relative error of the various layers is almost constant (as we can infer from Figure 4.4) can easily explain this. However, the contribution to the amplitude of the outer layers (with larger k) is bigger than the others. A quick back-of-the-envelope calculation shows that if we add two quantities a few orders of magnitudes apart but with the same relative error, the relative error on the sum is smaller than both.

We conclude our exploration by estimating the degree of divergence of this amplitude. The analytic calculation shows that the amplitude diverges with the cutoff as $\propto K^9$ at the leading order. Can we determine it numerically? We answer with a proof of concept analysis we use to validate the technique before applying it to more complex amplitudes where the analytic answer is unknown. We are not satisfied with a qualitative result. We could easily eyeball a

line on the logarithmic plot of the amplitude as a function of the cutoff. However, this approach is only helpful if we know the degree of divergence. We need to perform a fit to determine it numerically. We start with a model function with a polynomial form

$$A(K) = c_1 K^a + c_2 K^{a-1} \,. \tag{4.13}$$

We limit ourselves to the leading and subleading order terms. In general, the amplitude diverges as a polynomial with all the powers of the cutoff. Using it as a model to fit our data would undoubtedly lead to overfitting as we want to use a maximum cutoff of $K = 10$. We could compute this amplitude for larger values of the cutoff. However, the EPRL model's amplitude is too computationally demanding, and we must impose a small cutoff of $K = 10$. We use this limitation as an excuse to use a small cutoff with the topological model and anticipate some problems arising from this choice. We fit using only the last 10 data points available.

We perform a simple least squares fit using the `Julia` package `LsqFit`. We find the exponent $a = 8.81 \pm 11.99$ and coefficients $c_1 = 4.83 \pm 240.38$, $c_2 = 41.12 \pm 66.95$. Examining the uncertainties of the parameters, we conclude that the fit is clearly unreliable. Moreover, even if the fit value for the exponent a looks compatible with the exact value $a = 9$, we could not affirm it without knowing it in advance.

What is happening? The covariance between the coefficients c_1 and c_2 is huge. The fit procedure with a cutoff of order 10 cannot distinguish between the contributions from the leading and sub-leading orders (for example, if the exact coefficients are $c_1/c_2 \approx 10$).

A simple solution to this impasse would be to extend the fit to larger values of K. In this way, the contribution of the leading order would dominate the sub-leading one. Since we are limited by a maximum cutoff $K = 10$, we have to find a different solution. Alternatively, we can diminish the degree of divergence of the amplitude by changing the face amplitude introducing a tunable parameter μ

$$A_f(j_f) = (2j_f + 1) \to (2j_f + 1)^\mu \tag{4.14}$$

The case $\mu = 1$ corresponds to the standard case, but if we set $\mu < 1$ we lower the divergence of the amplitude. In general, the amplitude will diverge as $A_{se} \propto K^{6\mu+p}$ where p is a number we have to determine, and 6μ is the contribution coming from the six unbounded sums over the bulk spins. We pretend we do not know that for $\mu = 1$ the amplitude diverges as $A_{se} \propto K^9$ and therefore $p = 3$. And we try to determine $p = a - 6\mu$ fitting the amplitude with the same model (4.13) for various values of $\mu = 1/6$, $\mu = 0$ and $\mu = -1/6$. We take this opportunity to check if the Monte Carlo estimate of the amplitude is as good as in the case $\mu = 1$. We compare the Monte Carlo estimate of the amplitudes relative to their exact values with different μ. All the relative error bars, computed over $T = 20$ realizations as we did before, are within 0.1%, confirming that the Monte Carlo estimate is very accurate. We summarize the results in Figure 4.7.

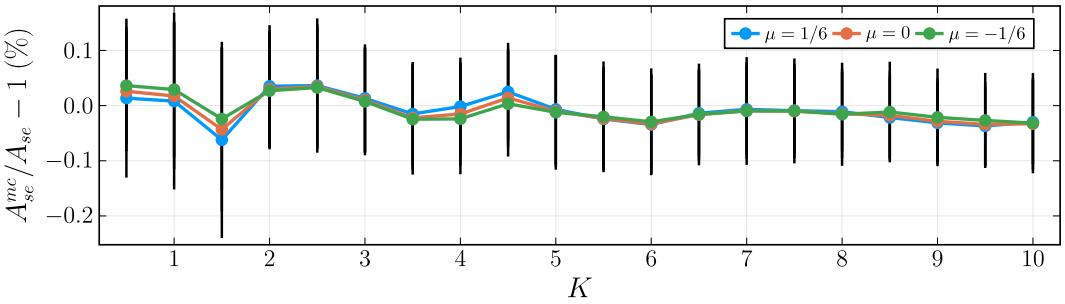


Figure 4.7: *Monte Carlo estimates of the $SU(2)$ BF melonic amplitude with $N_{mc} = 100\,000$ and $T = 20$ trials in each layer. We compare different values of the parameter μ in the face amplitude* (4.14).

We fit the amplitude with the model (4.13). The interpretation of the result, in this case, is more straightforward. We report the fitted coefficients in Table 4.2.

	a	c_1	c_2
$\mu = 1/6$	4.15 ± 0.34	0.91 ± 1.24	6.38 ± 0.17
$\mu = 0$	2.97 ± 0.04	2.01 ± 0.03	3.50 ± 0.05
$\mu = -1/6$	1.94 ± 0.02	2.83 ± 0.02	0.39 ± 0.05

Table 4.2: *Values of the coefficients of the model* (4.13) *obtained fitting the amplitude with different values of μ.*

A few comments are in order. First, all the fits indicate clearly that $p = a - 6\mu = 3$. Second, notice that we are not worried that a is not always compatible with the nearest integer value. This is an artifact of using just the leading order and next to the leading order of the polynomial in (4.13). We determined the degree of divergence of the amplitude as $A_{se} \propto K^{6\mu+3}$.

4.6 The vertex renormalization amplitude in the topological theory

Exact numerical calculations of spinfoam amplitudes with many bulk faces are accessible only for simple models, but become infeasible when the number of faces is too large. We showcase the problem by looking at the vertex renormalization or 5-1 Pachner move amplitude with the topological $SU(2)$ BF spinfoam theory. The issue with this computation is not the time we need to compute each term of the sums over the spins of the ten internal faces but the sheer amount of terms of these sums. With a cutoff $K = 10$ on the sums, we need to compute 30 788 382 715 terms in total, 11 892 969 195 of which belongs to last layer of the amplitude. They are almost six orders of magnitude more than in the melonic diagram case. This is where using the Monte Carlo framework to perform the sums is necessary.

Motivated by the analysis of the melonic diagram, we average the Monte Carlo calculation of each layer over $T = 20$ trials. We sum the layers' average to obtain the amplitude value for a given cutoff. We compute the amplitude variance by summing each layer's variance. We

consider the standard deviation as the error of the amplitude. We perform the calculation with three different choices of sample sizes $N_{mc} = 1\,000$, $N_{mc} = 10\,000$, and $N_{mc} = 100\,000$.

In this case, we cannot compare with the exact value of the amplitude to evaluate the Monte Carlo technique. The exact value is not computable for cutoff $K = 10$. It is exactly the reason we resort to Monte Carlo methods. We compare the relative error on the amplitude for the three sample size choices. We summarize the result of this analysis in the plot of Figure 4.8.

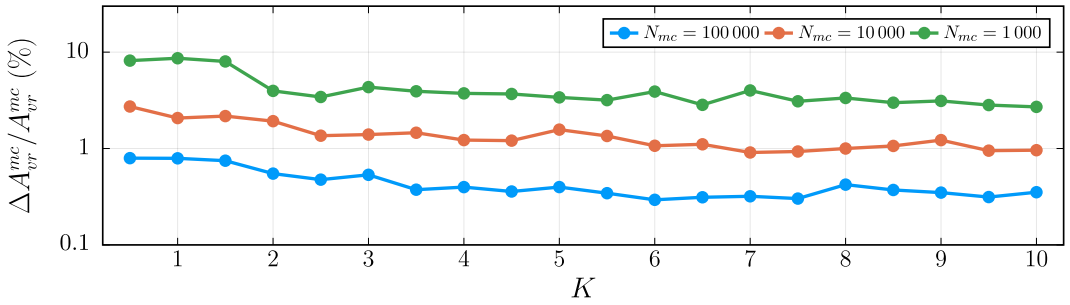


Figure 4.8: *Relative error on the Monte Carlo estimate of the $SU(2)$ BF vertex renormalization amplitude over 20 trials. We compare different Monte Carlo sampling sizes $N_{mc} = 1\,000$ (green), $N_{mc} = 10\,000$ (orange), and $N_{mc} = 100\,000$ (blue) samples.*

We see that $N_{mc} = 1\,000$ produces an estimate with a significant relative error between 8% and 3%. For $N_{mc} = 100\,000$ we find very modest relative errors between 0.8% and 0.3%. The increase in the relative errors compared to the melonic diagram case is expected. Each layer of this amplitude contains a few orders of magnitude more elements than the corresponding layer in the melonic amplitude.

We fit the amplitude computed with $N_{mc} = 100\,000$ samples averaged over $T = 20$ trials as a function of the cutoff K using the model (4.13). We employ only the amplitude value as a function of the cutoff between 5 and 10 since we expect the matching of the functional form (4.13) to be, at best asymptotic. The degree of divergence of this amplitude can be estimated analytically, finding $A_{vr} \propto K^{12}$ at the leading order.

With a simple least squares fit, we find an unreliable result with an exponent $a = 11.20 \pm 0.69$ and coefficients $c_1 = 1.91 \pm 4.39$, and $c_2 = 1.70 \pm 10.82$. The situation is analogous to the case of the melon amplitude. To determine numerically the degree of divergence of this amplitude with a maximum cutoff of $K = 10$ we modify the face amplitude as in (4.14) such that the amplitude diverge as $A_{vr} \propto K^{10\mu+p}$ with p to determine. We want to keep the exponent $10\mu + p$ as low as possible. Therefore, we pick three values of the weight $\mu = 0$, $\mu = 1/10$, and $\mu = 1/5$. We find

	a	c_1	c_2
$\mu = 0$	1.96 ± 0.02	0.18 ± 0.01	-0.05 ± 0.02
$\mu = 1/10$	2.89 ± 0.02	0.15 ± 0.01	0.07 ± 0.02
$\mu = 1/5$	4.00 ± 0.04	0.08 ± 0.01	0.26 ± 0.02

Table 4.3: *Values of the coefficients of the model* (4.13) *obtained fitting the amplitude A_{vr} with different values of μ.*

All three results are compatible with the analytic value of $p = 2$ resulting in the amplitude diverging as $A_{vr} \propto K^{10\mu+2}$.

4.7 The melonic amplitude in the EPRL theory

In Sections 4.5, we computed the melonic self-energy spinfoam amplitude with the topological BF $SU(2)$ model using Monte Carlo. We obtained a remarkably accurate amplitude approximation, employing only a fraction of the computational resources. Is the Monte Carlo technique applicable to spinfoam amplitudes with the EPRL model, and is it equally successful?

Before discussing the calculation details, we must disentangle two different overlapping approximations. One is due to the Monte Carlo sampling procedure, while the other is due to working with a finite truncation parameter Δl. In calculating the melonic amplitude, we can access the public data from [71] that employs an extensive truncation $\Delta l = 20$. We borrow that data to perform a detailed study of the truncation approximation independently from the Monte Carlo one.

We mitigate the dependence from a specific choice of truncation Δl using an extrapolation technique (see Appendix D.3). This idea was first introduced in [71], further formalized in [9], and additionally utilized in [133]. Our understanding of the extrapolation technique applied to EPRL spinfoam amplitudes with finite truncation has improved considerably. In this section, we revisit it in a new light. We explicitly show that its principal hypothesis is satisfied and compare possible alternatives.

The amplitude $A_{se}(K, \Delta l)$ at fixed cutoff K is a sequence in the truncation parameter Δl. Since the EPRL vertex amplitude is well-defined, the limit of infinite truncation is finite, and we can approximate it using the Aitken delta squared method.

$$A^{(ex)}_{se}(K, \Delta l) = \frac{A_{se}(K, \Delta l)A_{se}(K, \Delta l - 2) - A_{se}(K, \Delta l - 1)^2}{A_{se}(K, \Delta l) - 2A_{se}(K, \Delta l - 1) + A_{se}(K, \Delta l - 2)} \tag{4.15}$$

The sequence $A^{(ex)}_{se}(K, \Delta l)$ converges to $A_{se}(K)$ faster than $A_{se}(K, \Delta l)$. Therefore approximating the limit with the truncation of the sequence $A_{se}(K) \approx A^{(ex)}_{se}(K) \equiv A^{(ex)}_{se}(K, \Delta l)$ is, in general, a better approximation than using the truncation of $A_{se}(K, \Delta l)$. The extrapolation is effective if the rate of convergence of the amplitude is at least linear in Δl, as discussed in Appendix D.3. For this purpose, we study the ratio

$$\lambda_{A_{se}(K)}(\Delta l) = \frac{A_{se}(K, \Delta l) - A_{se}(K, \Delta l - 1)}{A_{se}(K, \Delta l - 1) - A_{se}(K, \Delta l - 2)} \,. \tag{4.16}$$

If the limit of the ratio (4.16) for infinite truncation is smaller than 1, the convergence of $A_{se}(K, \Delta l)$ is linear. Proving numerically the existence of the limit is challenging. Therefore, we settle with some numerical evidence for linear convergence. Moreover, truncating the sequence of extrapolations $A^{(ex)}_{se}(K, \Delta l)$ to a finite Δl to approximate its limit is reliable only if the ratio approached (at least approximately) a horizontal asymptote. Part of this analysis has already been performed in [71]. Here we re-propose it in light of our improved understanding. For all half-integers $K \leq 10$, the ratio (4.16) approaches a horizontal asymptote smaller than 1. This behavior is evident for any $8 \leq \Delta l \leq 20$ and legitimizes the extrapolation of the amplitude

(4.15). The ratio approaches the asymptote from below. Therefore, we expect the extrapolations from larger truncation to increase. We summarize the analysis for some cutoff values in Figure 4.9.

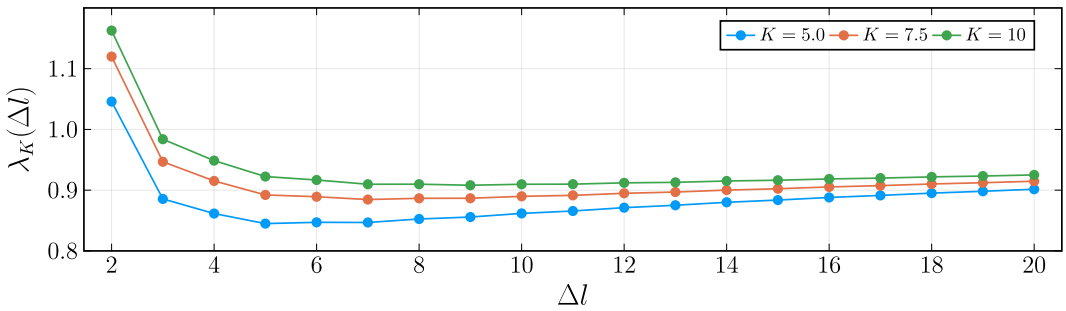


Figure 4.9: *Plot of the ratio* (4.16) *for various values of the cutoff* $K = 10$, $K = 7.5$, $K = 5$. *A horizontal asymptote is evident even for modest values of the truncation.*

How effective is the extrapolation (4.15)? We answer this question by comparing the extrapolation obtained from $\Delta l = 10$ and $\Delta l = 20$. The last is usually inaccessible due to its exceptional computational cost, while the former is less precise but cheaper to compute. The value of the amplitude $A_{se}(K, \Delta l)$ changes substantially when we increase the truncation. For example, at cutoff $K = 10$, the amplitude with $\Delta l = 20$ is 40% larger than the one with $\Delta l = 10$. This was expected since the spins entering the calculation at $K = 10$ are of order 20, and the amplitude truncated at $\Delta l = 10$ cannot approximate the real amplitude value well. However, we find a milder difference between the extrapolation (4.15) with different truncations. With the same cutoff, the difference between the extrapolation done with truncation $\Delta l = 10$ and $\Delta l = 20$ is just 6%. In both cases, the extrapolations approximate the amplitude better than any truncated amplitude we have access to. We summarize the results in Figure 4.10.

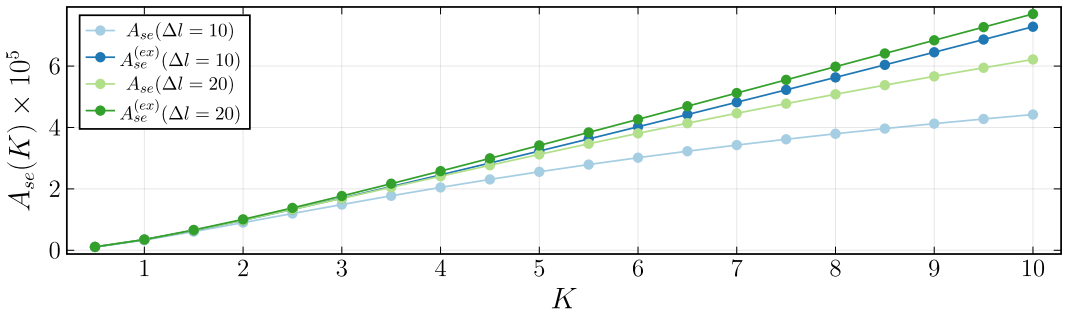


Figure 4.10: *Comparison between the amplitude extrapolations and the truncated values* $A_{se}(K, \Delta l)$ *with* $\Delta l = 10$ *and* $\Delta l = 20$ *as a function of the cutoff. For larger cutoff values, the effect of truncation is critical. The extrapolation successfully mitigates this dependence.*

It is natural to question if extrapolating the amplitude sequence (4.15) is the only way to proceed. We explored multiple alternatives (vertex-by-vertex, bulk amplitude, and layer-by-layer), and they all turned out to perform worst. We review in detail one of them. The

amplitude is a sum limited to a maximum K of contributions of layers (4.8) we compute with a fixed truncation Δl

$$A_{se}(K, \Delta l) = \sum_{k=0}^{K} S_k(\Delta l) \,. \tag{4.17}$$

Each layer contribution $S_k(\Delta l)$ is a convergent sequence in the truncation parameter. We can use Aitken extrapolation on each layer contribution and obtain a sequence of amplitudes summing them

$$A_{se}^{(ex,S)}(K, \Delta l) = \sum_{k=0}^{K} S_k^{(ex)}(\Delta l) \qquad \text{with} \qquad S_k^{(ex)}(\Delta l) = \frac{S_k(\Delta l) S_k(\Delta l - 2) - S_k(\Delta l - 1)^2}{S_k(\Delta l) - 2S_k(\Delta l - 1) + S_k(\Delta l - 2)} \,. \tag{4.18}$$

We approximate the limit of the sequence of amplitude truncating the accelerated convergence sequence $A_{se}^{(ex,S)}(K) \approx A_{se}^{(ex,S)}(K, \Delta l)$ (4.18). Since the amplitude (4.17) is a finite sum over layers, the sequence obtained extrapolating layer-by-layer (4.18) has the same limit as the sequence (4.15) and both, of course, converge to the value of the amplitude $A_{se}(K)$ without any truncation. In practice, we do not have access to an arbitrarily large truncation but to a relatively small one, and we want to approximate the limit truncating the extrapolated sequences. Which one approximates the amplitude better is an open question that we can answer by exploring different possibilities.

First, we verify that every layer amplitude is at least linearly convergent in the truncation parameter by studying the ratio

$$\lambda_{S_k}(\Delta l) = \frac{S_k(\Delta l) - S_k(\Delta l - 1)}{S_k(\Delta l - 1) - S_k(\Delta l - 2)} \,. \tag{4.19}$$

We show some explicit examples in Figure 4.11. All the layer amplitudes are compatible with the linear convergence hypothesis. Differently from (4.16) the ratio (4.19) approach a horizontal asymptote from above. As a consequence, the extrapolation of the layer amplitudes decreases in value for increasing truncation. In particular, the last layer $k = 10$ for $\Delta l = 10$ is still far from the horizontal asymptote. For this reason, we expect the extrapolation to change significantly if we increase the truncation from $\Delta l = 10$ to, for example, $\Delta l = 15$ or 20 and approximate. This is not the case for lower levels $k \leq 5$.

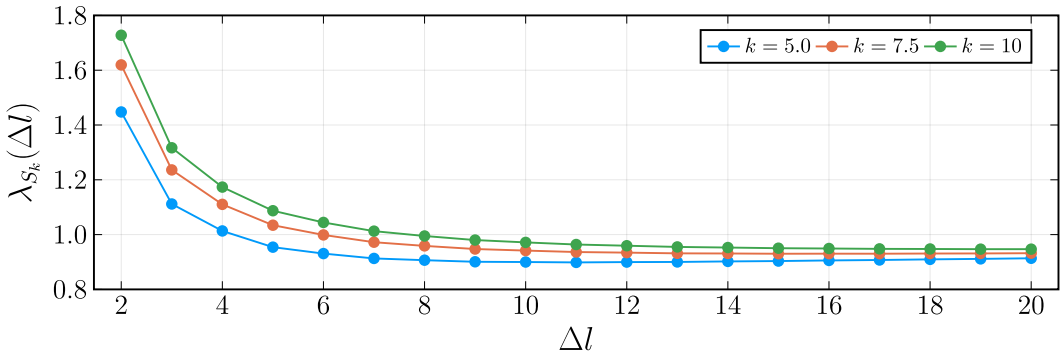


Figure 4.11: *Plot of the ratio* (4.19) *for various values* $k = 10$, $k = 7.5$, $k = 5$. *A horizontal asymptote is evident but is reached for different truncation values.*

We extrapolate all the layer amplitudes and sum them. We truncate the sequence (4.18) to approximate its limit. It is useful to compare the result with the extrapolation of the whole amplitude (4.15) using different truncation parameters. We summarize our findings in Figure 4.12.

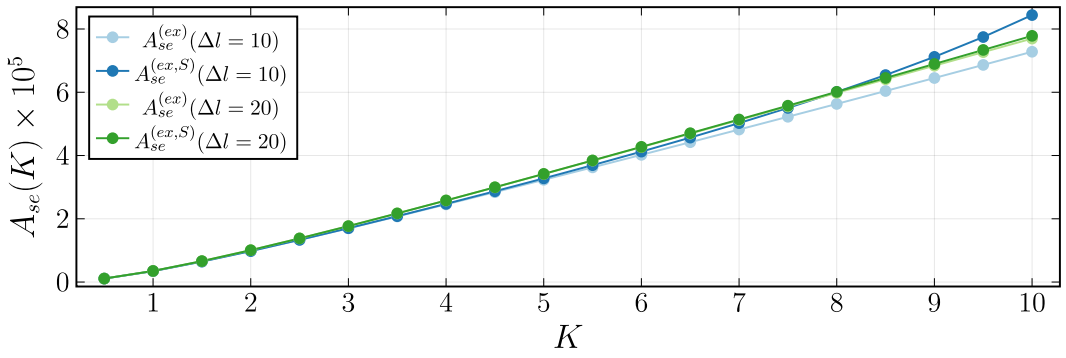


Figure 4.12: *Comparison of the melonic self-energy amplitude obtained extrapolating the amplitude layer-by-layer or the whole amplitude.*

For large cutoff, the extrapolation (4.18) with truncation $\Delta l = 10$ is much larger than its counterpart (4.15). At cutoff $K = 10$, the layer-by-layer extrapolation (4.18) is 16% larger than whole amplitude extrapolation (4.15). The difference reduces drastically if we repeat both extrapolations with a larger truncation $\Delta l = 20$. At cutoff $K = 10$ the two differ only by 1%. In particular, the value of the extrapolation (4.18) decreases substantially. This agrees with what we observed studying the ratio (4.19). With low truncation, the layer-by-layer extrapolation results in a poor amplitude approximation. In contrast, the two extrapolation schemes almost coincide for larger truncations.

In the following, we will study the Monte Carlo approximation of the amplitude and limit ourselves to a small truncation $\Delta l = 10$ to conserve computational resources. We will use only the whole amplitude extrapolation scheme (4.15) since it is the most accurate within this setting.

Next, we evaluate how effective Monte Carlo techniques are if applied to the bulk spin summations in the EPRL model using the melonic amplitude as a testing ground. Following the cost-benefit analysis of the first part of this section, we set the truncation to $\Delta l = 10$. We use Monte Carlo to estimate the contribution to the amplitude of each layer averaging over $T = 20$ trials. The amplitude is given by the sum of the averages of the layers and its error by the square root of the total variance. We perform the calculation with three different choices of Monte Carlo sample sizes $N_{mc} = 1\,000$, $N_{mc} = 10\,000$, and $N_{mc} = 100\,000$. As displayed in Figure 4.13 the relative error on the amplitude is more or less stable at 1% for the small sample size and 0.1% for the large one.

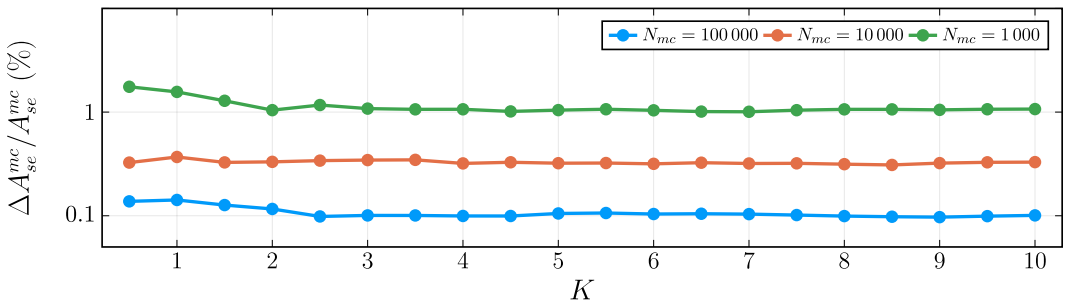


Figure 4.13: *Relative error of the EPRL melonic amplitude as a function of the cutoff computed with* 20 *trials and truncation fixed to* $\Delta l = 10$. *We compare different Monte Carlo sampling sizes* $N_{mc} = 1\,000$ *(green),* $N_{mc} = 10\,000$ *(orange), and* $N_{mc} = 100\,000$ *(blue) samples.*

The error is compatible with the analog error computed with the topological theory. We could have expected it as the distribution of the layers' amplitudes value is relatively flat. The dominant factor in the error is the ratio between the number of configurations in the layer and the Monte Carlo sample size, which is model-independent. At fixed truncation $\Delta l = 10$, we can also compare the amplitude computed using Monte Carlo with the exact one. The amplitude computed using Monte Carlo is compatible with the exact value within the error. We summarize the comparison in the plot in Figure 4.14.

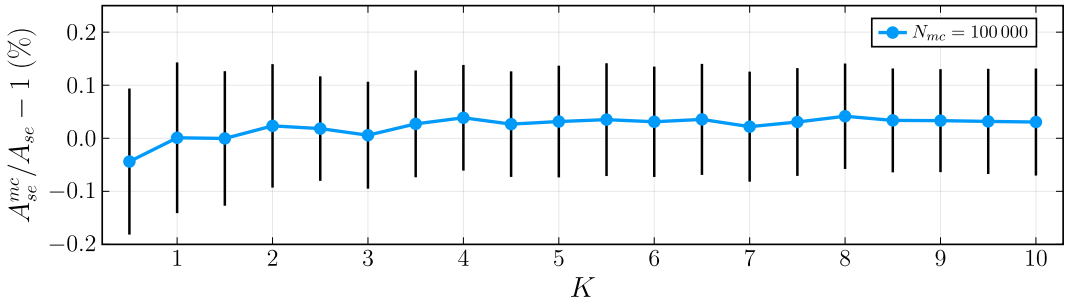


Figure 4.14: *Monte Carlo estimate of the EPRL melonic amplitude with finite truncation* $\Delta l = 10$ *relative to the exact value with* $N_{mc} = 100\,000$ *and* $T = 20$ *trials in each layer.*

How well the extrapolation technique (4.15) is compatible with the Monte Carlo sum over the bulk spins? Instead of averaging over $T = 20$ trials, we sum the layer amplitudes of each realization to obtain 20 different realization of the amplitude at fixed truncation $\Delta l = 10$. This is possible since each Monte Carlo estimate of each layer amplitude is independent. We extrapolate the amplitude (4.15) for each trial and approximate it averaging over the trials and considering as error its standard deviation.

We compare the extrapolation of the amplitude computed with Monte Carlo with the one calculated without that approximation. We find a 0.1% average error due to Monte Carlo for all cutoff values (see Figure 4.15). This is compatible with the finite truncation case with $\Delta l = 10$. However, this is just an error due to the Monte Carlo approximation. We expect it to be firmly subdominant with respect to the error due to the presence of the truncation despite the extrapolation.

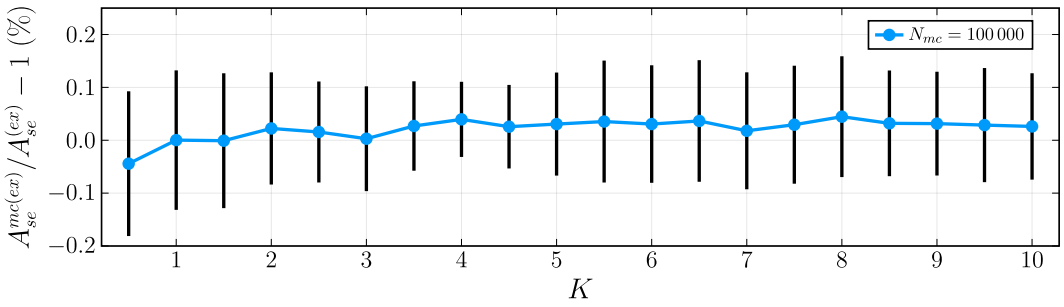


Figure 4.15: *Monte Carlo estimates of the EPRL melonic amplitude extrapolated from a finite truncation* $\Delta l = 10$ *relative to the exact value with* $N_{mc} = 100\,000$ *and* $T = 20$ *trials in each layer.*

Finally, we determine the degree of divergence of the amplitude performing a numerical fit. The same calculation was already done in [71] with an exact amplitude computation. We showed that using Monte Carlo allows us to estimate the amplitude with a 0.1% error compared to the exact value (at fixed truncation). Therefore is no surprise that we find a good fit with

$$A(K) = c_1 K^a + c_2 \tag{4.20}$$

with $a = 1.091 \pm 0.005$, $c_1 = (6.186 \pm 0.093) \cdot 10^{-6}$, and $c_2 = (-3.476 \pm 0.253) \cdot 10^{-6}$ that coincide with the result of [71]. We changed the model of the fit from the topological models, as (4.13) gives unreliable results. If we generalize the face amplitude (4.14) introducing a weight μ, we conclude that the scaling of the amplitude is compatible with $A(K) \propto K^{6\mu+p}$ with $p \approx -5$.

For completeness, we could look at the melonic divergence with different weight μ values as we did for the topological $SU(2)$ model. However, the result we obtain is unreliable. A more solid calculation requires a larger truncation and, consequently, way more computational resources that we currently do not have access to. We will comment on these issues in more detail in the next section. We leave this interesting consistency check to future work.

4.8 The vertex renormalization amplitude in the EPRL theory

Computing the degree of divergence of the EPRL vertex renormalization amplitude (4.11) has never been attempted. The theory is too complex to do it numerically or analytically. We can use Monte Carlo to calculate this amplitude for the first time. We build upon the experience accumulated in the previous sections and our choices and approximations. For this reason, we use a truncation parameter $\Delta l = 10$, and $N_{mc} = 100\,000$ Monte Carlo samples. The calculation of this amplitude required $\sim$ 400 CPU hours, which is a minimal fraction of what would be required without Monte Carlo. We compute the Monte Carlo error as the standard deviation of the amplitude over $T = 20$ relative to its average. We find a very stable error of approximately 0.9% for all values of the cutoff as summarized in Figure 4.16

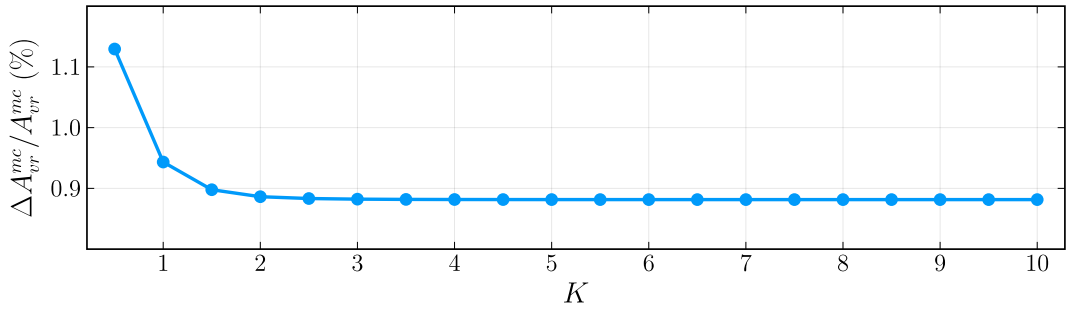


Figure 4.16: *Relative error of the EPRL vertex renormalization amplitude as a function of the cutoff, computed with* 20 *trials and truncation fixed to* $\Delta l = 10$*. We use* $N_{mc} = 100\,000$ *Monte Carlo samples.*

The relative error of the amplitude is curiously constant for increasing cutoff K. While this behavior could seem odd initially, it has a straightforward explanation. As we discuss below, the amplitude seems convergent. The amplitude layers decrease very fast. The contribution to the relative error of larger layers is irrelevant.

The amplitude's statistical fluctuations are slightly bigger than in the EPRL self-energy amplitude. However, they are compatible with the errors of the topological model. It is not surprising since the ratio between the number of samples and the cardinality of the set we are summing over dominates the error of a Monte Carlo calculation.

We reduce the dependence of the amplitude from the truncation using the extrapolation (4.15) as discussed in Section 4.7. We show the value of the extrapolated amplitude as a function of the cutoff in Figure 4.17. The amplitude is essentially constant for cutoff $K > 2$. A power law fit is inadequate to capture the functional scaling of the amplitude. Therefore we opt for a model capturing the constant behavior plus a correction.

$$A(K) = c_1 + c_2/K \,. \tag{4.21}$$

Fitting the amplitude result in $c_1 = 0.765 \pm 2.667 \times 10^{-5}$ and $c_2 = -0.0006 \pm 0.0002$. We should take these values with a grain of salt as they depend strongly on the model we decide to use.

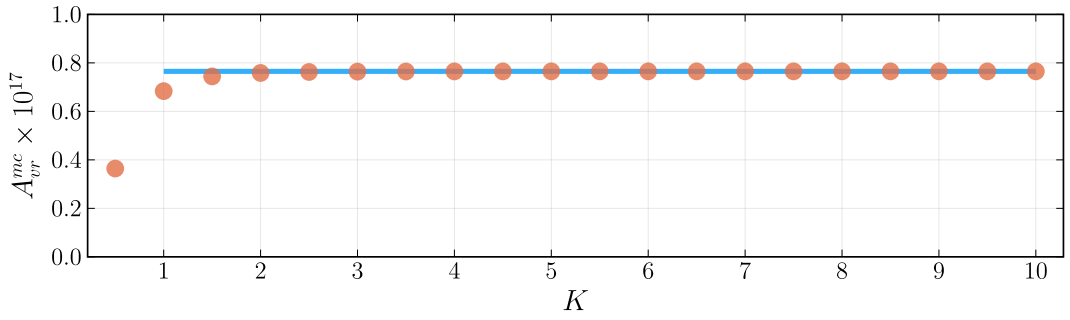


Figure 4.17: *Monte Carlo estimate of the EPRL vertex renormalization amplitude as a function of the cutoff* K*. We use* $\Delta l = 10$*,* $N_{mc} = 100\,000$ *and* $T = 20$ *trials. We plot the extrapolated amplitude and the fit using the model* (4.21)*.*

We are tempted to enhance the divergence of the amplitude by modifying the face amplitude (4.14) introducing the weight μ. We observe that increasing μ we need to increase Δl.

Otherwise, the extrapolation technique fails in estimating the amplitude well.

We tried different face amplitudes weights and studied the amplitude ratios (4.16). We show it in Figure 4.18. While for standard face amplitude $\mu = 1$ and $\mu = 1.4$, the ratio reaches a horizontal asymptote smaller than 1 very soon, it is not the case for $\mu = 1.8$, and $\mu = 2$. We can see how for these two cases, the ratio is still decreasing and greater than 1 at truncation $\Delta l = 10$. Therefore, to obtain a reliable extrapolation we would need a larger truncation, not accessible with the computational resources at our disposal.

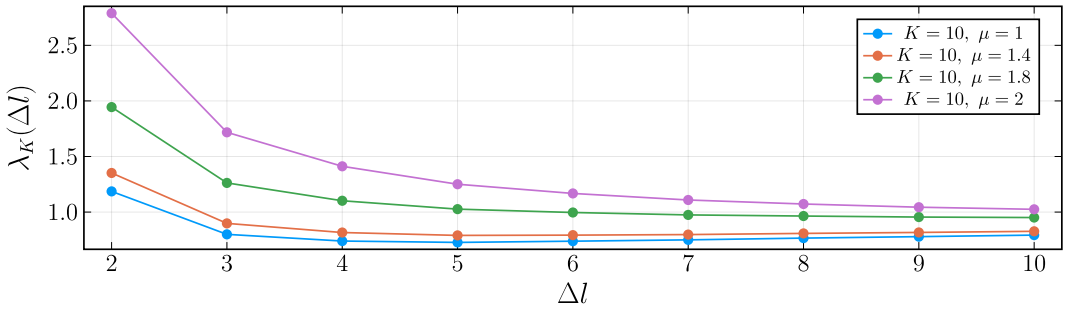


Figure 4.18: *Plot of the ratio* (4.16) *for increasing values of* μ *at cutoff* $K = 10$*. For* $\mu = 1.8$ *and* $\mu = 2$ *the ratio has not reached a horizontal asymptote smaller than* 1 *at* $\Delta l = 10$*. Therefore the extrapolation* (4.15) *cannot be applied.*

We conclude that for this transition amplitude, the extrapolation scheme (4.15) is sensible to the weight on the face amplitudes. The slower convergence of the amplitude sequence is also evident from the plot of the amplitude for different truncations as well displayed in Figure 4.19.

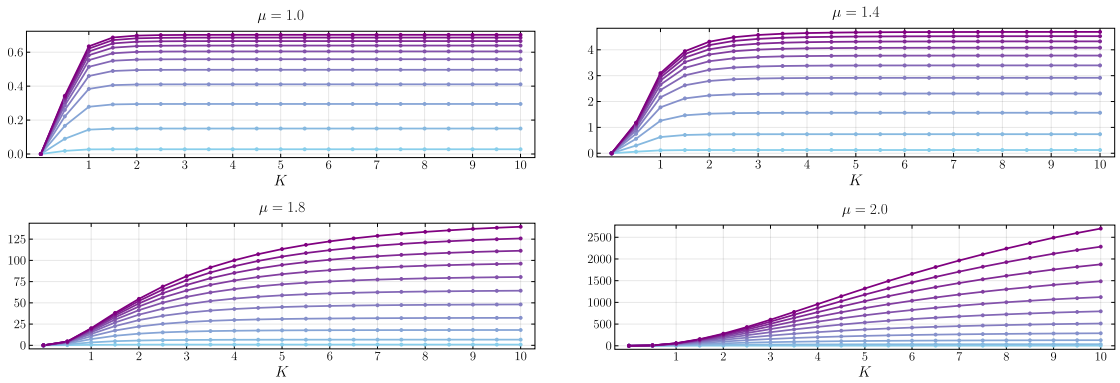


Figure 4.19: *Plots of vertex renormalization EPRL amplitude* $A_{vr}^{mc} \times 10^{17}$ *for increasing values of the weight factor* μ*. In each panel, we report all the curves obtained for increasing values of the truncation parameter* Δl*. The bottom curve (azure) corresponds to* $\Delta l = 0$ *while* $\Delta l = 10$ *is the top one (purple).*

For completeness, we report the fit of the convergent amplitude with $\mu = 1.4$ with the model (4.21). We find $c_1 = 5.535 \pm 0.011$ and $c_2 = -0.464 \pm 0.074$. Due to the invalidity of the extrapolation, we cannot perform a fit in the other two cases. Consequently, we cannot estimate p in the divergence of the amplitude $A_{vr} \propto K^{10\mu+p}$. Nevertheless, our numerical analysis shows strong indications that the infrared bubble of the EPRL theory is convergent.

4.9 Conclusion and Discussion

The `sl2cfoam-next` library allows fast and reliable calculations of EPRL spinfoam transition amplitudes. While it is optimized to compute vertex amplitudes, calculating a spinfoam amplitude with many vertices and internal faces still presents a huge technical obstacle. To sum over the bulk degrees of freedom, we have to compute an enormous number of components that scale exponentially with the number of internal faces.

We apply Monte Carlo to the spinfoam bulk summations and show that it is a very promising strategy to overcome this obstacle. The complexity of the calculation depends on the number of Monte Carlo samples N_{mc} we can freely choose. Of course, the result's precision depends on N_{mc} and how we choose the probability distribution of the Monte Carlo sampling. We decide to use a uniform probability distribution. We acknowledge it is not the optimal choice as it equally weights all the bulk spins configurations. However, it is efficient and allows us to parallelize the sampling algorithm for a single amplitude across multiple threads. Moreover, we can use it democratically with any amplitude. Alternatively, we could abandon uniform sampling in favor of Markov Chain Monte Carlo (MCMC) methods. This would result in faster convergence, but on the other hand, the sampling algorithm for calculating a single amplitude would no longer be parallelizable. We leave the study and implementation of MCMC to bulk degrees of freedom for future works. Finally, in this paper, we used intertwiners as boundary states. In other cases, one could attempt to take advantage of the state's properties to perform importance sampling Monte Carlo. For example, in the case of extrinsic boundary states, it would make sense to tailor the probability distribution sampling from a normal distribution. However, one has to deal with the well-known sign problem with highly oscillatory distributions. We leave the analysis of different boundary states for future works.

We evaluate the proposed strategy and discuss its choices for computing the melonic self-energy and the vertex renormalization spinfoam amplitudes with the $SU(2)$ BF and the EPRL theory. These amplitudes with many vertices and internal faces provide a good test. In particular, the topological theory is convenient as partial analytical calculations are possible and help us evaluate the performance of our method. The calculations of the amplitudes with Monte Carlo are surprisingly effective already with a modest number of Monte Carlo samples $N_{mc} = 100\,000$. Remember that the vertex renormalization amplitude contains $\approx 10^{10}$ possible (non trivially vanishing) amplitudes to appreciate this result. Obtaining a good approximation computing just order $\approx 10^5$ of them is remarkable. We find minor uncertainties of 0.1% and 0.9% for the EPRL self-energy and the vertex renormalization amplitude, respectively. Similar results hold for the $SU(2)$ BF model suggesting that the dominant factor in the error is the ratio between the cardinality of the space of all the possible spin configurations and the (square root of the) Monte Carlo samples. The calculations in the EPRL theory are carried out with a finite truncation $\Delta l = 10$. We alleviate the dependence of the result using extrapolation techniques to accelerate the convergence of the amplitude. We explore different extrapolation schemes. Extrapolating the full amplitude at a finite cutoff is the most convenient option. We also formalize the regime of validity of the extrapolation scheme and develop a test to verify if the amplitude falls into it.

At the same time, analyzing the divergence of these amplitudes for the EPRL theory is an essential step toward understanding the theory's continuum limit. While the self-energy ampli-

tude is already studied in the literature [71, 46, 102], for the vertex renormalization amplitude, we know only a loose upper bound [46]. The amplitude was too complex to try any numerical or analytical calculation. The estimate we provide in this paper is a complete and important novelty. Performing the sums over the bulk degrees of freedom with statistical methods is enough to confirm the linear divergence of the melonic self-energy amplitude.

The numerical evaluation of the EPRL vertex renormalization spinfoam amplitude provides a convincing argument to claim its convergence. However, it is a shocking result as it contradicts any intuition from the analytical calculations of the topological models where the amplitude is more divergent than the melonic self-energy one.

Our result is a numerical computation of the amplitude, not analytical proof. One should always keep in mind its limitations. We performed a calculation with fixed boundary spins $j_b = \frac{1}{2}$, Immirzi parameter $\gamma = 0.1$, an extrapolation based on the truncation $\Delta l = 10$, and a uniform cutoff on all the faces limited to $K \leq 10$. Technical limitations and convenience dictate some choices. Nevertheless, we explored alternatives when possible, and the result seems general. Numerically we infer that the convergence of the vertex renormalization amplitude is determined by the destructive interference of the vertex amplitudes' oscillations of the EPRL theory. The booster functions are responsible for the interference, which appear in the EPRL vertex amplitude and are not present in the topological one. They encode the imposition of the simplicity constraints in the theory. This interpretation agrees with the results in [46]. Neglecting this interference results in a divergent upper bound estimate identical to the topological model.

Consider the Ponzano Regge model, a simpler spinfoam theory that describes Euclidean quantum gravity in three dimensions. The vertex renormalization amplitude in that theory (the 1-4 Pachner move) is cubically divergent in the cutoff K^3. The divergence is related to a residual gauge invariance in the path integral that is not entirely fixed [144]. Geometrically it can be interpreted in the ∞^3 ways we can divide a tetrahedron in four with an extra point. The convergence of the EPRL vertex renormalization amplitude could signal that a similar symmetry is absent in the theory. The restriction to Lorentzian geometries with space-like boundaries of EPRL vertex in the large spins regime breaks the BF action's shift symmetry. Whether or not it indicates that simplicity constraints are imposed correctly in the EPRL theory remains an open question. We hope that a detailed analytical study and the contribution of other upcoming numerical techniques tailored to the study of the large spin limit of the theory [32] could help solve this mystery.

Chapter 5

Markov Chain Monte Carlo methods for graph refinement in Spinfoam Cosmology

Statement of co-authorship

This Chapter presents the paper [78], written by me, Francesco Gozzini, and Francesca Vidotto. It was published in *Classical and Quantum Gravity* on 14 April 2023. My supervisor, Francesca Vidotto, originally started this line of research with Francesco Gozzini in [77]. Francesco Gozzini initiated the application of Markov Chain Monte Carlo methods to investigate spinfoam refinement just after [77] was published. After he completed his Ph.D., this project was left unfinished. I took over the work and rewrote all the algorithms, adapting the old codes to the numerical libraries and more recent strategies developed in the meantime. I took care of the numerical computations, drafted sections, and realized plots and images. I estimate the percentage of the total work conducted by me between 40% and 60%. This version corresponds to the manuscript accepted for publication on *Classical and Quantum Gravity*, not to the final published version in the same journal. To be compatible with the IOP's author rights policy, an embargo period of 4 months will be requested for this Ph.D. thesis.

5.1 Introduction

The last years have seen a lively development of numerical methods in the covariant, or *spinfoam*, formulation of Loop Quantum Gravity [1]. A key step in this direction has been the introduction of Markov chain Monte Carlo (MCMC) methods and their application to the study of the EPRL propagator [57]. In [57], the authors exploited the known properties of the single-vertex semiclassical limit to perform a Monte Carlo sampling over particular subspaces of the complexified parameter space, using semiclassical boundary states [145]. Exploiting the stationary phase approximation [146], the authors found a good agreement with the semiclassical results obtained via analytical methods.

The evaluation of spinfoam amplitudes in the Engle-Pereira-Rovelli-Livine model (EPRL) [31, 70, 71, 9], the introduction of effective models [63, 64] and the numerical study of cuboid renormalization [147] have shed considerable light on several aspects of the theory, such as the role of the Immirzi parameter, the accidental flatness constraints, and the refinement limit.

This paper introduces a technique that can be applied to some calculations in a regime where the number of degrees of freedom is large, but the relevant spin quantum numbers are small. We combine the Metropolis-Hastings algorithm [148] with some recently developed high-performance computing techniques in covariant LQG [48]. Not based on analytical approximations, this method requires minimal knowledge of spinfoam geometry. On the other hand, the algorithm becomes resource-demanding as the complexity of the spinfoam increases. We test the algorithm in the case of a single 4-simplex, and then we use it to study a spinfoam with six vertices in the bulk and twenty nodes on the boundary. This corresponds to the cellular decomposition obtained from one elementary 4-simplex by splitting each of the five boundary tetrahedra into four tetrahedra. The resulting spinfoam is a refinement of the 4-simplex vertex, which does not add any internal (dynamical) face to the spinfoam two-complex. The following refers to it as the "star" spinfoam.

We restrict the calculation to the homogeneous sector where the spins of all boundary links have the same value. In this sector, the boundary intertwiners give the spinfoam degrees of freedom, which encode the shapes of the boundary tetrahedra. We numerically compute these variables' amplitude as a function and use the Monte Carlo sampling to study the expectation values of different boundary operators and their correlations. For the sake of completeness, we investigate both the BF and EPRL models.

Spinfoam amplitudes with a regular graph on the boundary and homogeneous data can be interpreted as cosmological states [17]. In particular, when a single boundary state is considered, the amplitude can be seen as a transition from nothing into a 3-dimensional geometry, compatible with the 4-dimensional Lorentzian bulk [149, 150]. The computed amplitude truncates the spinfoam vertex expansion of the *nothing-to-geometry* transition amplitude. This provides a spinfoam Lorentzian version of the Hartle-Hawking wave function of the universe [151]. We refer to the literature in spinfoam cosmology for the physical interpretation of these states [16, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164].

These states are rich enough to describe a boundary geometry that is regular on average but allows quantum fluctuations. The recent advance in numerical methods applied to spinfoam amplitudes has opened the possibility of computing concrete observables with spinfoam cosmological states. The first computation of this kind was introduced in [77] using a single 4-simplex. In this paper, we would like to focus on numerical methods. The cosmological spinfoam states provide an exciting framework for investigating novel techniques.

A critical open question in the spinfoam approach to quantum gravity is the convergence of the amplitudes under refinement of the two complexes of the spinfoam. In this regard, we can confirm numerically that the refinement studied is stable for the boundary observable, in the sense that there is excellent agreement between expectation values computed on the single vertex graph and the refined graph. However, the correlations turn out to be different, as well as the quantum information entropy between different boundary nodes, reflecting the finer scale at which they become accessible. The results provide a quantitative estimate of the quantum correlations between different spatial patches in the manifold boundary whose truncation is represented by the boundary spin network.

The paper is organized as follows. In Section 5.2, we define the boundary state we use in our analysis. In Section 5.3, we discuss the Markov chain Monte Carlo method applied to the spin sums over the boundary degrees of freedom. In Section 5.4, we test the Monte Carlo sampler to the case of a single 4-simplex, where calculations can be performed using deterministic

approaches. In Section 5.5, we study the star spinfoam, investigating the numerical results of geometrical operators (boundary angles and volumes) and the related correlations. We also discuss the entanglement entropy between boundary nodes, considering different partitions into subsystems.

The code used for all the computations described in this paper is available on GitHub [103][1] All the computations were performed on Compute Canada's Cedar, Graham, Beluga, and Narval clusters (www.computecanada.ca). Therefore, the computational resources employed for this paper can be quantified as approximately $8 \cdot 10^4$ CPU hours.

5.2 The boundary state

We study the boundary state $|\psi_0\rangle$ introduced in [77], which explored the simplest triangulation of a 3-sphere, emerging from a single 4-simplex. Let Γ be a graph with L links and N nodes. Then, the LQG Hilbert space for the graph is:

$$\mathcal{H}_\Gamma = L_2\left[SU(2)^L/SU(2)^N\right] . \tag{5.1}$$

The spin network basis in $\mathcal{H}_\Gamma$ is made by the states $|\{j_l\},\{i_n\}\rangle$ (from now on, we omit the Γ subscript), where $\{j_l\}$ is a set of half-integer spins and $\{i_n\}$ an intertwiner set, $n = 1\dots N$, $l = 1\dots L$. An intertwiner i_n is a basis element of the invariant subspace of the tensor product of 4 $SU(2)$ representations at the node n. In the following, we fix all the spins to be equal, namely $j_l = j$. We denote a boundary spin network state of this reduced space as:

$$|j,\{i_n\}\rangle \equiv |j,i_1\dots i_N\rangle = |j,i_1\rangle\otimes\cdots\otimes|j,i_N\rangle , \tag{5.2}$$

suppressing the curly brackets for the spin label j, as one common spin is attached to all the links. We define the state $|\psi_0\rangle$ in the Hilbert space (5.1) by

$$\langle j,\{i_n\}|\psi_0\rangle \equiv A\left(j,\{i_n\}\right) , \tag{5.3}$$

where $A\left(j,\{i_n\}\right)$ is the LQG amplitude of the state in the spin network basis. The amplitude (5.3) can be interpreted as the amplitude associated with the transition nothing-to-$|j,\{i_n\}\rangle$. Hence $|\psi_0\rangle$ gives the natural state projected out of the empty state by the LQG dynamics. The amplitude function depends on the common spin j on the links and all the N intertwiner indices. The state $|\psi_0\rangle$ is therefore defined as:

$$|\psi_0\rangle = \sum_{\{i_n\}} A\left(j,\{i_n\}\right)|j,\{i_n\}\rangle . \tag{5.4}$$

The sum is over all possible values of all the intertwiners in the set $\{i_n\}$, compatible with triangular inequalities. If $j_l = j$, then every intertwiner i_n can assume integer values between 0 and $2j$. This gives $(2j+1)^N$ boundary basis elements that enter the sum (5.4). Following the geometrical interpretation of the covariant LQG phase space in terms of twisted geometries [128], we might interpret the constraint $j_l = j$ as imposing strongly at the quantum level that all the areas of the faces of the boundary tetrahedra must be equal. The intertwiner degrees

[1]The code works on any operating system with an updated version of the Julia programming language (the code was tested with Julia 1.6.2).

of freedom model the "shape" of the boundary tetrahedra, which are relational observables at a given value j. They are directly linked to the boundary 3d dihedral angles, as discussed in Section 5.5.1. The definition (5.4) doesn't depend on the details of the triangulation. The triangulation determines how the amplitude $A(j,\{i_n\})$ must be computed.

5.2.1 Expectation values

We consider local geometrical operators acting on single boundary nodes of $\mathcal{H}$. We specify the matrix elements for each operator in the basis states (5.2). We start defining the normalized expectation value on the boundary state (5.4) of an operator O_k, acting on the Hilbert space associated with the k-th node, as:

$$\langle O_k\rangle \equiv \frac{1}{Z}\langle\psi_0|O_k|\psi_0\rangle\ . \tag{5.5}$$

The normalization factor is computed as follows:

$$Z \equiv \langle\psi_0|\psi_0\rangle = \sum_{\{i_n\}} A(j,\{i_n\})^2\ . \tag{5.6}$$

From (5.4) we write:

$$\langle\psi_0|O_k|\psi_0\rangle = \frac{1}{Z}\sum_{\{i_n\}}\sum_{\{i'_n\}} A(j,\{i_n\})\,A(j,\{i'_n\})\,\langle j,\{i'_n\}|O_k|j,\{i_n\}\rangle\ . \tag{5.7}$$

By using the orthogonality of the spin-network states (5.2) we find:

$$\langle j,\{i'_n\}|O_k|j,\{i_n\}\rangle = \delta_{i'_1,i_1}\ldots\langle j,i'_k|O_k|j,i_k\rangle\ldots\delta_{i'_N,i_N}\ , \tag{5.8}$$

therefore we conclude:

$$\langle O_k\rangle = \frac{1}{Z}\sum_{\{i_n\}}\sum_{i'_k=0}^{2j} A(j,\{i_n\})\,A(j,\{i_n\},i'_k)\,\langle j,i'_k|O_k|j,i_k\rangle\ , \tag{5.9}$$

where $A\left(j,\{i_n\},i'_k\right)$ is defined as:

$$A(j,\{i_n\},i'_k) \equiv A(j,i_1\ldots i'_k\ldots i_N)\ , \tag{5.10}$$

namely, the amplitude computed with i'_k in place of i_k. Since $i_k \in \{i_n\}$, the sum over i_k is contained in the sum over the set $\{i_n\}$. It is now straightforward to compute $\langle O_kO_m\rangle$, which turns out to be:

$$\langle O_kO_m\rangle = \frac{1}{Z}\sum_{\{i_n\}}\sum_{i'_k=0}^{2j}\sum_{i'_m=0}^{2j} A(j,\{i_n\})\,A(\{j\},\{i_n\},i'_k,i'_m)\,\langle j,i'_k|O_k|j,i_k\rangle\langle j,i'_m|O_m|j,i_m\rangle\ , \tag{5.11}$$

where the meaning of $A\left(j,\{i_n\},i'_k,i'_m\right)$ is transparent by looking at (5.10). We refer to the amplitude with i'_k instead of i_k and i'_m in place of i_m. In the case of diagonal operators D_k in the spin-network basis, equations (5.9) and (5.11) become respectively:

$$\langle D_k\rangle = \frac{1}{Z}\sum_{\{i_n\}} A^2(j,\{i_n\})\,\langle j,i_k|D_k|j,i_k\rangle\ , \tag{5.12}$$

$$\langle D_k D_m \rangle = \frac{1}{Z} \sum_{\{i_n\}} A^2 \left(j, \{i_n\}\right) \langle j, i_k | D_k | j, i_k \rangle \langle j, i_m | D_m | j, i_m \rangle \ . \tag{5.13}$$

Normalized correlations are defined as:

$$C(O_k, O_m) = \frac{\langle O_k O_m \rangle - \langle O_k \rangle \langle O_m \rangle}{(\Delta O_k)\,(\Delta O_m)} \ , \tag{5.14}$$

where the quantum spread is:

$$\Delta O_k = \sqrt{\langle O_k^2 \rangle - \langle O_k \rangle^2} \ . \tag{5.15}$$

The fact that the connected correlation function (5.14) between the nodes k and m is non-vanishing turns out to be a necessary condition to have correlated fluctuations between the shapes of the tetrahedra dual to nodes k and m [165, 166, 167, 146, 126].

5.3 Monte Carlo over intertwiner space

As discussed in Section 5.2.1, an operator's expectation value must sum over all possible eigenstates of the quantum system. Numerically, this rapidly becomes intractable as the number of degrees of freedom increases. In the present context, for a graph with N boundary intertwiners, there are $(2j+1)^N$ values to compute and to sum. Suppose that the amplitude function $A\left(j, \{i_n\}\right)$ can be computed in 10^{-6} seconds on reference hardware (the real-time is typically orders of magnitude larger). For 20 boundary tetrahedra, which is the case of the star spinfoam discussed in Section 5.5, a spin $j = 2$ computation would take three years. We cannot use blind summation if we want to approach this problem numerically. Even parallelizing the computation on multiple machines cannot solve this issue in the case of many boundary degrees of freedom.
A solution is Monte Carlo summation. This technique is used to compute the expectation values of random variables. We obtained the best results by adapting the Metropolis-Hastings algorithm [148, 168] to the discrete sums over the boundary intertwiners. For completeness, we briefly describe the Metropolis-Hastings algorithm[2] in its general form.

5.3.1 Metropolis-Hastings algorithm

Let us consider a quantity O, which can be computed as:

$$O = \sum_{x \in \chi} \tilde{f}_\chi \left(x\right) o\left(x\right) \ , \tag{5.16}$$

where x is a (possibly multidimensional) discrete variable on a state space χ which must be summed over, while $\tilde{f}_\chi$ is a probability distribution function on χ, so that:

$$\sum_{x \in \chi} \tilde{f}_\chi \left(x\right) = 1 \ . \tag{5.17}$$

[2]We refer to the original paper [148] or to the numerous texts available for a deeper description of the algorithm.

From now on, we define $\tilde{f}_\chi$ as the target distribution and omit the state space label χ. Since the target distribution is normalized, we write:

$$\tilde{f}(x) \equiv \frac{f(x)}{\sum_x f(x)} , \tag{5.18}$$

from which:

$$O = \frac{\sum_x f(x)\, o(x)}{\sum_x f(x)} . \tag{5.19}$$

If the target distribution (5.18) can be computed up to a multiplying constant, the Metropolis-Hastings algorithm allows constructing on state space χ an ergodic Markov chain with length N_{MC}: $x_1,\ x_2,\ x_3 \ldots, x_n\ \ldots, x_{N_{MC}}$ such that x_n is converging (in distribution) to $\tilde{f}$, exploring the space χ progressively. If we define:

$$O_{N_{MC}} = \frac{1}{N_{MC}} \sum_{n=1}^{N_{MC}} o(x_n) \ , \tag{5.20}$$

Then, since the chain can be considered as a statistical sample, the law of large numbers ensures that:

$$\lim_{N_{MC}\to\infty} O_{N_{MC}} = O . \tag{5.21}$$

The computation is stochastic, and the correct result is found only in the limit of an infinite number of samples. This allows us to write:

$$O_{N_{MC}} \approx O \quad \text{for } N_{MC} \gg 1 . \tag{5.22}$$

We estimate the original sum (5.16). The soundness of the procedure comes from known theorems on Monte Carlo summation. We can estimate the error by comparing many different runs. Since the simulation is Markovian and the chain can be considered a statistical sample, the latter usually depends on the starting value. In the following, we randomly choose the starting point of the Markov chain. While the chain is in thermalization, the initial steps are typically removed as burn-in iterations. To transit from the chain state x_n to x_{n+1}, we require a proposal distribution g defined on space χ. If g is positive everywhere, then the Metropolis-Hastings algorithm preserves $\tilde{f}$ as the stationary distribution to which the chain is progressively converging. In the random walk Metropolis-Hastings, the proposal distribution consists of a local exploration of the neighborhood of the current value x_n of the Markov chain. That is, the proposed value x'_n is simulated as:

$$x'_n = x_n + \delta x_n , \tag{5.23}$$

where δx_n is a random perturbation with distribution g. The proposed state x'_n is sampled from a probability distribution $g(x'_n|x_n)$, which suggests a candidate given the previous sample value x_n. As proposal distribution, we choose a truncated normal distribution rounded to integers centered around x_n with standard deviation σ:

$$g(x'_n|x_n) = \mathcal{N}_{d,t}(x_n, a, b; \sigma) , \tag{5.24}$$

Algorithm 3 Random walk Metropolis-Hastings

Choose the number of iterations N_{MC}, the burn-in parameter b and the standard deviation σ of the proposal distribution g.
Set a random initial configuration x and compute $f(x)$
Set initial multiplicity to 1
for $n = 1 \dots N_{MC}$ **do**
    Generate a candidate x' from x according to the proposal distribution g
    if $x' = x$ **then**
        Increase the multiplicity by 1
        continue
    else
        Compute $f(x')$
        Compute $p = \min\left\{1, \frac{f(x')}{f(x)} \frac{g(x|x')}{g(x'|x)}\right\}$
        Generate a uniform random number r between 0 and 1
        if $r < p$ **then**
            if $n > b$ **then**
                Store x and $f(x)$ with the corresponding multiplicity
            Set $x = x'$, $f(x) = f(x')$
            Set the multiplicity to 1
        else
            Increase the multiplicity by 1
Dump to disk the x's, $f(x)$'s and the corresponding multiplicities.

where the definition of $\mathcal{N}_{d,t}$ is reported in E.1. The full algorithm is summarized in the flowchart 3, in which we report the steps to implement the random walk Metropolis-Hastings and build the Markov chain. The multiplicity factors and the storage of $f(x)$'s have been introduced as a matter of efficiency. This considerably speeds up the algorithm and the consequent computation of operators. Technically, the Markov chain obtained at the end of the algorithm 3 has a length less than N_{MC} (as this depends on the acceptation ratio). Since it is sufficient to take into account the multiplicity of each single chain state to restore the original length, in the following, we refer to the Markov chain obtained at the end of the algorithm 3 as having length N_{MC} without losing any generality.
Since x_n depends on the previous element along the Markov chain, this induces a non-zero correlation between x_n and x_{n+d}. The correlation between x_n and x_{n+d} is the autocorrelation at lag d. The autocorrelation should decrease as the lag increases for a Markov chain that converges to a stationary distribution. Although the most common approach is to evaluate the autocorrelation of operators, a measure of the degree of autocorrelation of the Markov chain is represented by the autocorrelation of the sequence of the amplitudes of the states. If we define the average of the states' amplitude as:

$$\bar{f}(x) = \frac{1}{N_{MC}} \sum_{n=1}^{N_{MC}} f(x_n) \,, \tag{5.25}$$

the definition of the autocorrelation function at lag d associated with the sequence $f(x_1) \dots f(x_n)$ can be written as:

$$\mathrm{ACF}\left(x_1 \dots x_{N_{MC}}; d\right) = \frac{\sum_{n=1}^{N_{MC}-d} \Big(f(x_n) - \bar{f}(x)\Big)\Big(f(x_{n+d}) - \bar{f}(x)\Big)}{\sum_{n=1}^{N_{MC}} \Big(f(x_n) - \bar{f}(x)\Big)^2} \,. \tag{5.26}$$

We can compare the results of operators' expectation values over different runs to obtain an unbiased estimate of the statistical fluctuations due to the Monte Carlo sampling. We store multiple Markov chains according to algorithm 3, computing operators for each. This is extremely useful in determining the convergence of the Markov Chain to the stationary distribution and the corresponding unbiased statistical dispersion of the operators. This is discussed in Section 5.3.2.

5.3.2 Expectation values with Monte Carlo

To apply the Metropolis-Hastings algorithm discussed in Section 5.3.1 to the computation of spinfoam observables of Section 5.2.1, a direct comparison between equations (5.12) and (5.16) is enlightening. Namely, if we associate to the state space χ the intertwiners' boundary space (5.2), so that $x = \{i_n\}$, then the target distribution becomes:

$$\tilde{f}\left(x\right) = \frac{A^2\left(j, \{i_n\}\right)}{\sum_{\{i_n\}} A^2\left(j, \{i_n\}\right)} \,. \tag{5.27}$$

Namely, we run a stochastic sampling routine that extracts draws of intertwiners $[i_1 \dots i_N]$ from their whole configuration space, according to the Markov chain. The proposal distribution corresponds to a discrete multivariate truncated normal distribution, and each intertwiner is proposed sampling from an independent one-dimensional distribution. The a, b parameters in (5.24) for each intertwiner are 0 and $2j$, respectively. After storing the intertwiner draws, we can use them to compute the expectation values of operators. We can summarize the introduction of the Monte Carlo with the following substitution in the formulas of the expectation values of the operators:

$$\sum_{\{i_n\}} A^2\left(j, \{i_n\}\right) o\left(\{i_n\}\right) \approx \sum_{[i_n]} o\left([i_n]\right) \,. \tag{5.28}$$

In the right side of equation (5.28), the sum over the intertwiners is intended as the sum over the stored draws $[i_n] \equiv [i_1 \dots i_N]$ in which the intertwiners have a fixed value compatible with triangular inequalities. We are no longer considering all the independent summations over the intertwiners. This hugely reduces the computational cost, making the computation feasible in the case of many boundary degrees of freedom. With (5.28), the normalization factor (5.6) becomes:

$$Z \approx \sum_{[i_n]} = \text{Number of MC iterations} \equiv N_{MC} \,. \tag{5.29}$$

We can easily find the expression for the expectation values of non-diagonal operators by multiplying and dividing for $A\left(j, \{i_n\}\right)$ and then using (5.28), remembering (5.29). In fact, (5.9)

becomes:

$$\langle O_k \rangle \approx \frac{1}{N_{MC}} \sum_{[i_n]} \sum_{i'_k=0}^{2j} \frac{A\left(j, [i_n], i'_k\right)}{A\left(j, [i_n]\right)} \langle j, i'_k | O_k | j, i_k \rangle \, . \tag{5.30}$$

Equation (5.11) becomes:

$$\langle O_k O_m \rangle \approx \frac{1}{N_{MC}} \sum_{[i_n]} \sum_{i'_k=0}^{2j} \sum_{i'_m=0}^{2j} \frac{A\left(j, [i_n], i'_k, i'_m\right)}{A\left(j, [i_n]\right)} \langle j, i'_k | O_k | j, i_k \rangle \langle j, i'_m | O_m | j, i_m \rangle \, . \tag{5.31}$$

In case of diagonal operators, from (5.12) and (5.13) we obtain:

$$\langle D_k \rangle \approx \frac{1}{N_{MC}} \sum_{[i_n]} \langle j, i_k | D_k | j, i_k \rangle \, , \tag{5.32}$$

$$\langle D_k D_m \rangle \approx \frac{1}{N_{MC}} \sum_{[i_n]} \langle j, i_k | D_k | j, i_k \rangle \, . \langle j, i_m | D_m | j, i_m \rangle \, . \tag{5.33}$$

Notice that in the case of diagonal operators, it is unnecessary to compute any amplitude except those necessary for sampling the draws of intertwiners. This makes the computation of diagonal operators several orders of magnitude faster than non-diagonal ones. We consider the numerical analysis up to the value $j = 6$ for the spins associated with the boundary links. This numerical approach is intended to be applied in the full quantum regime, i.e., when the spin quantum numbers are small and the semiclassical approximation is invalid. In addition, the computational complexity represented by increasing j strongly depends on the type of the considered operator and the Metropolis-Hastings parameters. The value $j = 6$ allows computing all the operators and correlation functions we consider with a stable precision of up to 3 significant digits.

Crucially, this approach is not based on analytical approximations and requires minimal knowledge of spinfoam geometry, which typically becomes quite complicated for models beyond the single 4-simplex. On the other hand, the price to pay is the calculation of the spinfoam amplitude at each iteration of the algorithm 3. Much effort has been devoted in recent times to the efficient computation of spinfoam amplitudes [9, 48, 47]. At present, the best numerical framework to compute BF and EPRL vertex amplitudes, which can be seen as the elementary building blocks of more general triangulations, is the `sl2cfoam-next` library [48]. Calculating the spinfoam amplitude with high-performance computing techniques is fundamental for applying the Metropolis-Hastings algorithm discussed in 5.3.1. Recent developments have made it possible to compute potentially divergent spinfoam amplitudes with many internal faces [133, 71]. This makes the algorithm presented in this paper a good candidate to be used in the case of spinfoams with a non-trivial dynamic structure.

After we have stored a number C of Markov chains, each with the same length N_{MC} and Metropolis-Hastings parameters, we can compute the expectation value (5.30) of an operator O_k for each chain $\langle O_k \rangle_1 \ldots \langle O_k \rangle_C$ and then consider the corresponding average and standard deviation:

$$\mu_{\langle O_k \rangle} = \frac{\sum_{c=1}^{C} \langle O_k \rangle_c}{C} \, , \tag{5.34}$$

$$\sigma_{\langle O_k\rangle} = \sqrt{\frac{\sum_{c=1}^{C}\left(\mu_{\langle O_k\rangle} - \langle O_k\rangle_c\right)^2}{C}}\,. \tag{5.35}$$

For each considered operator, we plot the corresponding Gaussian distribution:

$$G(x;\mu_{\langle O_k\rangle},\sigma_{\langle O_k\rangle}) = \frac{1}{\sigma_{\langle O_k\rangle}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu_{\langle O_k\rangle}}{2\sigma_{\langle O_k\rangle}}\right)^2}\,. \tag{5.36}$$

The number of Monte Carlo iterations N_{MC} for each Markov chain, the number C of averaged chains, and the relevant parameters in the Metropolis-Hastings algorithm are all listed in tables in the E.2.

5.4 The 4-simplex

The 4-simplex, or (referring to the dual triangulation) the vertex, is the simplest geometrical triangulation of the 3-sphere. It is formed by 5 tetrahedra glued on 10 faces. There are 5 boundary degrees of freedom and no internal faces. As shown in Figure 5.1, there is a complete self-duality between the geometrical triangulation of the 4-simplex and the corresponding boundary graph. The geometry in Figure 5.1 doesn't depend on the Lorentzian character of the

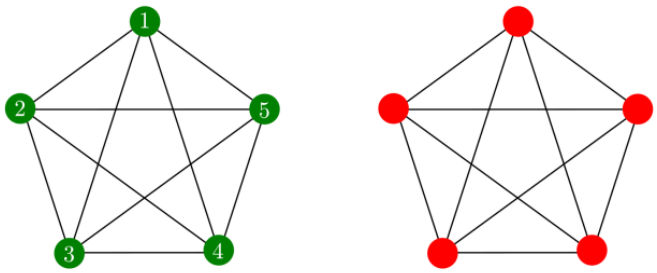


Figure 5.1: *Left: Geometry of the 4-simplex triangulation. Numbered circles correspond to points. Right: The corresponding boundary spin network. Each red circle corresponds to a boundary node, and each line corresponds to a boundary link.*

triangulation or not. Namely, it is the same for the $SU(2)$ topological BF or the Lorentzian EPRL 4-simplex. We need to explicitly represent the spinfoam associated with the amplitude to emphasize the difference between the two models. We denote such amplitudes associated with a single vertex as V_{BF} and V_{EPRL}. We write the corresponding expression in the general form, in which all spins have different values. As discussed in Section 5.2, in the present context, we focus on the case where all spins are equal.

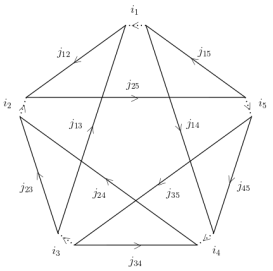
5.4.1 The BF and EPRL vertex amplitudes

The vertex amplitude of the topological BF model can be written as an $SU(2)$ invariant Wigner $\{15j\}$ symbol. The choice of the recoupling basis on each intertwiner determines if the symbol

can be reduced to the product of lower-dimensional symbols. We choose the irreducible symmetric $\{15j\}$ symbol of *first kind*, following the convention of [169]. The definition in terms of $\{6j\}$ symbols is:

$$V_{\mathrm{BF}}(j_{ab};\, i_a) = (-1)^{\sum_{k=1}^{5}(j_k+i_k)} \sum_s (2s+1) \begin{Bmatrix} i_1 & j_{25} & s \\ i_5 & j_{14} & j_{15} \end{Bmatrix} \begin{Bmatrix} j_{14} & i_5 & s \\ j_{35} & i_4 & j_{45} \end{Bmatrix} \begin{Bmatrix} i_4 & j_{35} & s \\ i_3 & j_{24} & j_{34} \end{Bmatrix} \times \begin{Bmatrix} j_{24} & i_3 & s \\ j_{13} & i_2 & j_{23} \end{Bmatrix} \begin{Bmatrix} i_2 & j_{13} & s \\ i_1 & j_{25} & j_{12} \end{Bmatrix}$$

$$=$$



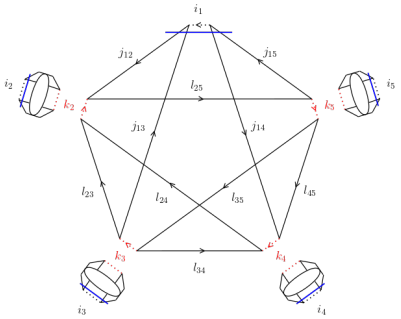
, (5.37)

where $a, b = 1 \ldots 5$, $a \neq b$. To avoid weighing down the notation, boundary dimensional factors attached to intertwiners i_e and spins j_{ab} have been neglected. The $\{6j\}$ Wigner symbols in (5.37) can be computed efficiently with libraries such as `wigxjpf` and especially `fastwigxj` [91, 170].

The EPRL vertex amplitude is built from the topological $SL(2, C)$) spinfoam vertex amplitude once the simplicity constraints have been imposed [25, 113]. We use the form of amplitude originally derived in [42], which results in a linear combination of $\{15j\}$ symbols weighted by one booster functions B_4^γ per edge (see E.3 for explicit formulas). We write the vertex amplitude according to the graphical notation discussed in detail in [9]:

$$V^{\gamma}_{EPRL}(j_{ab},\, i_a;\, \Delta l) = \sum_{l_{eq}=j_{eq}}^{j_{eq}+\Delta l} \sum_{k_e} \left(\prod_e (2k_e+1) B_4^\gamma(j_{1e}, l_{eq}; i_e, k_e) \right) V_{BF}\left(j_{1e}, l_{eq}; k_e, i_a\right)$$

$$= \sum_{l_{eq}=j_{eq}}^{j_{eq}+\Delta l}$$



, (5.38)

where $e, q = 2 \ldots 5$, $e \neq q$. We introduced the Δl parameter to truncate the summation over the auxiliary spins l_{eq}. The role of this truncation parameter in the context of infrared divergences has been deeply discussed in detail in [71, 133, 72]. In this paper, we consider $\Delta l = 20$ and $\gamma = 1.2$ as the value of the Barbero-Immirzi parameter. It is interesting to notice how the truncation parameter Δl seems to play a minimal role in the computation of boundary observables, as already observed in [71] in the case of divergent infrared amplitudes. In the present context, except for a slight systematic shift in the expectation values of the dihedral

angles of the star model (discussed in Section 5.5.1), we found no difference[3] between the case $\Delta l = 0$ and $\Delta l = 20$. From a computational point of view, the advantage of using a small Δl parameter is remarkable. The 4 spins associated with the gauge-fixed edge are $(j_{12}, j_{13}, j_{14}, j_{15})$, as the elimination of a redundant $SL(2,\mathbb{C})$ integration along one edge in the EPRL vertex (5.38) is necessary to ensure that the corresponding amplitude is well defined [85]. As in (5.37) we neglected the dimensional factors attached to the boundary intertwiners and spins.

5.4.2 Testing the Monte Carlo sampler

The 4-simplex is an excellent model to test the sampling routine over the intertwiners space discussed in Section 5.3.2. Since the number $N = 5$ of boundary degrees of freedom is low, it is possible to compute observables and related functions without resorting to Monte Carlo methods, allowing for a direct comparison. To perform the sampling algorithm, it is necessary to compute the vertex amplitudes (5.38)-(5.37) at a fixed value of boundary spin j and for all the possible $(2j+1)^5$ values of intertwiners. For the EPRL model, as mentioned in Section 5.4.1, the computational time considerably increases with j, especially with a high value of the truncation parameter Δl. The computation of the EPRL vertex amplitude (5.38) for $j = 0.5, 1, \dots 6$ has been distributed over several machines and hundreds of CPUs, and it has taken about 5 days to complete. The amplitudes are available at the public repository [103], along with the corresponding BF counterparts (5.37). We start by looking at the autocorrelation function (5.26) of the vertex amplitudes sampled during the random walk. To measure the autocorrelation, we set $b = 0$ in the flowchart 3, while all the other parameters are the same as reported in E.2. We report the data in Figure 5.2. Obviously, the most important parameter

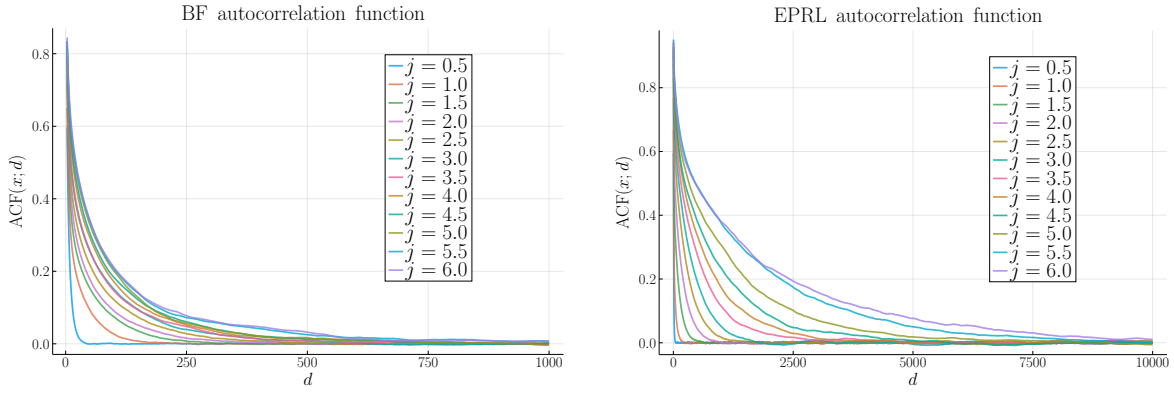


Figure 5.2: *Autocorrelation function* (5.26) *of the BF and EPRL vertex amplitudes as a function of the lag d for different values of j. As expected for a Markov chain that converges to a stationary distribution, the autocorrelation decreases as a function of the lag.*

to verify the sampler's effectiveness is the observables' computation. In this respect, a direct comparison is made easier because this simple model's external dihedral angle operator has been studied in [77]. The dihedral angle operator is the simplest operator to compute in the intertwiner basis, and it describes the cosine of the external dihedral angle $\cos(\theta_{ab})$ between two faces a and b of a boundary tetrahedron. Faces a and b depend on the recoupling basis

[3] In order not to be redundant, we do not explicitly report the data in the case $\Delta l = 0$.

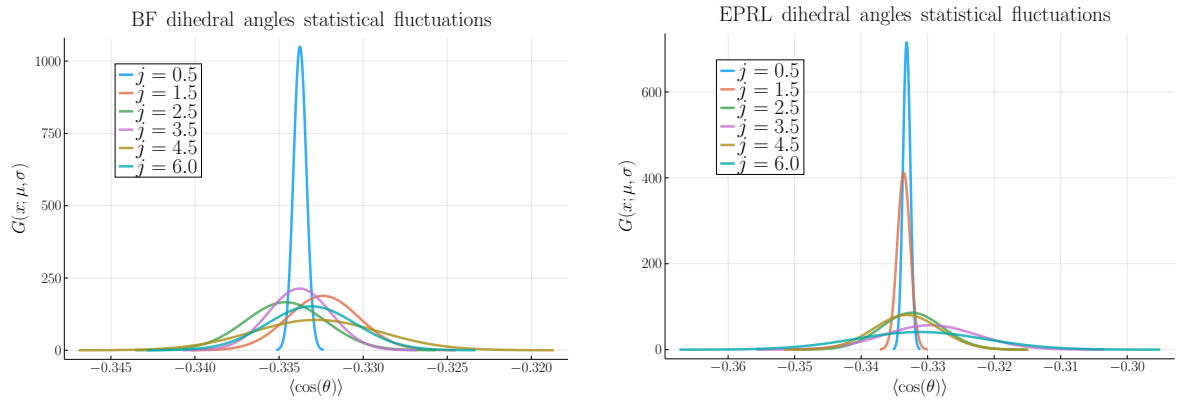


Figure 5.3: *Gaussian distribution* (5.36) *of the expectation values* (5.32) *of the dihedral angle operator* (5.39) *in the 4-simplex model. We averaged over several runs, computing the (average) angle* $\mu_{<\cos\theta>}$ *defined on a single node and the corresponding standard deviation* $\sigma_{<\cos\theta>}$ *for different values of* j.

chosen for the invariant $SU(2)$ $\{15\,j\}$ symbol, which appears both in the EPRL (5.38) and BF amplitude (5.37). The external dihedral angle of the tetrahedron dual to the node n in the symmetry-reduced space basis states (5.2) is [77]:

$$\langle j, i_n | \cos(\theta) | j, i_n \rangle = \frac{i_n(i_n+1) - 2j(j+1)}{2j(j+1)} \,. \tag{5.39}$$

The dihedral angle operator (5.39) is diagonal in the spin-network basis. Therefore, we can quickly compute it with equation (5.32). We show the statistical fluctuations (5.36) of the expectation values (5.32) in the case of the dihedral angle operator (5.39) in Figure 5.3 for some values of j. All the parameters used in the Metropolis-Hastings algorithm are reported in E.2. The results show that the angle average corresponds to a regular tetrahedron, which is the result originally obtained in [77] with deterministic calculations. It is clear how stochastic fluctuations in the random-walk over intertwiner space tend to grow much faster for EPRL, rather than for BF, as boundary spin j increases. This interesting behavior is the main reason why we used a number of N_{MC} iterations larger than one order of magnitude in the analysis of the star model, as discussed in Section 5.5. In order to have a stable precision up to the third significant digit, especially for large values of j, it is necessary to set a sampling number $N_{MC} \sim 10^6$, compared to $(2j+1)^5$ sums to be carried out in the exact calculation (5.12). Although this indicates that the sampler works as expected, this makes using the random walk Metropolis-Hastings algorithm extremely inefficient in the case of a single 4-simplex. However, as for other (Markov Chain) Monte Carlo methods, the advantage obtained in the case of many degrees of freedom emerges surprisingly, as we show in the case of the star spinfoam amplitude. This is because the Metropolis-Hastings algorithm (as well as other MCMC methods) is not affected by the problem known as *curse of dimensionality*.

As mentioned in Section 5.3, the statistical fluctuations in Figure 5.3 are very useful in determining the convergence of the Markov Chain to the stationary distribution, as well as the dispersion of the operators. For the star model, which is the main element of analysis in the present context, we explicitly report the fluctuations (5.36) computed for all the considered operators and for the entanglement entropy.

5.5 The star

After testing the algorithm discussed in Section 5.3 to the 4-simplex case, we are ready to study the star spinfoam model, in which a computation without resorting to Monte Carlo methods would be impossible. The 2-complex of the star is composed of 6 vertices (one completely internal), 5 edges, and has no internal faces. The boundary graph is a refinement of the 4-simplex graph obtained by splitting each of the 5 nodes into 4 nodes. The final result of this refinement process is that we obtain 20 nodes on the boundary, which correspond dually to 20 boundary tetrahedra. Therefore the full triangulated manifold is composed of five 4-simplices glued on 5 internal tetrahedra, each 4-simplex showing 4 tetrahedra on its boundary. The triangulation of the star graph is shown in Figure 5.4, along with the boundary spin network. The triangulation of the star model is non-regular since there are segments that are shared

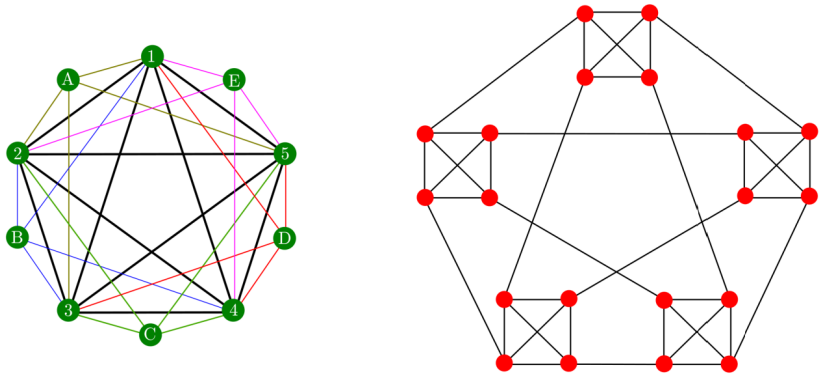


Figure 5.4: *Left: Geometry of the star triangulation. Numbered circles correspond to points, and each letter is associated with a unique 4-simplex. Colored lines are shared by three tetrahedra belonging to the same 4-simplex. Right: The corresponding boundary spin network. Each red circle corresponds to a boundary node, and each line corresponds to a boundary link.*

between 3 tetrahedra and other segments that are shared by 6 tetrahedra. Notice that the dual graph is a sort of “magnification" of the dual 4-simplex. Iterating the same procedure, we obtain a fractal structure. The EPRL star amplitude is constructed starting from the vertex amplitudes (5.38), shown in Figure 5.5. The diagram for the BF is similar, with the difference that each vertex amplitude is simply given by (5.37). That is, with respect to the graphical amplitude in Figure 5.5, there are no booster functions (and therefore no Y-map). In order not to be redundant, we do not also report the BF spinfoam diagram explicitly. The labels refer to the triangulation shown in Figure 5.4. The 4-simplices are labeled with one letter and four points. The boundary intertwiners are labeled by one letter and three points, which indicate the corresponding tetrahedron in the triangulation (as there is one intertwiner for each node). The links shared by three 4-simplices are labeled by three points, as they are dual to triangles. Those connecting two nodes belonging to the same 4-simplex are labeled with one letter and two points. Finally, the intertwiners connecting the booster functions with the $\{15\,j\}$ symbols are labeled with the position of the node in the corresponding 4-simplex. We can write the analytical expression of the amplitude associated with the star spinfoam EPRL and

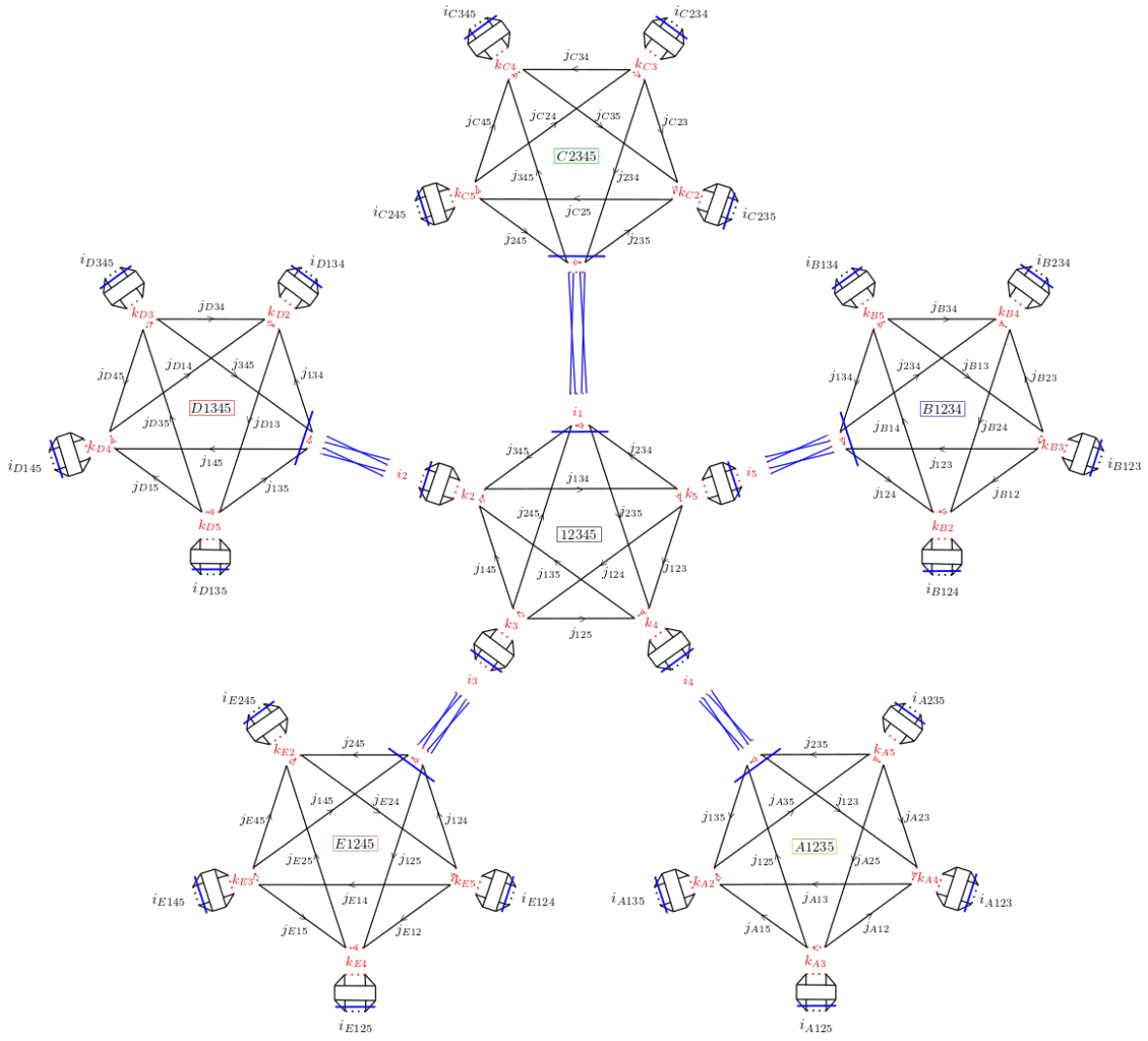


Figure 5.5: Graphical representation of the EPRL spinfoam amplitude corresponding to the star triangulation geometry described in Figure 5.4. Excluding the intertwiners connecting the booster functions with the $\{15j\}$ symbols in the same vertex, there are 5 internal intertwiners that need to be summed over.

BF, respectively, as:

$$A^{\gamma}_{EPRL}(j,\, i_b,\, \Delta l) = \sum_{i_1 \dots i_5} V^{\gamma}_{EPRL}(j,\, i_1,\, i_2,\, i_3,\, i_4,\, i_5,\, \Delta l) \prod_{a=1}^{5} V^{\gamma}_{EPRL}(j,\, i_a,\, i_b,\, \Delta l)\ , \tag{5.40}$$

$$A_{BF}(j,\, i_b) = \sum_{i_1 \dots i_5} V_{BF}(j,\, i_1,\, i_2,\, i_3,\, i_4,\, i_5) \prod_{a=1}^{5} V_{BF}(j,\, i_a,\, i_b)\ , \tag{5.41}$$

where we used the expressions for the EPRL and BF vertex amplitude (5.38)-(5.37). The dependence on intertwiners for each vertex (apart from those on which it is necessary to sum over to assemble the amplitude) has been generically indicated with i_b in order not to weigh down the notation. The combinatorial structure of the spinfoam should be clear by looking at Figure 5.5.

A simple benchmark

Before discussing the expectation values, it is interesting to estimate the computational time of the Metropolis-Hastings algorithm, discussed in Section 5.3.2, applied to the star spinfoam amplitude. As shown in equations (5.40)-(5.41) and in the flowchart 3, at each step of the Markov chain, we need to perform the contraction of the vertex amplitudes over 5 bulk intertwiners. This is undoubtedly the computationally most expensive part of the algorithm. For low spins, it is sufficient to perform the contraction with HPC techniques exploiting the CPU solely. We found the best performance using the `LoopVectorization` Julia package. The offloading of tensor contractions on the GPU with parallelization on the GPU cores [171], exploiting the recent tensor network techniques [104], will be implemented in future works. The best improvement is obtained for large values of the spins [48], making this approach more suitable for a study of the semiclassical limit of spinfoams rather than the quantum regime. We show a random walk sampling algorithm benchmark in Figure 5.6 for increasing values of the total number of iterations N_{MC} in the Markov chain. The sampling in Figure 5.6 has been carried

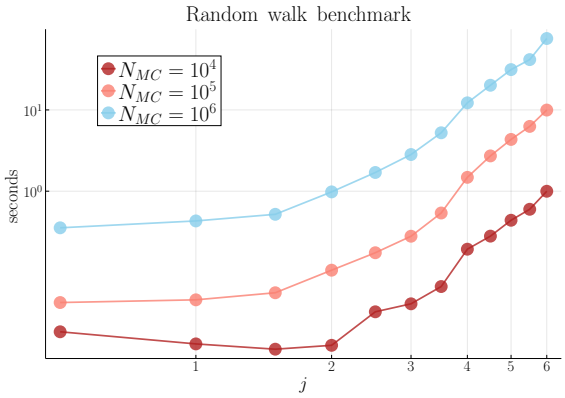


Figure 5.6: *Benchmark of the sampling and storage process of intertwiners draws in the random walk 3 over the 20-dimensional intertwiners' space of the star spinfoam amplitude 5.5. Computation time asymptotically scales as* $\sim j^{4.5}$.

out on a laptop with processor Intel(R) Core(TM) i7-10750H 2.60GHz. The acceptance rate of intertwiners draws has been set between 30% and 33%, with a burn-in parameter $b = 10^3$. In the code available at the repository [103], the Markov chains are automatically parallelized on the number of available CPUs, eventually distributing the computation on multiple machines. As discussed in Section 5.3.2, building more Markov chains is useful for improving accuracy and estimating the error committed due to the statistical fluctuations of the random walk.

5.5.1 Numerical results: operators

We now describe the numerical values obtained for the expectation value of local geometric operators (5.9) with the boundary state (5.4) for the star spinfoam amplitude. We also compute the quantum spread (5.15) and correlation functions (5.11) between different nodes. We discuss the results obtained with each geometrical operator's BF and EPRL models. The parameters used for sampling the draws employed for the computation of the operators are discussed in detail in E.2.

The dihedral angle operator

The dihedral angle operator has already been discussed in Section 5.4.2, as we used it to test the Monte Carlo sampler. In the star model, since all boundary tetrahedra are equal and regular, we can improve the statistic with a further average[4]. Namely, we can compute the expectation value of the dihedral angle (5.39) and the corresponding quantum spread (5.15) for all 20 nodes of the spinfoam independently at fixed boundary spin j, then averaging the results, which are shown in Figure 5.7. The results show that the expectation value of the boundary dihedral angle (5.39) is peaked to the value corresponding to an equilateral tetrahedron, which is the same result obtained in Section 5.4 with the simplest possible triangulation of a 3-sphere. This indicates that in the evolution from 1 to 6 vertices, the spatial metric of the boundary state still averages to that of the 3-boundary of a regular 4-simplex, i.e., that of a 3-sphere. This is not a trivial consequence of the reduction (5.2) but it turns out to be a dynamic result of the global geometry. In (5.12), we consider the sum over all the spinfoam boundary intertwiners. Different geometries might give different (non-regular) average boundary angles, for example, the case of the Δ_3 spinfoam graph [31]. To get an idea of the gain obtained with respect to the

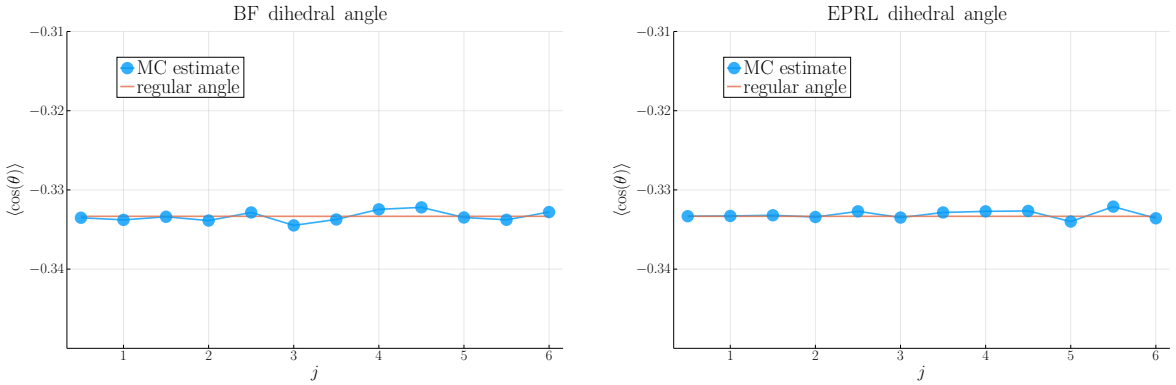


Figure 5.7: *Expectation values* (5.32) *of the dihedral angle operator* (5.39)*, averaged over all the* 20 *nodes of the spinfoam. The orange line shows the value of the cosine of the dihedral angle of a regular tetrahedron, which is* $\cos\left(\theta_{\text{regular}}\right) = -0.\bar{3}$.

blind summation, it is sufficient to consider that at spin $j = 6$. It is possible to compute the expectation value of an operator (for example, the dihedral angle operator) stably up to the third significant digit with several Monte Carlo iterations $N_{MC} \sim 10^7$, as in the 4-simplex model. The exact sum would require performing $(2j+1)^{20} \sim 10^{22}$ sums. Therefore, the Metropolis-Hastings algorithm adapted to the spinfoam formalism reduces the computation complexity by about 15 orders of magnitude.

The corresponding quantum spread $\Delta\cos(\theta)$ is shown in Figure 5.8. It turns out to be rapidly increasing for EPRL and slightly increasing for the BF model. As initially noticed in [77], this suggests that quantum fluctuations of the metric in the Lorentzian model are wide and are not suppressed in the asymptotic regime with few vertices. This, however, might be a simple consequence of the boundary state (5.2), which fixes the areas of the boundary triangles at the quantum level, implying that the boundary angles are quantum spread. The results suggest

[4]This step is justified a posteriori once it has been verified that the expectation value of the operator over all the nodes is identical.

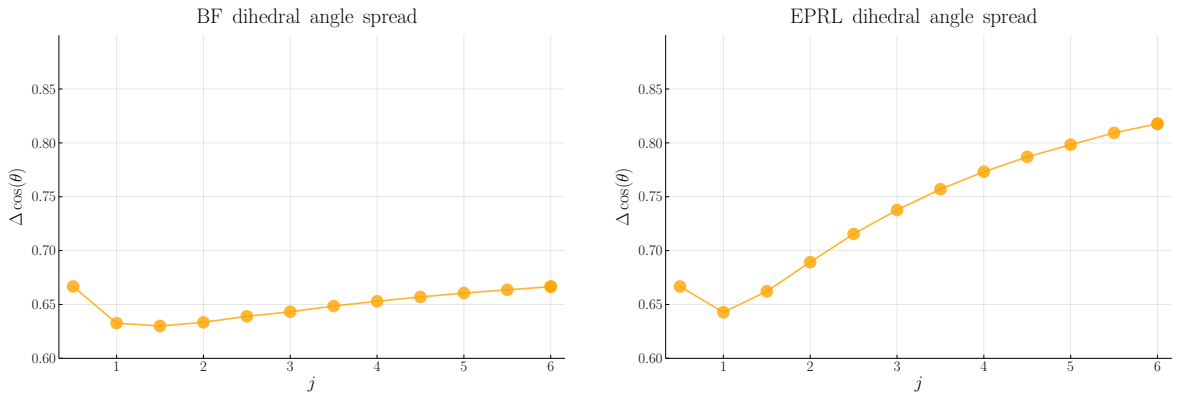


Figure 5.8: *Expectation values of the spread* (5.15) *for the dihedral angle operator* (5.39)*, averaged over all the* 20 *nodes of the spinfoam. In the EPRL model, the quantum spread increases faster as a function of the boundary spin* j*.*

that, even if it is not a regular triangulation, the star model is suitable to discretize a closed geometry as a simplicial manifold bounded by a topological 3-sphere. In fact, in addition to the similarity with the results obtained in [77], preliminary results on the 16-cell spinfoam model, which constitutes the second regular triangulation of the 3-sphere after the 4-simplex, exhibit a similar striking behavior [172]. The Gaussian distributions (5.36), measuring the statistical fluctuations in the Monte Carlo sampling, are shown in Figure 5.9. For the sake of clarity, we

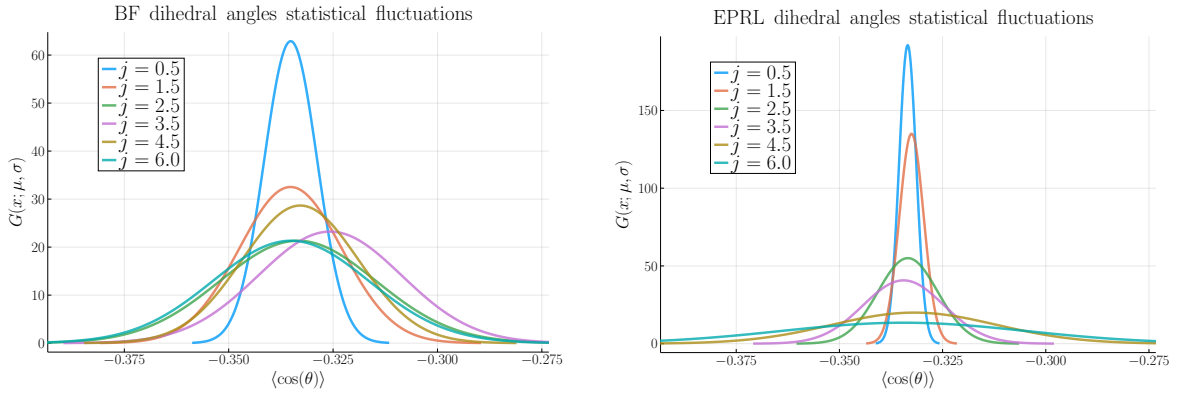


Figure 5.9: *Gaussian distribution* (5.36) *of the expectation values* (5.32) *of the dihedral angle operator* (5.39)*. We averaged over several runs, computing the (average) angle* $\mu_{<\cos\theta>}$ *defined on a single node and the corresponding standard deviation* $\sigma_{<\cos\theta>}$ *for each* j*.*

underline that the statistical fluctuations in Figure 5.9 were computed by averaging the expectation value (5.32) for the operator (5.39) on a single node over several runs, according to (5.36). In contrast, in Figure 5.7, we performed a further average on the 20 nodes of the spinfoam.
We computed the correlation functions (5.14) between dihedral angles for all the 190 independent node combinations of the spinfoam. The result is shown in Figure 5.10. For the dihedral angle operator (5.39), we found that correlations can assume two types of values, both for correlations between operators defined on nodes belonging to the same vertex and for different vertices as well, finding 4 different total possible numerical values. In the case of nodes on the

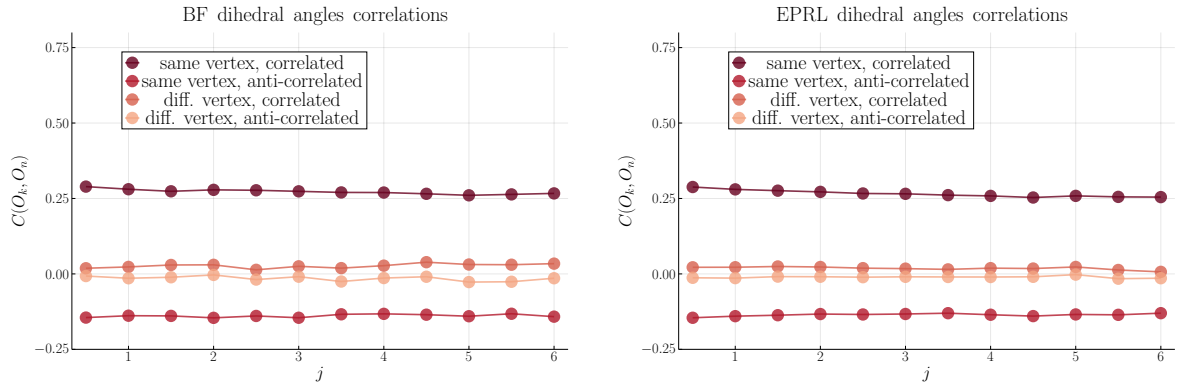


Figure 5.10: *Expectation values* (5.33) *of the correlations between dihedral angle operators* (5.39)*. The correlations computed respectively for the EPRL and the BF models are essentially indistinguishable.*

same vertex, we get the same correlations originally computed in [77]. In the second case, we observe that also angles between distant vertices can be (only) positively or negatively correlated, and the absolute value of the correlations is small compared to the first case. This agrees with the results on the entanglement entropy, as discussed in Section 5.5.2.
The numerical results show that the EPRL and BF models give rise to essentially indistinguishable dynamic correlations in the dihedral angle operator (5.39). This suggests that at least in the approximation described in Section 5.2, the $SU(2)$ topological model (typically much easier to compute) provides an excellent approximation for studying dynamical correlations.

The volume operator

There are two slightly different prescriptions for the volume operator in LQG. Here we follow the Rovelli-Smolin prescription in [11]. Since the general expression of the volume matrix elements in the spin-network basis is not trivial [1], here we limit ourselves to the equations in symmetric-reduced space of (5.1) in which all the spins have the same value j and the basis states are given by (5.2).
Let A be the $(2j+1)\times(2j+1)$ Hermitian matrix:

$$A = i \times \begin{pmatrix} 0 & -a_1 & 0 & 0 & \dots & 0 \\ a_1 & 0 & -a_2 & 0 & \dots & 0 \\ 0 & a_2 & 0 & -a_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \end{pmatrix} \tag{5.42}$$

where the coefficients a_k are defined as:

$$a_k = \frac{1}{4} \frac{k^2((2j+1)^2 - k^2)}{\sqrt{4k^2 - 1}}\,, \tag{5.43}$$

Let q_k be its real eigenvalues and $|q_k\rangle$ the corresponding eigenvectors. For each j the eigenvalues come in pairs of opposite signs, plus one 0 eigenvalue for j integer. The volume operator matrix can be written as:

$$\langle j, i_n | V | j, i'_n \rangle = \frac{\sqrt{2}}{3} (8\pi G\hbar\gamma)^{\frac{3}{2}} \sum_k \sqrt{|q_k|} \langle j, i'_n | q_k \rangle \langle q_k | j, i_n \rangle. \tag{5.44}$$

Contrary to the dihedral angle (5.39), the volume operator (5.44) is not diagonal in the basis (5.2). In terms of expectation values (5.30) and correlations (5.31), this involves recomputing the amplitude function for each element of the sampling, hence it is much slower than the corresponding diagonal evaluation (however it is still incomparably faster than blind summation (5.11) which would be required without the Monte Carlo approximation (5.28)). This means that computing the expectation value (5.30) in the case of the volume operator (5.44) for more than one node of the spinfoam takes too long. The expectation values of the volumes are shown in Figure 5.11, in which we neglected all the constant factors in the expression (5.44) since it simply corresponds to a homogeneous re-scaling of all points. It turns out that the scaling of the boundary volume, as a function of the boundary spin j, corresponds to that existing between the volume of a regular tetrahedron and the area of one of its faces. That is, $V \propto j^{3/2}$, as the eigenvalue of the area operator is proportional to $\sqrt{j(j+1)} \approx j$. We observe this in both BF and EPRL models, although the spectrum is not the same. An interesting feature of the volume operator spectrum is a systematic shift between integer and half-integer spins. That is, these are two slightly shifted curves. The quantum spread of the volume operator is shown

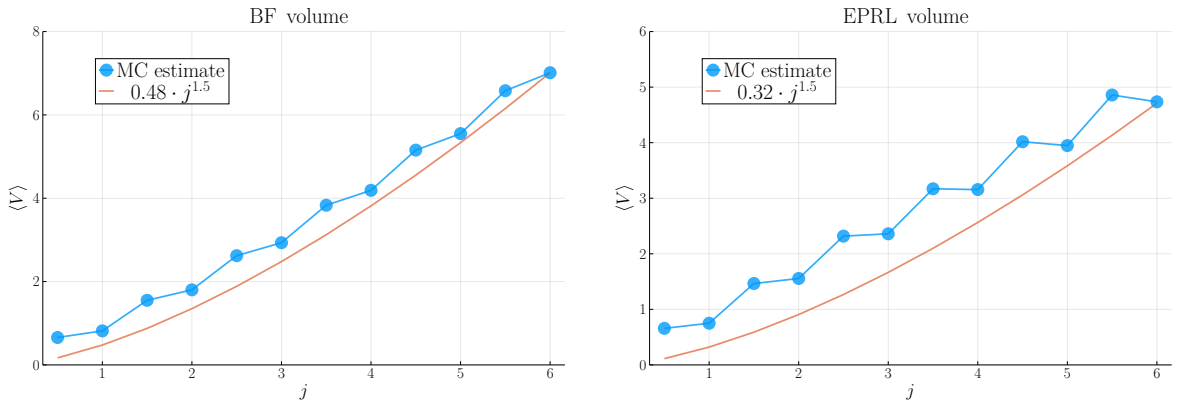


Figure 5.11: *Expectation values* (5.30) *of the volume operator* (5.44). *The orange line is proportional to the functional dependence between the volume of a regular tetrahedron and the area of one of its faces:* $V \propto j^{3/2}$.

in Figure 5.12. The shift between the curves corresponding to integer spins and half-integers is manifest. Differently with respect to the angle operator 5.8, for the volume operator, the quantum spread increases faster for BF rather than EPRL. The Gaussian distributions (5.36) are shown in Figure 5.13. The average values of the volumes used as mean in the Gaussian distributions (5.36) are the same plotted in Figure 5.11. The probability distribution for non-diagonal operators in equation (5.30) is not strictly positive. Interestingly, as shown in Figure 5.13, this does not affect the convergence. Volume correlations are shown in Figure 5.14. As for the angles, the volume correlations between nodes belonging to the same vertex are much higher than those between different vertices. For the volumes, the latter appears to be essentially zero. It is interesting to notice that, contrary to what happens with the angles, there is only one type of correlation between volumes. That is, only exists one common value for all the correlations between volumes of tetrahedra on the same vertex, and the same is true for non-adjacent tetrahedra.

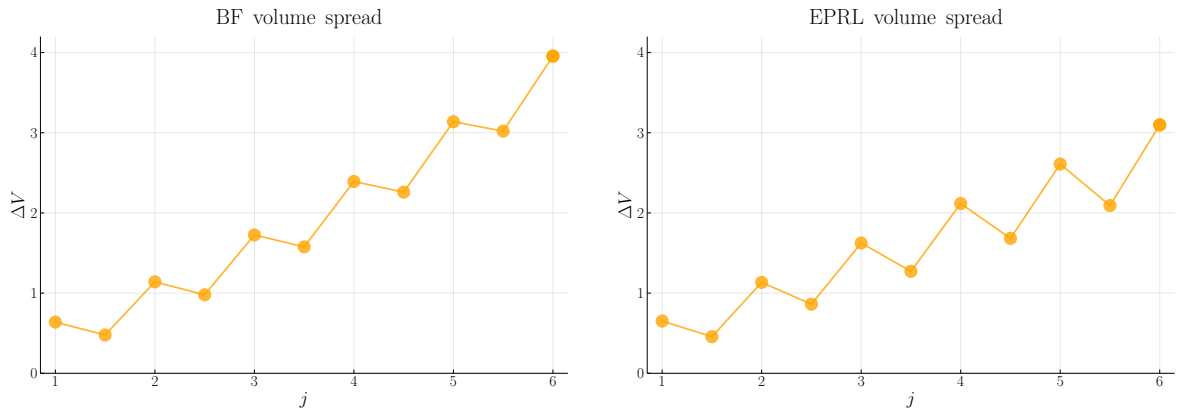


Figure 5.12: *Expectation values of the quantum spread* (5.15) *for the volume operator* (5.44). *As in Figure 5.7, it is evident that the spectrum of the volume operator gives rise to two distinct curves for integer and half-integer spins, which turn out to be shifted with respect to each other.*

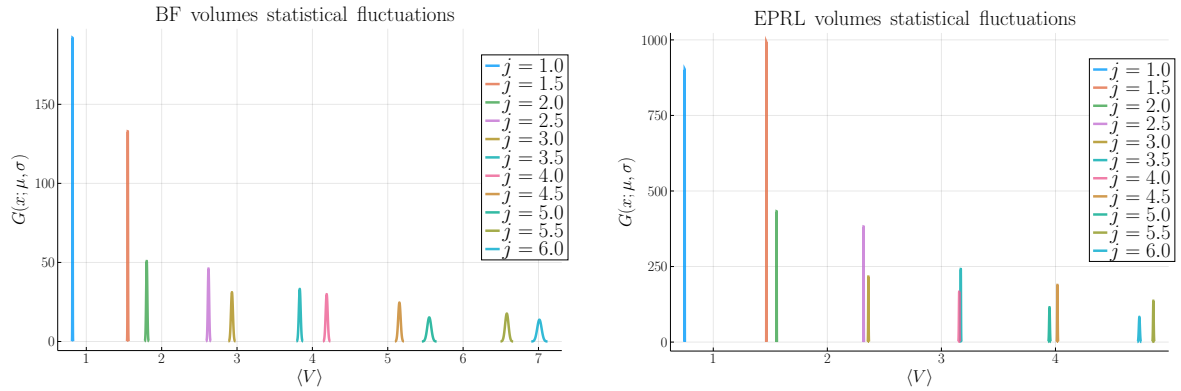


Figure 5.13: *Gaussian distribution* (5.36) *of the expectation values* (5.30) *of the volume operator* (5.44). *We averaged over several runs, computing the (average) volume* $\mu_{<V>}$ *defined on a single node and the corresponding standard deviation* $\sigma_{<V>}$ *for each* j.

5.5.2 Numerical results: entanglement entropy

We now discuss the results of the computation of the entanglement entropy between different nodes, viewed as quantum subsystems of the whole graph. The entanglement entropy is the most relevant quantity to study the degree of correlation between operators, as it gives the mutual information between different subsystems [173, 174]. In the topological BF model, the way in which we defined the boundary state (5.4) in Section 5.2 coincides with the definition of the Bell-Network states, whose entanglement entropy is computed in [174] for different combinations of graphs and subsystems.
In general, a quantum system composed of two subsystems A and $\bar{A}$ has a Hilbert space given by the tensor product:

$$\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_{\bar{A}} \,. \tag{5.45}$$

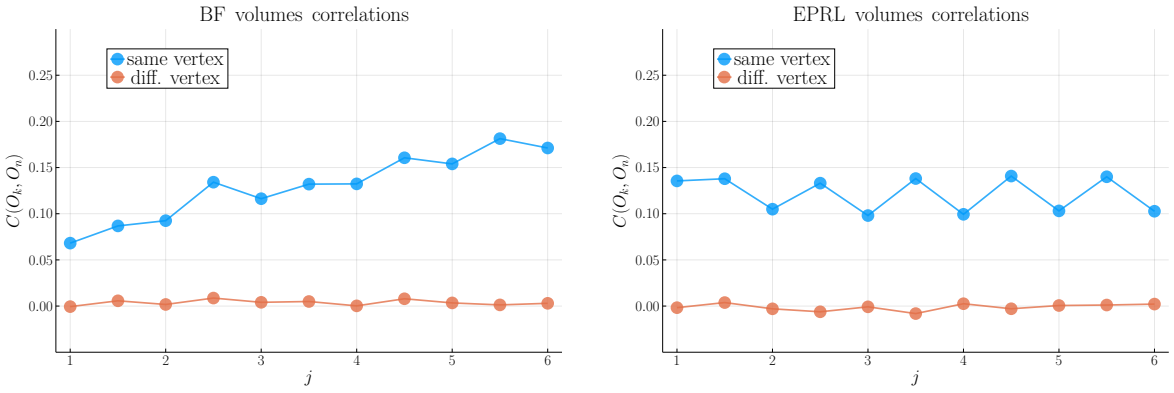


Figure 5.14: *Expectation values* (5.31) *of the correlations between volumes. While in the BF model, the correlations between volumes on the same node seem to slowly increase with the boundary spin j, in EPRL, this does not happen. Unlike angles (see Figure 5.10), all nodes on the same vertex have the same correlation, while nodes on different vertices are essentially uncorrelated.*

Given the boundary state $|\psi_0\rangle$ in the Hilbert space (5.1), the normalized reduced density matrix of the subsystem A is defined by the partial trace over its complement $\bar{A}$:

$$\rho_A = \frac{1}{Z} \mathrm{Tr}_{\bar{A}} \ |\psi_0\rangle\langle\psi_0| \,. \tag{5.46}$$

The entanglement entropy of the subsystem A is then defined as the von Neumann entropy of the reduced density matrix:

$$S_A = -\,\mathrm{Tr}\,(\rho_A \log \rho_A) \,. \tag{5.47}$$

Using the expression (5.4) in (5.46), after some algebraic manipulations the normalized reduced density matrix (5.46) can be written as:

$$\rho_A = \frac{1}{Z} \sum_{\{i_a\}} \sum_{\{i'_a\}} M\left(j, \{i_a\}, \{i'_a\}\right) \bigotimes_a |j, \{i_a\}\rangle\langle j, \{i'_a\}| \,, \tag{5.48}$$

where $a \in A$. The coefficients $M\left(j, \{i_a\}, \{i'_a\}\right)$ are defined by tracing over the intertwiners $i_{\bar{a}}$ in the complement subsystem $\bar{A}$:

$$M\left(j, \{i_a\}, \{i'_a\}\right) = \sum_{\{i_{\bar{a}}\}} A\left(j, \{i_a\}, \{i_{\bar{a}}\}\right) A\left(j, \{i'_a\}, \{i_{\bar{a}}\}\right) \,. \tag{5.49}$$

We replaced the sum over the full set $\{i_n\}$ with $\{i_a\}$, namely, the intertwiners involved in the partition (5.45). By introducing the Monte Carlo approximation (5.28), the expression for the density matrix becomes:

$$\rho_A \approx \frac{1}{N_{MC}} \sum_{[i_n]} \sum_{\{i'_a\}} \frac{A\left(j, \{i'_a\}, [i_n]\right)}{A\left(j, [i_n]\right)} \bigotimes_a |j, [i_a]\rangle\langle j, \{i'_a\}| \,. \tag{5.50}$$

The notation $[i_a]$ is a label for the set of intertwiners draws $[i_n]$ in which the nodes belonging to the subsystem A have a value compatible with the position in the density matrix, and the

meaning of the amplitude $A(j, \{i'_a\}, [i_n])$ should be clear by looking at (5.10). Notice that the density matrix is symmetric and $\mathrm{Tr}(\rho_A) = 1$. The entropy (5.47) becomes:

$$S_A \approx -\nu_i \log \nu_i \,, \tag{5.51}$$

where ν_i is the i-th eigenvalue of the density matrix (5.50). Notice that the computational time of the density matrix considerably increases as the number of nodes N_A in subsystem A grows, as the matrix (5.50) has dimensions $(2j+1)^{N_A} \times (2j+1)^{N_A}$. Furthermore, at a fixed number of Monte Carlo iterations N_{MC}, the statistical fluctuations increase along with N_A since each matrix element is sampled by a set that becomes smaller and smaller. For these reasons, we introduced a second multi-threading parallelization scheme using multiple machines in the numerical computation of the density matrices. Each node computed the density matrix using a different Markov chain by distributing the calculation over multiple CPUs. The same hybrid parallelization scheme can be used to speed up the calculation of non-diagonal operators (5.30) and corresponding correlations (5.31).

Subsystem with 1 and 2 nodes

The values of the entanglement entropy as a function of the boundary spin j are shown in Figure 5.15 for the subsystem A in the partition (5.45) consisting in 1 and 2 nodes. We used the parameters listed in E.2 for the data in Figure 5.15. Notice that, by choosing a common value j for all the links of the star spinfoam (see Figure 5.5), we cannot distinguish correlations between nodes belonging to distinct vertices whether or not the same link connects them. For example, labeling the nodes according to the notation illustrated in Figure 5.5, computing the entanglement entropy for the subsystems $A = \{C235, A235\}$, $A' = \{C235, D134\}$ and $A'' = \{C235, B134\}$ we obtain the same numerical value. The value of the EPRL entropy for the

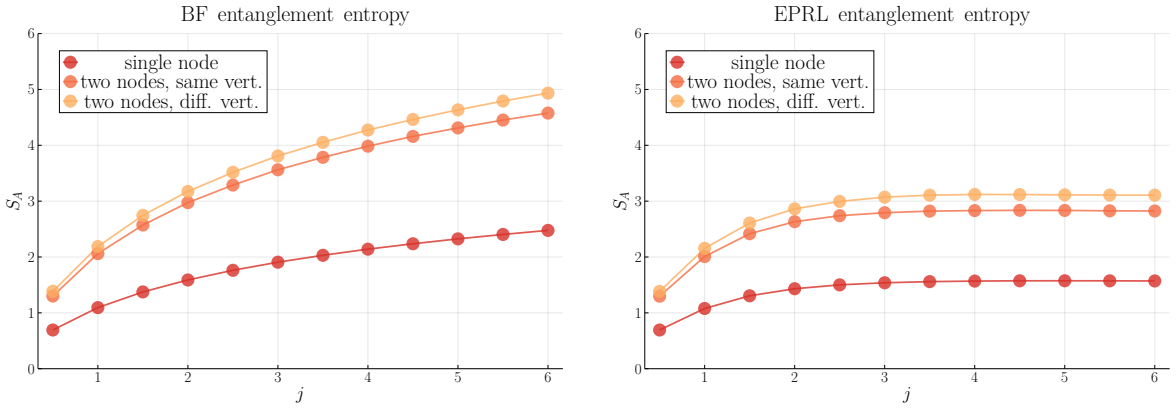


Figure 5.15: *Values of the entanglement entropy* (5.51) *for different subsystems A in the partition* (5.45) *with* $N_A = 1$ *and* $N_A = 2$ *for the star spinfoam in Figure 5.5.*

subsystem consisting of a single node is similar to the value obtained in [77] for the single vertex graph. It is interesting to notice that the EPRL entropy of all considered subsystems tends asymptotically to a constant value as the boundary spin j increases. The statistical fluctuations of entropy are shown in Figure 5.16. Notice that, with the same number of iterations N_{MC},

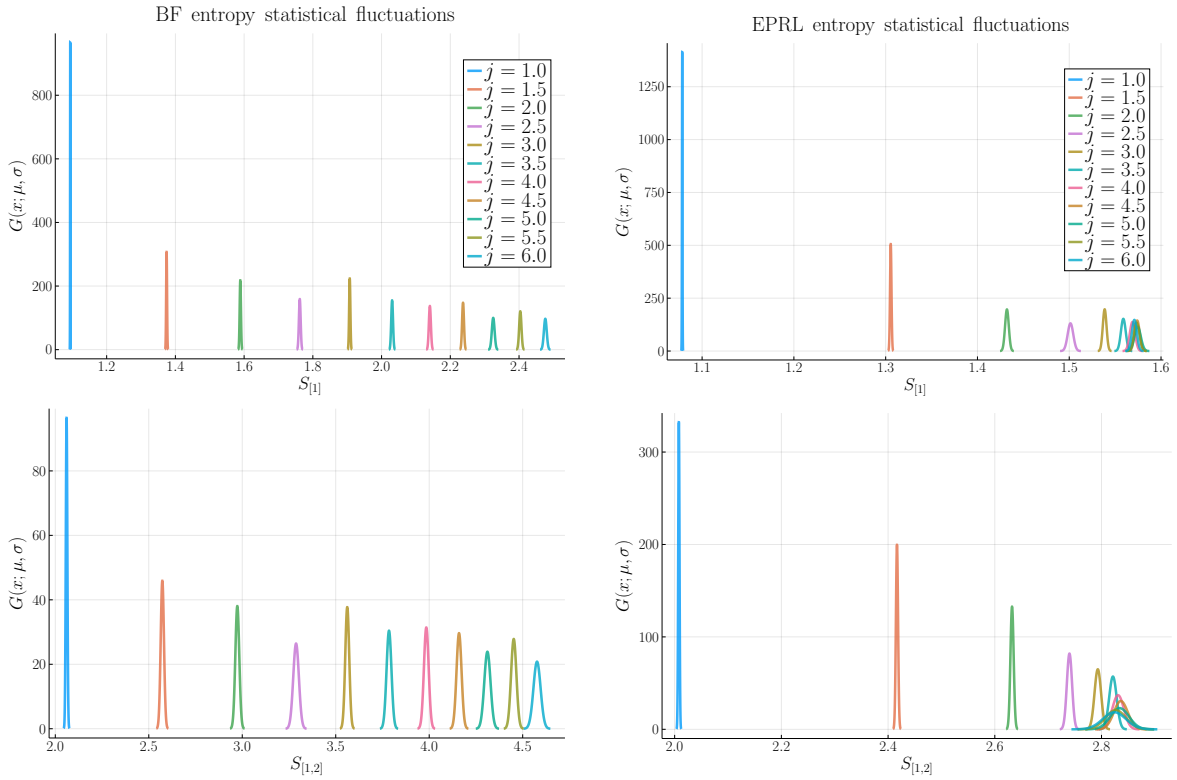


Figure 5.16: *Gaussian distribution* (5.36) *of the entanglement entropy* (5.51) *for the subsystems in Figure 5.15. We proceeded as in the case of angles (Figure 5.9) and volumes (Figure 5.13). Top panel: fluctuations for the subsystem with* $N_A = 1$. *Bottom panel: fluctuations for the subsystem with* $N_A = 2$.

statistical fluctuations in Figure 5.16 are more significant for $N_A = 2$ for the abovementioned reasons. The fact that the entropy of a subsystem composed of nodes on different 4-simplices is slightly greater than the one of the subsystem composed of nodes on the same 4-simplex is connected to the smaller value of the correlations between nodes belonging to different four simplices. To discuss this point, we first define the mutual information $I(k,m)$ between two generic nodes k and m as:

$$I(k,m) = S_k + S_m - S_{km}\,, \tag{5.52}$$

where S_{km} is the entropy of the subsystem A composed by the nodes k and m. It turns out that the mutual information (5.52) between k and m actually provides a bound on correlations [173, 174]:

$$\frac{\left(\langle O_k, O_m\rangle - \langle O_k\rangle\langle O_m\rangle\right)^2}{2\|O_k\|^2\|O_m\|^2} \leq I(k,m)\,. \tag{5.53}$$

where $\|O_k\|$ is the norm of the local operator O on the node k. Therefore, equations (5.52) and (5.53), along with the results in Figure 5.15 imply that the correlation function (5.14) has a more stringent upper bound for the subsystem containing nodes defined on different 4-simplices.

Subsystem with four nodes

For completeness, we show in Figure 5.17 the values computed for the entropy of subsystem A composed of 4 adjacent nodes, that is, four nodes on the same 4-simplex $A = \{i_{C235}, i_{C234}, i_{C345}, i_{C245}, \}$. For this computation, we limited the analysis to a maximum value $j = 5$ since the computational cost is significantly higher than the other calculations reported in this paper. Unlike the parameters in E.2, for this specific calculation, we set $N_{MC} = 10^7$ for $j = 0.5, 1, \ldots 5$, averaging over 17 independent runs both for BF and EPRL. Following the hybrid parallelization scheme discussed at the beginning of this Section, we used 17 processes, each with 64 CPUs, for 1088 CPUs. With this configuration, the total computation time for the data in Figure 5.17, including the sampling of the intertwiners draws, took about three days.

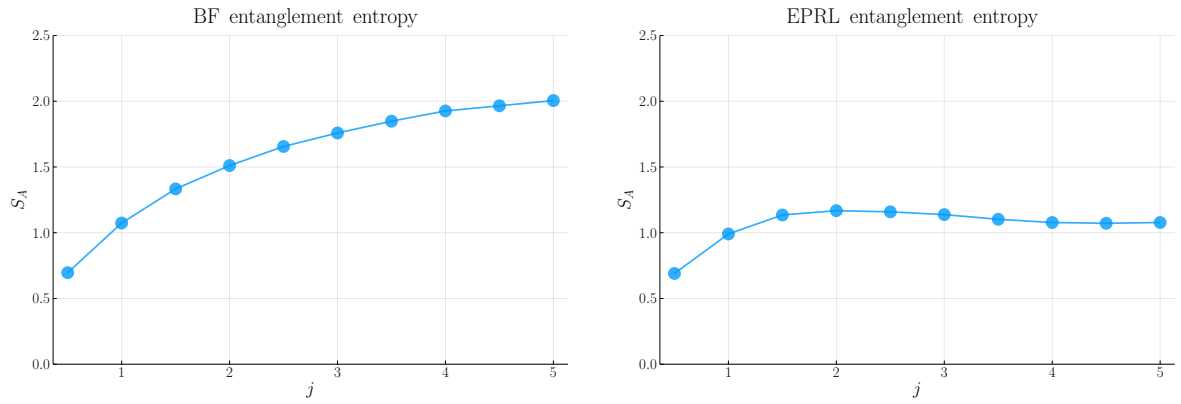


Figure 5.17: *Values of the entanglement entropy* (5.51) *for the subsystem A in the partition* (5.45) *with* $N_A = 4$. *All the nodes in A belong to the same 4-simplex.*

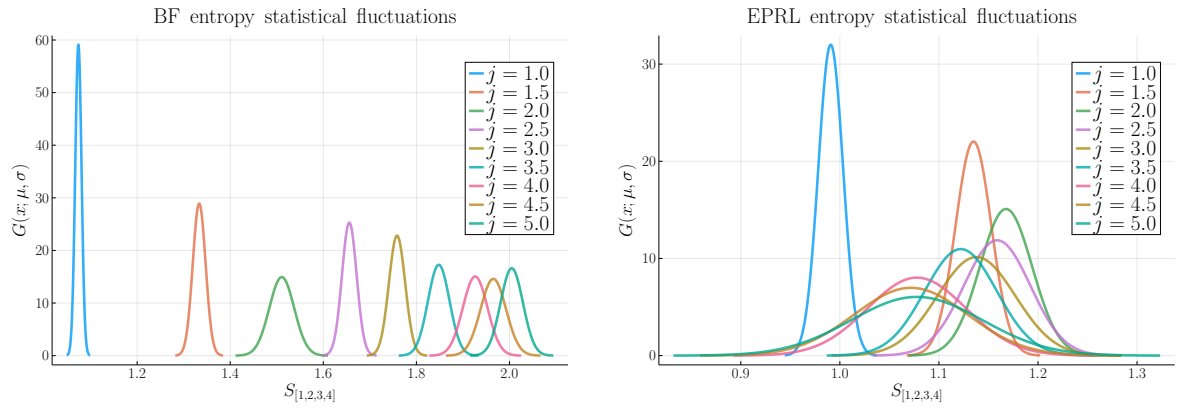


Figure 5.18: *Gaussian distribution* (5.36) *of the entanglement entropy* (5.51) *for the subsystem in Figure 5.17. We used the same number of iterations* $N_{MC} = 10^7$ *for BF and EPRL, averaging over 17 runs.*

Conclusions

In this paper, we combined the Metropolis-Hastings algorithm [148] with recently developed high-performance codes in LQG [48, 47] to compute the expectation value and correlation functions of operators over large spinfoam graphs in the low spins regime. After testing the method, we applied it to the computations of boundary geometrical observables, correlation functions, and entanglement entropy in a spinfoam model with 20 boundary nodes, obtained as a refinement of the 4-simplex graph. We investigated both the EPRL and the BF models. Our results show that the BF and, more significantly, the EPRL model have a well-defined behavior under the refinement of the boundary graph. The computed boundary geometry agrees in terms of expectation value with the geometric interpretation of the operators. Furthermore, we found that correlations are present in neighboring patches but decay sharply when moving to patches that belong to different vertices. This opens the way to studying spinfoams composed of many vertices glued together. We also showed that the dynamical correlations between boundary operators in the BF and EPRL models are surprisingly similar in our case study. In contrast, the entanglement entropy shows a significant difference.

Our work provides essential hints on the well-definiteness of spinfoam refinement. By refinement, we mean a step toward a less radical truncation. This is implemented by replacing a given triangulation with a finer one. A single boundary graph allows the construction of a quantum state that describes *the complete evolution* of a (discretized) geometry in the (intrinsic, discrete) time provided by the area observables (here assumed equal). When stepping from a given triangulation to a finer one, the same overall continuous *spacetime* geometry is truncated as a different discretized geometry that, of course, has single cells with a different volume and single triangles with a separate area when, say, the overall volume or the overall size of a whole sphere are given. By stability under refinement, we mean that the quantum state computed with the first triangulation can be seen as a good first approximation to the one calculated with the finer one since geometrical operators scale as a function of boundary spins in the same manner before and after the refinement.

The method presented in this paper can be applied to spinfoam models with Lorentzian or Euclidean signatures and to compute bulk observables. An interesting perspective would be to compare the results obtained in the spinfoam-like path integral expansion formalism [175] with the ones in the full spinfoam one. This would give interesting insights into the canonical-covariant relation. Numerical methods currently allow performing computations using spinfoams with a much richer bulk structure than the one considered in this paper. For example, the study of infrared divergences [71, 133] or the analysis of the Δ_3 and Δ_4 triangulation [9]. The methods described in this paper can also be applied to different choices of boundary states. However, one has to deal with fluctuating sampling probabilities for complex coherent states, and different Monte Carlo techniques might be more effective. The proposed approach provides a needed complement to already existing numerical techniques in covariant LQG [57]. It is effective in the regime of low spins quantum numbers with many degrees of freedom, for which the other existing methods are not tailored. The next step in the development of this work consists of applying the algorithm presented here to study the correlation functions defined on a spinfoam model with 16 cells on the boundary: this is the next regular triangulation of the 3-sphere after the 4-simplex considered in [77]. This model is studied in [172].

Chapter 6

Primordial fluctuations from quantum gravity: 16-cell topological model

Statement of co-authorship

This Chapter presents the only research project in this Ph.D. thesis that has not yet been published in a peer-reviewed journal. The authors are myself, Francesco Gozzini, and my Ph.D. advisor, Francesca Vidotto. It chronologically follows the line of research started with [77] and [78], discussed in Chapter 5. This work has a very similar development to the previous Chapter. I performed most of the work as a final assignment of the Scientific Computing course at Western University, taught by Professor Karttunen, in the Winter Term of 2022. After that, this project was temporarily shelved. Although the published version may differ, this draft has been completed to constitute a Chapter of this thesis. I took care of the numerical computations, following the route set initially by Francesco Gozzini. I drafted all sections except for the introduction and realized plots and images. As for the previous Chapter, I estimate the percentage of the total work conducted by me between 40% and 60%.

6.1 Introduction

The early universe is believed to have emerged from a phase dominated by quantum gravitational effects. A quantum theory of gravity is needed to describe this phase and to derive the initial boundary data of the cosmological standard model. An approach to address this derivation in the covariant Loop Quantum Gravity context has been proposed in [77]. It is based on two hypotheses. First, the initial boundary data of the cosmological standard model can be computed as the result of a quantum gravitational transition amplitude from nothing, as initially suggested by Hartle and Hawkings in the (different) context of Euclidean quantum gravity. This state has been argued to be relevant both in the case of a genuine Big Bang and a Big Bounce [152, 153, 158, 159, 160, 161]. Second, truncating the theory to a finite number of degrees of freedom, capturing only a few lowest frequency modes at each cosmological time, represents a good first-order approximation in this regime. These hypotheses allow us, in principle, to compute the initial state using the transition amplitudes of the theory. In particular, fluctuations and correlations of the resulting quantum state can be calculated and compared

with those of the quantum field theory's initial state of standard cosmology.

In practice, the calculation is complicated. Initial numerical investigations have been developed in [78], relying on a drastic truncation, which can be interpreted as triangulation of the spatial geometry of a closed universe into five adjacent tetrahedra. A severe limitation of this truncation is that all distinguishable regions of space are adjacent, hiding any dependence on correlations from spatial distance. In this paper, we take a step towards removing this limitation. We consider a finer truncation, corresponding to a triangulation of the geometry of a closed universe into sixteen tetrahedra. In this triangulation, the distinguishable regions of space are not all adjacent, and we can analyze the dependence of the quantum correlations between different regions as a function of their separation.

The paper focuses on the setting up of the combinatorics and on showing that the calculations are, in principle, doable numerical but do not perform the calculations with the full covariant quantum gravity amplitudes, which will be explored in the future. Instead, we use the unphysical simplified version of these amplitudes provided by the $SU(2)$ BF model. This is a rather drastic simplification of the amplitudes, and therefore, we present the results here more as a proof of concept than an actual test of the theory. While there are regimes where the BF amplitudes are similar to the physical ones, this is not generally the case.

In the next Chapter, we describe the relevant aspects of the geometry of a 16-cell. Then, we set the calculation of the states' properties resulting from the transition to nothing. Finally, we give the numerical results showing the properties of this state.

6.2 The 16-cell geometry

A 16-cell is the name of a regular polytope (the high-dimension generalization of a polyhedron) in four-dimensional Euclidean space. It is defined as the convex hull of the 8 points with Euclidean coordinates $(0,0,0,\pm1),(0,0,\pm1,0),(0,\pm1,0,0),(\pm1,0,0,0)$. It is also called hexadecachoron and is analogous to the octahedron in three dimensions, namely the convex span of the points $(0,0,\pm1),(0,\pm1,0),(\pm1,0,0)$, and to the square in two dimensions, namely the convex span of the points $(0,\pm1),(\pm1,0)$. Its boundary is a regular triangulation of a 3-sphere into 16 tetrahedra. The triangulation has 32 triangular faces, 24 edges, and the 8 vertices. This is the second of the three regular triangulations of the 3-sphere: the first being the 4-simplex and the third being the 600-cell made by 600 tetrahedra. The dual of this triangulation is the surface of a hypercube (which has 16 vertices).

An intuitive visualization of the 16-cell triangulation is as follows. Start with two points, say a and b (See Figure 6.1). Adding two extra points, say c and d, each connected to a and b, we obtain a 1-sphere (a circle) split into 4 segments S_1....S_4 triangulated into a square. By connecting these segments to two external points, say e and f, we obtain 8 triangles that triangulate a 2-sphere into an octahedron, as in Fig. 6.1. By adding two more extra points, say g and h, connecting each to these 8 triangles, we obtain 16 tetrahedra that triangulate a two-sphere. The two skeleton of the resulting triangulation is depicted in Figure 6.1 (with a different numbering of the vertices).

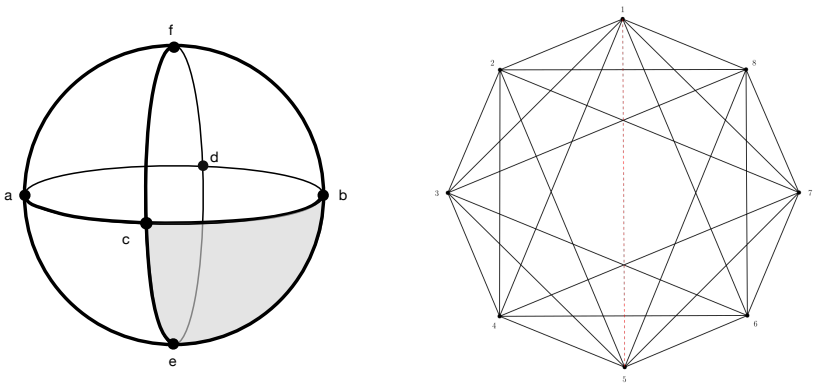


Figure 6.1: *Triangulation of a 2-sphere with 8 triangles, one of which is highlighted in grey (left). Notice that this can be obtained by adding points 5 and 6 to a triangulation of a 1-sphere (a circle) with four segments. By repeating the same step one dimension higher, namely adding points 7 and 6 and connecting them to the 8 triangles, we obtain the 16-cell triangulation of a 3-sphere (right, with different numbering).*

It is possible to triangulate the four-dimensional 16-cell polytope by splitting it into eight 5-simplices. Figure 6.2 illustrates this triangulation. This also shows how a spinfoam that defines an amplitude for this geometry can be computed [1]. The spinfoam has eight 4-simplices glued on six internal faces. The eight points in the triangulation are labeled using numbers $1, 2 \ldots 8$. Consequently, in the spinfoam diagram, each 4-simplex is labeled with five numbers, each tetrahedron with four numbers, and each face with three numbers. In Figure 6.2, only the bulk and boundary tetrahedra are labeled so as not to clutter the picture. As we discuss below, however, we shall not need the explicit form of this spinfoam.

6.3 Expectation values and correlations

6.3.1 Boundary state

Let Γ be an LQG graph with L links and N nodes. Then, the Hilbert space associated with Γ is:

$$\mathcal{H}_\Gamma = L_2\left[SU(2)^L/SU(2)^N\right] . \tag{6.1}$$

The states $|j_l, i_n\rangle$ form the spin network basis in $\mathcal{H}_\Gamma$, where $n = 1 \ldots N$, $l = 1 \ldots L$ (isolated indices range over multiple values). Half-integer spins constitute the set j_l, and i_n is an intertwiner set. Each intertwiner i is a basis element of the invariant subspace of the tensor product of 4 $SU(2)$ representations at the corresponding node. Here, we are interested in the Hilbert $\mathcal{H}_\Gamma$ defined by the graph formed by the two-skeleton of the triangulation of the boundary of the 16-cell. Hence $N = 8$, $L = 24$. From now on, we omit the Γ subscript to not weigh down the notation.

Following [77, 78], we fix all the spins to be equal so that $j_l = j$ for all the links of the boundary. Only one common spin is attached to all the links in this subspace. This define a subspace of $\mathcal{H}_\Gamma$ of the form

$$H_\Gamma = \oplus_j \otimes_{n=1}^{16} \mathrm{Inv}[V_j \otimes V_j \otimes V_j \otimes V_j] , \tag{6.2}$$

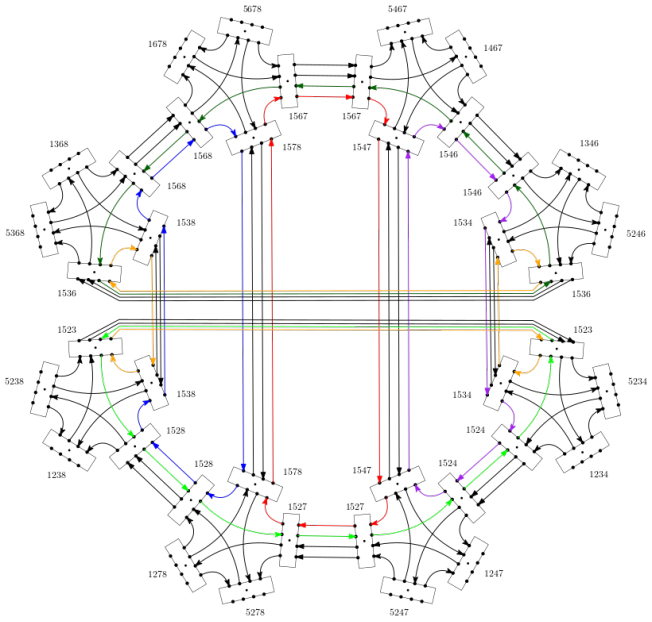


Figure 6.2: *The spinfoam, with eight vertices and six internal faces (highlighted with different colors). The labels refer to the points in the triangulation. Only the edges are labeled so as not to clutter the picture.*

where V_j is the spin-j representation of $SU(2)$ and Inv denotes the $SU(2)$ invariant part of the tensor product. We write a boundary spin network state as:

$$|j, i_n\rangle = |j, i_1\rangle \otimes \cdots \otimes |j, i_{16}\rangle \,, \tag{6.3}$$

where i_n are intertwiners in $\mathrm{Inv}[V_j \otimes V_j \otimes V_j \otimes V_j]$. Since all spins are equal to j, triangular inequalities constrain every intertwiner to assume only integer values between 0 and $2j$.

We are interested in the Hartle-Hawking state $|\psi_0\rangle$ defined in [77] and further studied in [78]. In the Hilbert space (6.1), it is defined as:

$$|\psi_0\rangle = \sum_{j,i_n} W(j, i_n)\, |j, i_n\rangle \,, \tag{6.4}$$

where $W(j, i_n) = \langle 0|j, i_n\rangle_{Ph}$ is the LQG physical transition amplitude from the sstatedefined by $j = 0$. This amplitude is associated with the transition from nothing to a boundary spin network state $|j, i_n\rangle$ with a given j. The state (6.4) can be interpreted as the natural state that is projected out of the vacuum state by the dynamics. In the $|j, i_n\rangle$ basis, the state reads

$$\psi_0(j, i_n) = W(j, i_n) \,. \tag{6.5}$$

In quantum gravity, evolution is relative with respect to one of the physical variables. It is natural to use j as the independent variable and interpret it as a proxy for a time variable related to the overall size of the spacial 3-sphere. The i_n are then interpreted as coding the local distortions of the geometry. Then we can interpret $\psi_0(j, i_n)$ in analogy with the way we usually interpret $\psi(t, x)$: namely, taking the first variable as labeling an evolution and the value

of the state fixed value of this variable as describing the quantum state that given moment of the evolution. In the following, we shall do so, thus studying

$$|\psi_0(j)\rangle = \sum_{i_n} W(j, i_n)\, |j, i_n\rangle. \tag{6.6}$$

at different values of j. The sum in (6.6) is over all the intertwiners in the set i_n, giving a total of $(2j+1)^{16}$ elements entering the sum. In Section 6.4.2, we describe how to perform calculations with the state (6.6) using Monte Carlo methods.

6.3.2 Computing operators

We define the expectation value on the state (6.6) of a local operator O over node k as:

$$\langle O_k \rangle \equiv \frac{\langle \psi_0 | O_k | \psi_0 \rangle}{\langle \psi_0 | \psi_0 \rangle} \,. \tag{6.7}$$

The operator O_k in (6.7) acts on the boundary k-th node of the Hilbert space (6.1). Using the orthogonality of the spin-network states (6.3)

$$\langle j, i'_n | j, i_n \rangle = \delta_{i'_1, i_1} \ \dots \ \delta_{i'_N, i_N} \,, \tag{6.8}$$

We define the normalization factor in the denominator of (6.7) as:

$$\langle \psi_0 | \psi_0 \rangle = \sum_{i_n} W(j, i_n)^2 \equiv Z \,. \tag{6.9}$$

For a diagonal operator in the spin-network basis (6.3), the expectation value (6.7) can be written as:

$$\langle O_k \rangle = \frac{1}{Z} \sum_{i_n} W(j, i_n)^2 \, \langle j, i_k | O_k | j, i_k \rangle \,. \tag{6.10}$$

In (6.10), the index k is fixed as it appears on both sides of the equations, while index n ranges from 1 to N. The expectation value of the product of two operators (on the node k and k' respectively) turns out to be:

$$\langle O_k O_{k'} \rangle = \frac{1}{Z} \sum_{i_n} W(j, i_n)^2 \, \langle j, i_k | O_k | j, i_k \rangle \langle j, i_{k'} | O_{k'} | j, i_{k'} \rangle \,. \tag{6.11}$$

We can now define the quantum spread of a local operator as:

$$\Delta O_k = \sqrt{\langle O_k^2 \rangle - \langle O_k \rangle^2} \,. \tag{6.12}$$

Finally, we write the (normalized) correlation between two local operators as follows:

$$C(O_k, O_{k'}) = \frac{\langle O_k O_{k'} \rangle - \langle O_k \rangle \langle O_{k'} \rangle}{(\Delta O_k)\,(\Delta O_{k'})} \,. \tag{6.13}$$

6.3.3 The 16-cell spinfoam amplitude

We now take a drastic simplification. Instead of the full covariant EPRL amplitude $W(j, i_n)$, we use the $SU(2)$ topological BF amplitude. This simplifies the numerical calculation substantially and allows us to disregard a potential difficulty problem: the six internal faces form a bubble in the spinfoam. (Bubbles have been recently studied numerically [71, 10, 72, 138].) In principle, we must sum infinite terms to compute the amplitude. Introducing a cut-off on the spins labeling the bulk faces would allow us to solve this issue, dealing with a finite number of internal face configurations. Then, we should compute all the required vertex amplitudes and perform the 12 contractions over bulk intertwiners for each configuration of internal faces. The 16-cell spinfoam amplitude at fixed values of boundary intertwiners (all boundary spins being fixed) is obtained by summing all such contributions. Nevertheless, it would still be too challenging to compute numerically all the 16-cell amplitudes we need to calculate the expectation values (6.10) and correlations (6.13). To investigate such quantities, we use the BF topological model. As discussed below, this solves two problems at the same time. Studying the 16-cell geometry using other non-topological spinfoam models (such as the EPRL theory) is left for future work.

Thanks to the topological invariance of BF theory, we can compute the amplitude by looking only at the boundary. This essentially reduces to computing a Wigner 48j-symbol (16 intertwiners plus 32 spins) because the amplitude $W(j, i_n)$ is simply the evaluation of the corresponding $SU(2)$ spin network. The steps for reducing the 48j symbol to simpler 21j symbols (each one decomposed in 6j and 9j Wigner symbols) are described below. This simplification allows us to avoid the introduction of an artificial cut-off on the spins, labeling the internal faces in the bulk. The expression of the amplitude is greatly simplified, making it feasible to compute a considerable amount of them.

We now describe the steps required to compute the 16-cell spinfoam amplitude. We start from the triangulation in the right panel of Figure 6.1. We insert an internal segment between points 1 and 5 (arbitrarily chosen) to derive the corresponding dual triangulation, as reported in Figure 6.2. The spinfoam has eight vertices and six internal faces. Each vertex in the spinfoam is labeled with 5 points in the triangulation (each vertex is dual to a 4-simplex), each edge with 4 points (each edge is dual to a tetrahedron), and each face with 3 (each face is dual to a triangle). All tetrahedra in the triangulation labeled with points 1 and 5 are in the spinfoam bulk, while the remaining ones (marked with points 1 or 5) are on the boundary. The tetrahedra in the triangulation sharing 3 points are glued on a face in the spinfoam. For example, the two tetrahedra labeled with points 5678 and 5278 are connected by a link on the boundary of the spinfoam.

We first perform the integrals over $SU(2)$ in each one of the 16 boundary tetrahedra using relation (F.11), defined in F.2. Applying (F.11), we have an intertwiner on each boundary tetrahedron of the spinfoam. Since the BF topological invariance allows computing the amplitude by focusing on the boundary, we perform the integrals over the 6 internal faces of the 16-cell spinfoam. We are left with the diagram reported in the top left panel of Figure 6.3, where each boundary intertwiner is represented with a brown dot. A Wigner 48j symbol constitutes the boundary of the 16-cell spinfoam amplitude. To compute it efficiently, we write the symbol as contractions of smaller Wigner 21j symbols. The definition of the 21j symbol split as the

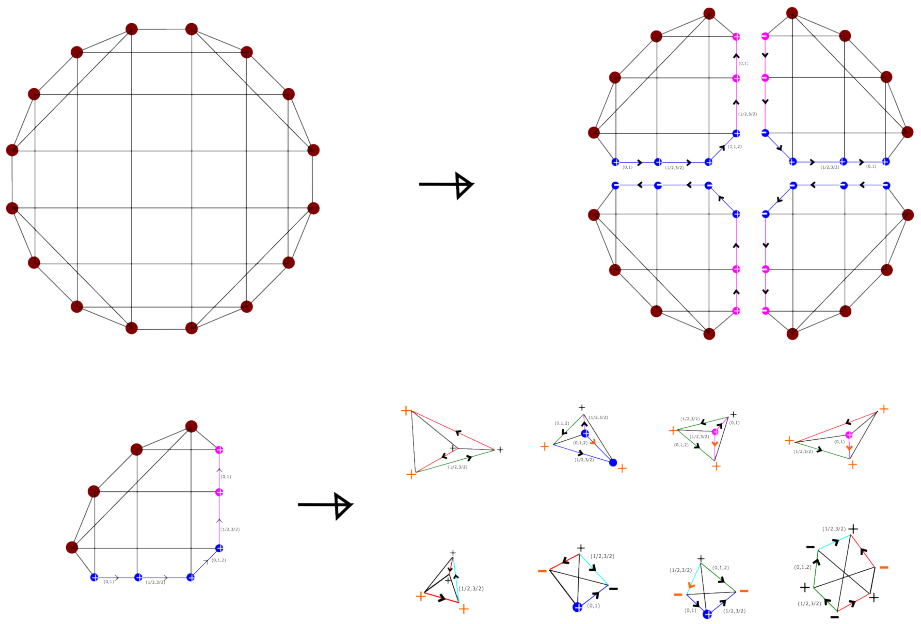


Figure 6.3: *Top: The boundary of the 16-cell spinfoam amplitude is a 48j Wigner symbol, split as the contraction of four 21j symbols. Bottom: Each 21j symbol is reduced to the sum of the products of seven 6j symbols and one 9j symbol.*

contraction of 6j and 9j symbols is bulky. Therefore it is reported in F.2 (see (F.14)). We take advantage of it to write the "north and south" amplitudes associated with the boundary of the top right panel in Figure 6.3. These are given by the contraction of 21j symbols along the vertical purple spins:

$$W_N = \sum_{p_1,p_2} \{21\,j\}\,(j, i_1, i_2, i_3, i_4, b_1, b_2, b_3, p_1, p_2)\,\{21\,j\}\,(j, i_{16}, i_{15}, i_{14}, i_{13}, b_5, b_4, b_3, p_1, p_2)\,d_{p_1} d_{p_2}\ ,$$

$$W_S = \sum_{p_1,p_2} \{21\,j\}\,(j, i_8, i_7, i_6, i_5, b_1, b_2, b_3, p_1, p_2)\,\{21\,j\}\,(j, i_9, i_{10}, i_{11}, i_{12}, b_5, b_4, b_3, p_1, p_2)\,d_{p_1} d_{p_2} (-1)^{\chi}$$

where $\chi = 2p_1 + 2p_2 + 3b_3$ and $d_{j_k} \equiv 2j_k + 1$. Finally, we contract the "north and south" amplitudes above along the five horizontal blue spins in the top right panel of Figure 6.3. Therefore, we write the expression for the 16-cell BF spinfoam amplitude as:

$$W(j, i_n) = \sum_{b_1 \ldots b_5} (W_N \cdot W_S \cdot d_{b_1} d_{b_2} d_{b_3} d_{b_4} d_{b_5}) \cdot \prod_{k=1}^{16} \sqrt{d_{i_k}}\ , \tag{6.14}$$

where $i_n = i_1 \ldots i_{16}$.

6.4 Computational strategy

We aim to compute the expectation values (6.10)-(6.11), the quantum spread (6.12), and the correlation function (6.13) for the 16-cell spinfoam described in section 6.2. We want to do it for increasing values of j, which requires computing many 16-cell spinfoam amplitudes (6.14).

Therefore, our priority is efficiently computing the amplitude (6.14) for each possible combination of boundary intertwiners. Next, we must assemble multiple amplitudes to calculate the quantities we are interested in.

6.4.1 Computing the 16-cell spinfoam amplitude

As discussed in Section 6.3.3, the topological BF 16-cell spinfoam amplitude can be written as a 48j Wigner symbol. Then, we can split the 48j Wigner symbol into a bunch of 21j symbols. Finally, each 21j symbol can be decomposed into 6j and 9j Wigner symbols. We proceed as follows to compute the amplitude numerically.

We first pre-compute all the 6j and 9j Wigner symbols with spins $j \leq 2j_{max}$. We do it for increasing values of j_{max}, relying on `fastwigxj` library [170, 91]. By default, it evaluates Wigner symbols quickly by looking at precalculated tables produced using the `wigxjpf` library. Then, we compute all the possible 21j symbols (F.14) with spins $i_1, i_2, i_3, i_4, b_1, b_2, b_3, p_1, p_2$ less than or equal to a characteristic spin $\sim 2j_{max}$ (a combination of others bounds some of these, as triangular inequalities constrain the spins). Again, we do it for increasing j_{max} values. The number of such symbols rapidly increases along with j_{max}. We store the 21j symbols using the `parallel hash map`, publicly available at [176]. The size of the hash tables as a function of j_{max} is shown in Figure 6.4. By measuring the size of the calculated tables with a simple best

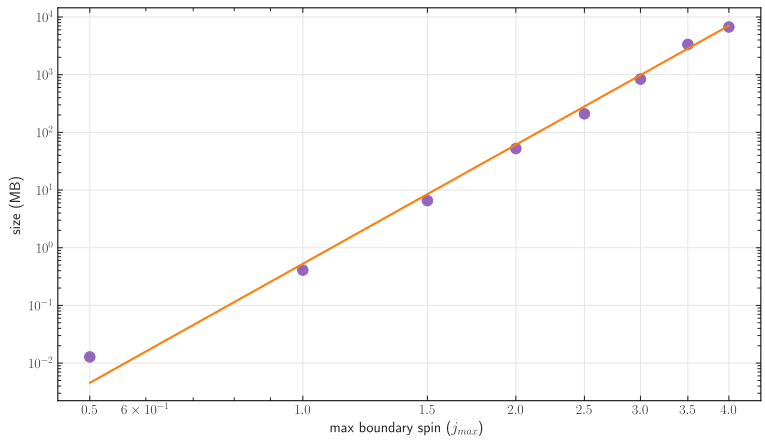


Figure 6.4: *Size of the computed parallel hash map tables of 21j Wigner symbols. Each symbol is stored in uint8 format to save memory. The hash table size roughly increases as* $j_{max}^{6.86}$.

polynomial fit, we conclude that the size roughly increases as:

$$\text{size} \approx j_{max}^{6.86} - 0.64 \,. \tag{6.15}$$

Our strategy is to compute the 16-cell spinfoam amplitude (6.14) by retrieving all the required 21j symbols from the hash tables. To do it efficiently, we must pre-load each table into the RAM or a cluster node-local storage before computing the amplitudes. The relation (6.15) imposes a significant constraint on the required memory. Furthermore, as j_{max} grows, the

number of Wigner symbols to sum over and the time necessary to perform lookups in the hash table increase. Consequently, we must limit j_{max} to very low values. We choose a maximum boundary spin $j_{max} = 4$. This also corresponds to the deep quantum physical regime we want to investigate in this paper.

6.4.2 Computing expectation values with MCMC

We can now compute the 16-cell amplitude (6.14) efficiently for different combinations of boundary intertwiners at fixed boundary spin j. We have the fundamental tool for computing the quantum expectation values and correlations. However, we need to address a major issue. In order to compute (6.10) there are $(2j+1)^{16}$ summations to be performed. For $j = 3$, there are $\sim 3.3 \cdot 10^{13}$ elements in the sum. Assuming (optimistically) that each amplitude (6.14) can be computed in $\sim 10^{-5}$ seconds, we would need ~ 10 years to compute a single quantum expectation value! We can solve this issue using Markov Chain Monte Carlo (MCMC) over the boundary intertwiners. Specifically, we employ the Metropolis-Hastings algorithm [148]. We report a brief introduction to the Metropolis-Hastings algorithm in F.1, and we describe the details of the implementation below. This technique has been originally applied to spinfoams in [78] in the case of a spinfoam model with 20 boundary tetrahedra, to which we refer for further details.

To compute (6.10) using MCMC, we choose the following normalized target distribution over the state space (6.3):

$$f_j(i_n) = \frac{W^2(j, i_n)}{\sum\limits_{i_n} W^2(j, i_n)} \,. \tag{6.16}$$

We use the Metropolis-Hastings algorithm to build a Markov chain with length N_{mc} constituted by intertwiner states $[i_n]_1, [i_n]_2, \ldots, [i_n]_{N_{mc}}$ so that each intertwiner state is generated from the distribution (6.16). We use a truncated normal distribution rounded to integers as the proposal distribution, and each intertwiner (in every state is sampled from a one-dimensional normal distribution. The center of the distribution is the value of the corresponding intertwiner in the previous state truncated between 0 and $2j$ (the range of all intertwiners). For example, if $j = 2$ and $i_1 = 3$ in the state $[i_n]_s$, the proposed value for i_1 in the state $[i_n]_{s+1}$ is sampled from a normal distribution centered around 3 and truncated between 0 and 4, and so on for the other intertwiners (each one is independent of others). We report in the flowchart 4 below the full implementation of the algorithm to store Markov chains constituted by intertwiner states, using the target distribution 6.16. The parameters of the MCMC algorithm, the definition of the proposal distribution, and the expression of the truncated coefficients related to the proposal distributions are reported in the F.1.

We show in Figure 6.5 the squared amplitude $W^2(j, [i_n]_s)$ as a function of the number of steps s for the first 10^4 iterations of the MCMC algorithm 4. We notice that the "hotspots" become less and less frequent as j increases. This is most probably because the space of states becomes more prominent, so the sampler requires more iterations to find the relevant contributions. To overcome this issue, an interesting approach (currently under development) consists of replacing MCMC in spinfoams with GFlowNets [177]. We tried many different values for the burn-in parameter b (ranging from 0 to 10^4) and found no significant dependence.

Algorithm 4 Random walk over boundary intertwiners

for $j = 1 \dots j_{max}$ **do**
  Choose N_{mc}, the burn-in parameter b and the standard deviation σ as in F.1
  Load into memory the hash table with the 21j Wigner symbols corresponding to j
  Sample a random intertwiners configuration $[i_n]_1$ and compute $W(j, [i_n]_1)$
  Set initial multiplicity to 1
  for $s = 1 \dots N_{mc}$ **do**
    Generate a new state $[i_n]$ from $[i_n]_s$
    if $[i_n] = [i_n]_s$ **then**
      Increase the multiplicity by 1
      continue
    else
      Compute $W(j, [i_n])$
      Compute $p = \min\left\{1, \frac{W^2(j,[i_n])}{W^2(j,[i_n]_s)} \frac{C_{0,2\,j,\sigma}([i_n]_s)}{C_{0,2\,j,\sigma}([i_n])}\right\}$
      Generate a uniform random number r between 0 and 1
      if $r < p$ **then**
        if $s > b$ **then**
          Store $[i_n]_s$, $W(j, [i_n]_s)$, and the corresponding multiplicity
        Set $[i_n]_s \to [i_n]$, $W(j, [i_n]_s) \to W(j, [i_n])$
        Set the multiplicity to 1
      else
        Increase the multiplicity by 1
  Dump to disk all the states, amplitudes, and the corresponding multiplicities

This can be because the distribution (6.16) has many isolated peaks regularly distributed in the space (6.4). The algorithm's time complexity 4 depends on the proposal distribution's standard deviation σ. The higher σ is, the higher the probability that the sampler moves from the current state high (therefore, a new amplitude must be computed). Moreover, the standard deviation of the proposal distribution affects the acceptance rate: the lower σ is, the higher the acceptance rate becomes (in the extreme case $\sigma = 0$, the sampler never moves, and all proposed states are accepted). We balance σ so that the acceptance rate of the Metropolis-Hastings algorithm ranges between $\sim 30\%$ and $\sim 45\%$.

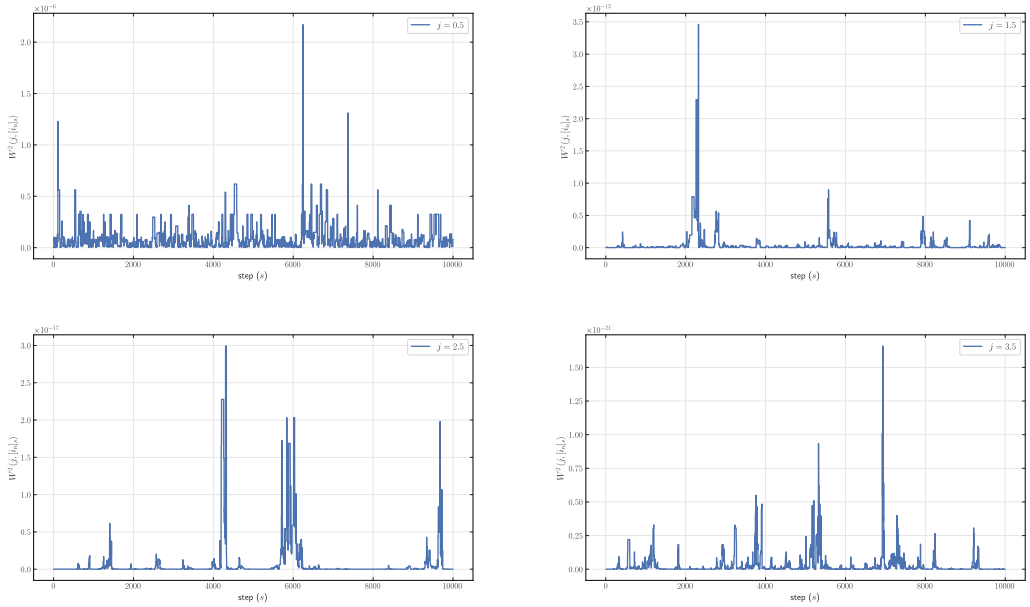


Figure 6.5: *Squared amplitude* $W^2(j,[i_n]_s)$ *computed in the algorithm 4 as function of the number of steps s along the Markov Chain. The frequency of amplitude's peaks tends to decrease as j increases.*

Next, we use the intertwiner draws stored during the algorithm 4 to evaluate the expectation value (6.10) by applying the Monte Carlo summation [138]:

$$\langle O_k\rangle_{mc} = \frac{1}{N_{MC}}\sum_{s=1}^{N_{mc}}\langle j,[i_k]_s|O_k|j,[i_k]_s\rangle\ . \tag{6.17}$$

With respect to (6.10), we replaced a sum over $(2j+1)^{16}$ intertwiners with a sum over N_{mc} elements. This hugely simplifies the calculation, which would be impractical otherwise. The application of Monte Carlo to the corresponding quantum spread (6.12) and to the correlation functions (6.13) is straightforward:

$$\Delta\langle O_k\rangle_{mc} = \sqrt{\langle O_k^2\rangle_{mc} - \langle O_k\rangle_{mc}^2}\ , \tag{6.18}$$

$$C(O_k,O_{k'})_{mc} = \frac{\langle O_kO_{k'}\rangle_{mc} - \langle O_k\rangle_{mc}\langle O_{k'}\rangle_{mc}}{(\Delta\langle O_k\rangle_{mc})\,(\Delta\langle O_{k'}\rangle_{mc})}\ . \tag{6.19}$$

We want to estimate the error in (6.17) due to the statistical fluctuations of MCMC. For this purpose, we repeat the calculation of (6.10) multiple times using different (and independent) Markov chains with the same parameters. Then, we compare the results, allowing us to improve the estimation and quantify the fluctuations. If we have C different estimates $\langle O_k\rangle_{mc}^{(1)}$, $\langle O_k\rangle_{mc}^{(2)}$, ... $\langle O_k\rangle_{mc}^{(C)}$, we compute the corresponding average:

$$\overline{\langle O_k\rangle}_{mc} = \frac{1}{C}\sum_{c=1}^{C}\langle O_k\rangle_{mc}^{(c)}\ . \tag{6.20}$$

For completeness, we write the corresponding average for the spread and correlations explicitly:

$$\overline{\Delta\langle O_k\rangle}_{mc} = \frac{1}{C}\sum_{c=1}^{C} \Delta\langle O_k\rangle^{(c)}_{mc} \,, \tag{6.21}$$

$$\overline{C(O_k, O_{k'})}_{mc} = \frac{1}{C}\sum_{c=1}^{C} C(O_k, O_{k'})^{(c)}_{mc} \,. \tag{6.22}$$

To have an intuitive visualization, we consider the normal distribution associated with the average (6.20). For this purpose, we introduce the standard deviation between the expectation values over the chains:

$$\sigma = \sqrt{\frac{1}{C}\sum_{c=1}^{C}\left(\overline{\langle O_k\rangle}_{mc} - \langle O_k\rangle^{(c)}_{mc}\right)^2} \,, \tag{6.23}$$

So that we can write the normal distribution as:

$$\mathcal{N}_O(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\overline{\langle O_k\rangle}_{mc}}{2\sigma}\right)^2} \,. \tag{6.24}$$

If the standard deviation (6.23) is low, the statistical fluctuations in the estimate (6.20) are reasonably under control. We notice that increasing C does not reduce (6.23), as the standard deviation is reduced by increasing the number of samples N_{mc} along each chain.

Therefore, we must run the algorithm 4 C times to store as many independent Markov Chains. Since each chain is independent, we can parallelize the code across multiple threads using the C/C++ OpenMP library [178]. Each thread runs a different Markov chain with the given parameters, storing the computed intertwiner draws. The parallel hash table with all the required 21j Wigner symbols is loaded only once into the RAM so that each thread retrieves the symbols (F.14) from the same table. By doing this, we can compute the average over C expectation values (6.20), spread (6.21) and correlations (6.22) computed with as many independent chains. For this task, we take advantage of the Python libraries `Pandas` and `Numpy` [179, 180]. As reported in table F.1, in this paper, we computed and stored $C = 30$ Markov chains, each with 10^6 intertwiner draws.

Finally, we notice that the expression of the 16-cell spinfoam amplitude (6.14) requires a considerable numerical effort to be computed, despite the strategy described in Section 6.4.1. This limitation allows us to calculate only diagonal boundary operators. For this class of operators, the only amplitudes (6.14) we need to compute are those required in the algorithm 4. In the case of non-diagonal operators, such as the volume of boundary tetrahedra, we would need to compute $2j$ additional amplitudes (6.14) for each term in the sum (6.17).

6.5 Numerical results

In this Section, we report the numerical values obtained for the expectation values (6.10), the quantum spread (6.12), and correlations (6.13) following the procedure described in Section 6.4.2. Here we focus on the dihedral angle operator, describing the external dihedral angle

between faces a and b of each boundary tetrahedron. Its expression in the basis states (6.4) is [77]:

$$\langle j, i_n | \cos(\theta)_k | j, i_n \rangle = \frac{i_k(i_k+1) - 2j(j+1)}{2j(j+1)} \,. \tag{6.25}$$

Due to its simplicity and geometrical interpretation, the dihedral angle operator 6.25 is an optimal candidate to investigate the geometrical properties of the 16-cell triangulation described in Section 6.2. Furthermore, it is the same operator studied in [77, 78], which allows immediate comparison with previously studied triangulations.

6.5.1 Expectation values

The expectation value (6.20) of the dihedral angle operator (6.25) and the quantum spread (6.21) are shown in Figure 6.6. Both these quantities have been computed averaging over $C = 30$ independent Markov chains.

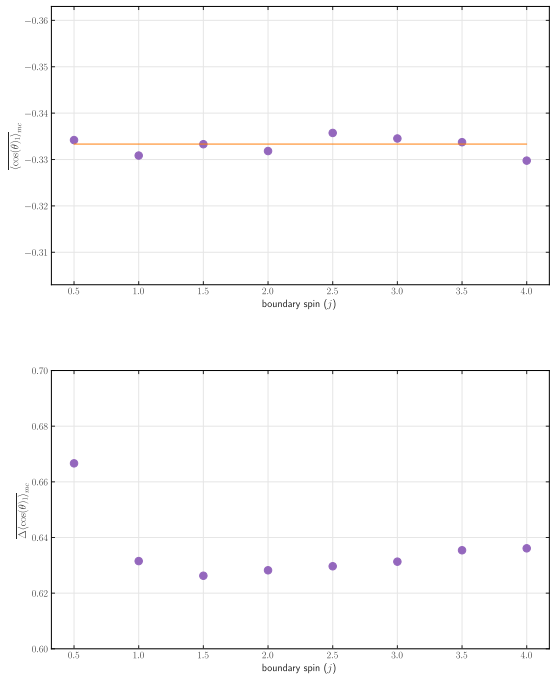


Figure 6.6: *Left: expectation values* (6.20) *of the dihedral angle operator* (6.25). *We show only the result for the first node, but all other nodes have the same behavior. The orange line shows the value of the cosine of the external dihedral angle of a regular tetrahedron* $\arccos\left(\frac{1}{3}\right)$. *Right: corresponding quantum spread* (6.21).

A few comments are in order. The first observation is that the expectation value (6.25) of all 16 boundary tetrahedra is peaked on the value of an external dihedral angle of an equilateral tetrahedron. This coincides with the result obtained with the simplest possible triangulation of a 3-sphere [77] and the triangulation used in [78]. Unlike the 4-simplex studied in [77], the 16-cell is not self-dual. This dynamical result of the global geometry confirms that the metric

of the state (6.4) averages to that of a regular 3-sphere even in the 16-cell geometry and is not a consequence of the reduction (6.3). The second observation is that the quantum spread is slightly increasing as a function of the boundary spin j, which indicates that the quantum fluctuations are ample. This agrees with the result obtained in [78] in the topological model. It is reasonable to expect more significant fluctuations for the EPRL model. We leave the analysis of the EPRL case for future work.

As described in F.1, a non-zero correlation exists between the intertwiner states stored during the algorithm 4. This is a consequence of the Markovian nature of the process. We want to check whether the quality and number of states generated with the algorithm (4) are sufficient to accurately approximate the target distribution (6.16). For this purpose, we compute the autocorrelation function (5.26) for the matrix elements of the dihedral angle operator (6.25) using the states generated with the Metropolis-Hastings algorithm. These terms define the expectation values (6.17). We computed the autocorrelation function over all the 30 chains stored and for all nodes, finding very similar behavior. We report explicitly in Figure 6.7 the autocorrelation function computed using chain one and considering the first node of the 16-cell spinfoam. The result clearly shows that the autocorrelation is significant at short lags but goes to zero pretty fast (we show a maximum lag of 8000 for visualization purposes, but each chain contains 10^6 states). This is precisely what we expect for a Markov chain that converges to a stationary distribution, indicating that the sample size that we considered is sufficient. Interestingly, the autocorrelation's decaying becomes slower as the boundary spin j increases.

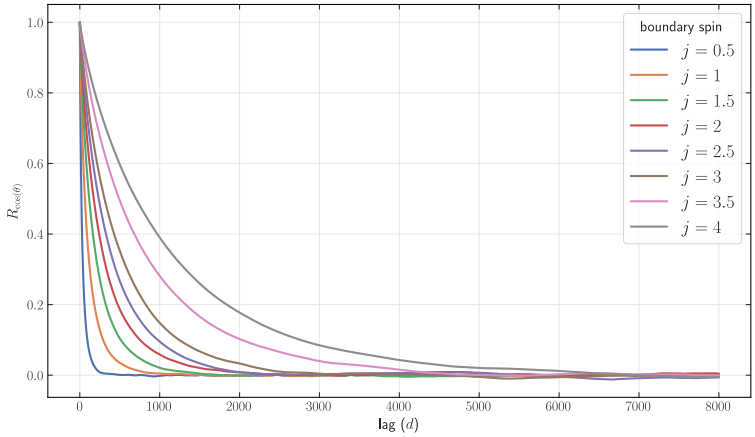


Figure 6.7: *Autocorrelation function* (5.26) *of the expectation value* (6.17) *of the dihedral angle operator* (6.25). *The autocorrelation decays more slowly as a function of the lag d as the boundary spin j increases.*

As a final check for the effectiveness and reliability of the MCMC algorithm described in Section 6.4.2, we repeat the calculation of the dihedral angle operator multiple times using equation (6.20). The plot of the corresponding normal distribution (6.24) is reported in Figure 6.8. For convenience, we report the plot considering node 1 of the 16-cell spinfoam, but we performed the calculation over all 16 nodes, finding a very similar behavior. Interestingly, the

statistical fluctuations become more significant as the boundary spins j increase. This confirms that we have enough control over the stochastic fluctuations due to Monte Carlo.

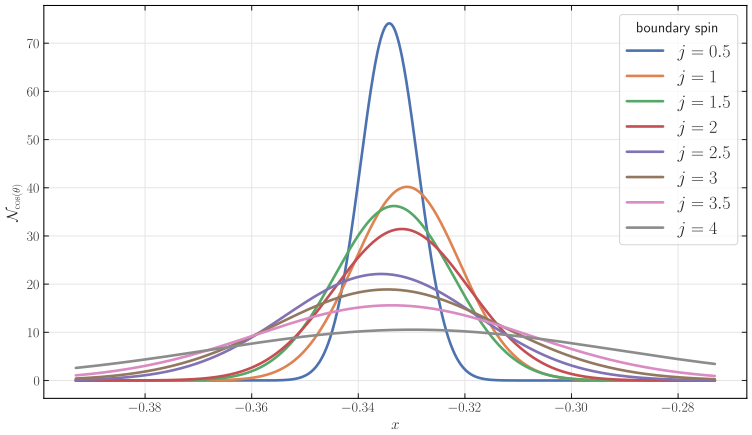


Figure 6.8: *Normal distribution* (6.24) *of the expectation values of the dihedral angle operator* (6.25). *The statistical fluctuations increase as a function of the boundary spin* j.

6.5.2 Correlations

We compute the correlations (6.13) of the dihedral angle operator (6.25) for the 16-cell geometry described in Section 6.2. We report the result of (6.22) in the right panel of Figure 6.9 and in Figure 6.10. In the left panel of Figure 6.9, we report the exact same boundary of the 16-cell spinfoam defined in Section 6.2, labeling the nodes with numbers. We do this to identify the correlations within the 16-cell geometry quickly. Looking at the correlation values, we notice that each node k is equivalent to node $k \pm 4$. This is because the 16-cell spinfoam boundary is symmetrical for 90 degree rotations. Interestingly, there are relatively high values of the correlations between nodes not directly connected by a link (for example, 1-9, 5-13, etc.). In other words, some couples of non-adjacent nodes are strongly correlated. This is the striking difference between the 16-cell geometry and the spinfoam model studied in [78] or the 4-simplex [77].

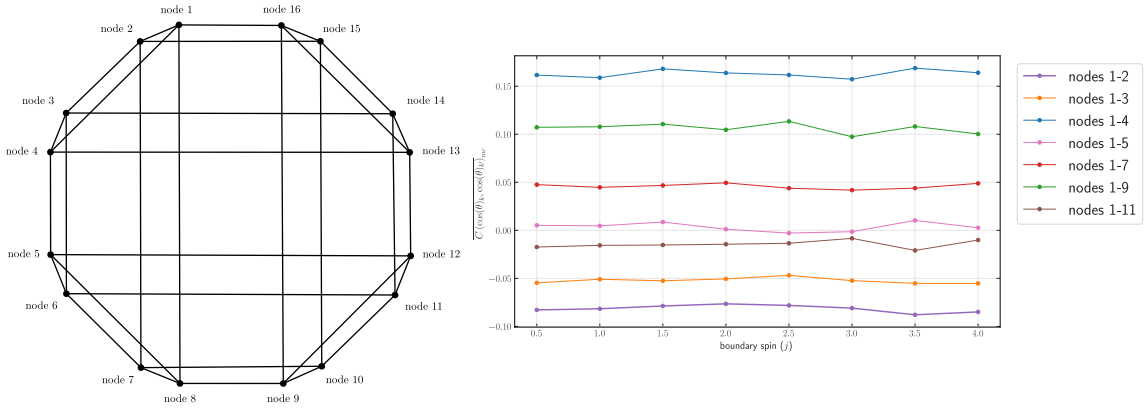


Figure 6.9: *Left: 16-cell spinfoam boundary (same as Figure 6.3) with nodes labeled by numbers Right: Some values of quantum correlations* (6.22) *as a function of the boundary spin* j.

In Figure 6.9, we explicitly show the correlation between just a few couples of nodes so as not to clutter the picture. We infer that correlations are constant as a function of the boundary spin j, confirming the trend observed in [78, 77]. In Figure 6.10, we show the numerical values of (6.22) computed between all possible combinations to emphasize the complete pattern of nodes. We report the tables for the minimum and maximum values considered for the boundary spins for visualization purposes. From Figure 6.9, it is evident that the values between $j = 0.5$ and $j = 4$ have very similar values.

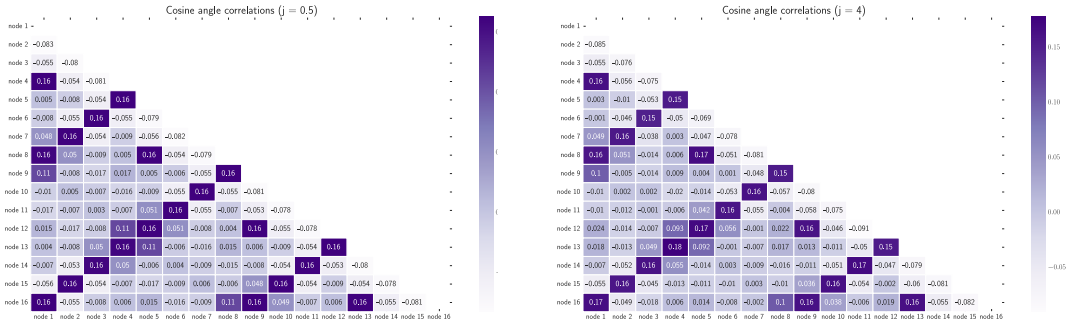


Figure 6.10: *Quantum correlations* (6.22) *computed for all possible couples of nodes at fixed boundary spin* j. *We show explicitly only the minimum and the maximum values* j. *As Figure 6.9 reveals, values in between give similar results. Each node* k *is equivalent to node* $k \pm 4$. *Left: case* $j = 0.5$. *Right: case* $j = 4$.

6.6 Conclusions

In the deep quantum regime, we have presented a numerical investigation of the non-perturbative Hartle-Hawking state defined in covariant loop quantum gravity. We have computed this state's mean geometry, fluctuations, and correlations. We have relaxed the truncation from the tetrahedron triangulation of the cosmological 3-sphere to a 16-cell triangulation. We have computed the quantities above in the simplified setting where the dynamics are topological.

The numerical analysis is consistent with the hypothesis that refining the triangulation does not substantially affect the global physical picture. The mean geometry remains consistent with an approximation of a metric threes-sphere. Fluctuations and correlations remain high, as in the tetrahedral truncation. We expected a clear dependence of correlation on the separation of the nodes, but we have not found it. We do not know if this is due to the topological nature of the *BF* simplification taken. The calculation must be repeated with the full physical amplitude to test so. This task requires pushing the numerical tools further.

***

Acknowledgments We thank Carlo Rovelli for many discussions on this project and for extensive comments on the final draft of this paper. PF thanks Mikko Karttunen for supporting the development of part of this project within his course in Scientific Computing. PF also warmly thanks Gregory Popovitch for his help in implementing the `parallel hashmap` for this research project and Hakan T. Johansson for exciting discussions and clarifications about `fastwigxj`. We acknowledge the Shared Hierarchical Academic Research Computing Network (SHARCNET) for granting access to their high-performance computing resources. We particularly thank the Compute/Calcul Canada staff for the constant support provided with the Cedar and Graham clusters. This work was supported by the Natural Science and Engineering Council of Canada (NSERC) through the Discovery Grant "Loop Quantum Gravity: from Computation to Phenomenology." We also acknowledge support from the ID# 62312 grant from the John Templeton Foundation, as part of the project “The Quantum Information Structure of Spacetime” (QISS). The Canada Research Chairs Program at Western University supports FV’s research. The Canada Research Chairs Program supports FV’s research at Western University. FV is affiliated with the Perimeter Institute for Theoretical Physics. The Government of Canada supports research at Perimeter Institute through Industry Canada and by the Province of Ontario through the Ministry of Economic Development and Innovation. Western University and Perimeter Institute are located in the traditional lands of Anishinaabek, Haudenosaunee, Lūnaapèewak, Attawandaron, and Neutral peoples.

Chapter 7

Numerical approach to the Black-to-White hole transition

Statement of co-authorship

This Chapter illustrates the paper [141], written by me. It was published in *Physical Review D* on 15 June 2023. Since it is a solo-author paper, it seems unnecessary to remark on my contributions in this specific case. I started working on this paper after studying [18], which for me was a revelation, as well as a source of inspiration throughout most of my Ph.D.. This manuscript corresponds to the article's *Physical Review D*-published version, which agrees with the APS copyright policies.

7.1 Introduction

At present, black holes seem to be perfectly described by classical general relativity, including their behavior in the strong field regime [181]. We have no reason to suspect that, beyond the black hole horizon, general relativity does not continue to provide a reasonable physical description inside it. The region where our knowledge falters is the center: we have no idea what happens to an object after it reaches the singularity. Furthermore, the distant future of a black hole is still quite a mystery. The calculation originally made by Hawking [182] shows that the black hole shrinks due to the back reaction of the Hawking radiation. The black hole should become smaller and smaller, but after this phase, nothing is known. The perturbative formulation of quantum gravity disregards non-perturbative quantum-gravitational phenomena. This is the reason why the full theory of quantum gravity is required. The possibility of black hole decay via gravitational quantum tunneling is currently one of the most intriguing hypotheses on the future of these objects [183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203].

In the last few years, considerable effort has been devoted to investigating the phenomenon using the covariant "spinfoam" formulation of loop quantum gravity [18, 204, 205, 206, 207, 208]. At the same time, there have been remarkable advances in the development of computational methods in the field. A few examples are the application of MCMC methods to

investigate the semiclassical limit [57] as well as the deep quantum regime [78, 138], the study of cuboid renormalization [147], the introduction of effective spinfoams [63, 64], and the study of the EPRL amplitudes using high-performance computing [47, 9, 48, 143]. One of the main reasons for developing techniques to compute EPRL spinfoam amplitudes was to investigate the black-to-white transition using computational methods [209, 10]. In this sense, this paper aims to be the first direct link between these two research directions.

We outline an algorithm to compute the amplitude and apply it to calculate the crossing time of the transition. We estimate it numerically and analytically, modifying the boundary state with respect to the calculation of the same observable currently present in the spinfoam literature, corresponding to a different physical regime [209, 206, 204]. We find the same result: the crossing time scales linearly with the mass. The calculation described here is remarkably simpler and shows that the crossing time does not depend on the extrinsic curvature of the boundary geometry. Therefore, our result is in excellent agreement with the previous estimates, supplementing them with new physical information. The paper is organized as follows. Section 7.2 briefly reviews the quantum tunneling hypothesis and the necessity of a full quantum gravity theory to describe it. In Section 7.3, we describe the external geometry of the process. In Section 7.4, we write the four-dimensional spinfoam amplitude of the black-to-white hole transition, and in Section 7.5, we outline the algorithm to compute it. Finally, in Section 7.6, we evaluate the crossing time of the transition. Unless explicitly indicated otherwise, we use the Planck unit system ($c = \hbar = G = 1$) in the following.

7.2 The quantum tunneling

Regardless of what happens in the future, after the full evaporation of the black hole has occurred, it is reasonable to expect that in a distant forward time, all that remains is regular spacetime. That is, we expect that there is a spacetime with a causal structure after the end of the black hole evaporation. We know from classical GR that a collapsed star creates a horizon. The "cosmological censor" conjecture [210] states that every singularity in classical GR is always hidden inside a horizon. Assuming that this conjecture is true since GR is invariant under time reversal, the opposite also turns out to be true. Therefore, even in the future, the "putative singularity" should be closed inside a horizon. To describe the process, we need a description of the external classical geometry and a quantum one, which provides information on the tunneling transition inside the black hole.

These ingredients are provided precisely by loop quantum gravity, which describes the transition amplitude between classical geometries. In particular, the closed surface surrounding the classical region can be arbitrarily chosen. Therefore, the central singularity is enclosed inside a boundary, namely a space-like surface Σ with a classical geometry defined on it. A tunneling effect exists between the two space-like regions on "opposite sides" of the singularity. This is a purely quantum effect as these regions have no classical transition. Thus, there is no possible classical evolution. A possibility considered in the literature [211] is that there is a white hole after the evaporation of the (small) black hole, which remains as a remnant, which should radiate in the low-frequency spectrum [212]. In Figure 7.1, we sketch the tunneling process using an (extended) Penrose diagram. The blue lines denote the gravitational horizon,

and the green ones indicate the space-like boundary surface. The B region represents the future of the black hole after the evaporation, whereas A is the region around the singularity center. The quantum theory describes the tunneling process. It is reasonable to expect that the degrees of freedom of the latter are not arbitrarily small but comparable to the black hole size.

First, we must truncate the theory to compute transition amplitude using covariant loop quantum gravity. We do so by discretizing Σ and the four-dimensional interior region. If we know the (intrinsic and extrinsic) geometry of Σ, we can write an extrinsic coherent state in the Hilbert space of loop quantum gravity to the truncation.

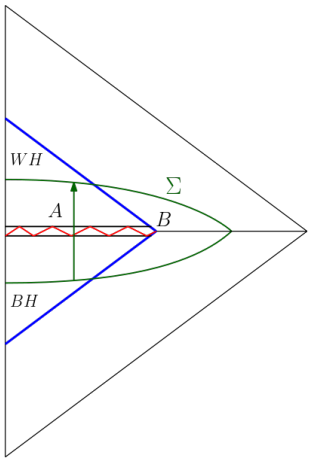


Figure 7.1: *Extended Penrose diagram describing the tunneling process. The B region represents the future of the black hole after the evaporation, whereas A is the region around the singularity center.*

7.3 Geometry

The final external metric depends only on two parameters: the mass of the black hole m and the "time" T between the lower and upper regions [213]. Therefore, these two parameters completely describe the external geometry. There are several characteristics and time scales involved in the process. These have been deeply described in [204]. Crucially, since the external metric can be explicitly given as a function of m and T, the transition amplitude describing the tunneling process also depends on the same parameters. The surface Σ in Figure 7.1 is formed by two flat 2-spheres joined at their boundary. We conventionally define these "upper" and "lower" boundary surfaces, associated with the future and past of the black hole, as Σ_+ and Σ_-.

7.3.1 Discretization of Σ

To write the transition amplitude explicitly and compute the crossing time, choosing a discretization of Σ is necessary. We use the same discretization originally introduced in [18]. The geometry of the triangulation in terms of the Ashtekhar variables was derived entirely in the same paper, to which we refer for further details.

Triangulation: each 2-sphere $\Sigma_\pm$ is first triangulated using a single equilateral flat tetrahedron. Then, the triangulation is refined by splitting each tetrahedron into four equal isosceles tetrahedra, as shown in Figure 7.2. Therefore, the total surface Σ is triangulated with eight

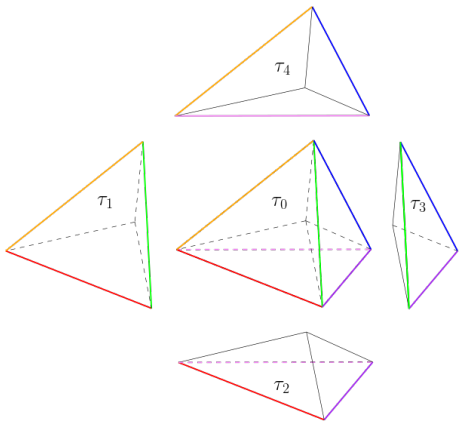


Figure 7.2: *Regular tetrahedron τ_0 split into four isosceles tetrahedra $\tau_1, \tau_2, \tau_3, \tau_4$. Both Σ_+ and Σ_- are triangulated with such four isosceles tetrahedra.*

boundary tetrahedra. The geometry is consequently composed of two 4-simplices (each with zero 4-volume) joined by a tetrahedron.

Two-complex: the corresponding two-complex has two vertices contracted over a bulk intertwiner. The boundary graph is constituted of 16 links. Of these, 4 "angular" links l_a connect the nodes between different vertices, while for each vertex, there are 6 "radial" links $l^\pm_{ab}$. So there are two different types of links. This discretization ultimately defines the spinfoam associated with the transition amplitude, which is described in section 7.4.

7.3.2 Extrinsic boundary states

After defining a discretization of Σ, it is possible to write down a coherent state describing the geometry. Among the possible definitions of coherent states, in [18], the authors considered the "extrinsic" coherent states [1], originally introduced by Thiemann [214], parametrized as in [215] in terms of twisted geometries [128]. These states depend on two unit-length source and target vectors $\vec{n}_s$, $\vec{n}_t$ and on a complex number z, which we write as:

$$z = \eta + i\,(\beta + \gamma\zeta)\ . \tag{7.1}$$

In (7.1), $\eta \in \mathbb{R}^+$ is the dimensionless area of the triangular face dual to the link, $\zeta \in [0, 4\pi)$ is the boost angle between the normals of the tetrahedra [216, 217] and β is an extra rotation. We refer to [18] for the connection between the boost angle and the discretized holonomy along each link. Extending analytically the definition of the Wigner matrices $D^j(h)$ to complex parameters, where $h \in SU(2)$, the extrinsic coherent states can be written as:

$$\Psi_{\sigma,n_s,n_t,z}(h) = \sum_j (2j+1) e^{-j(j+1)/2\sigma} \times \\ \mathrm{Tr}[D^j(h) D^j(n_t e^{z\frac{\sigma_3}{2}} n_s^{-1})]\ , \tag{7.2}$$

where j is the spin attached to the link. The $SU(2)$ elements n_s, n_t in (7.2) rotate the unit vector along the $\hat{z}$ axis into the source vector $\vec{n}_s$ and the target vector $\vec{n}_t$, respectively. The state on the graph is defined as the product of a factor (7.2) for each link. Notice that apart from a phase factor, the ratio between two terms which differ by one unit in the component of the magnetic moment in the diagonal Wigner matrix $D^j\left(e^{z\frac{\sigma_3}{2}}\right)$ is:

$$\frac{e^{\eta n}}{e^{\eta(n-1)}} = e^{\eta} > 10^3 \quad \text{for} \quad \eta \geq 7 \,, \tag{7.3}$$

where $n \in [-j, j]$. Therefore, when the real part η of ζ is large enough, the trace in (7.2) is completely dominated by the highest magnetic moment component. As a consequence, the Wigner matrix can be approximated as:

$$D^j_{k,q}\left(e^{z\frac{\sigma_3}{2}}\right) \approx \delta^j_k \delta^j_q e^{zj} \,. \tag{7.4}$$

When condition (7.4) is satisfied, the state (7.2) can be expressed as:

$$\begin{aligned}\Psi_{\sigma,n_s,n_t,z}(h) \approx \sum_j (2j+1) e^{-j(j+1)/2\sigma + zj} \times \\ \sum_{n,m} D^j_{n,j}(n_t) D^j_{m,n}(h) D^j_{j,m}(n_s^{-1}) \,,\end{aligned} \tag{7.5}$$

and the sum over j in (7.5) is peaked on the minimum of $j(j+1)/(2\sigma) - \eta j$, which is

$$j_m = \eta\sigma - \frac{1}{2} \,. \tag{7.6}$$

7.3.3 Balancing the spread

The quantity σ in the definition of the extrinsic coherent states (7.2) plays an important role, as it determines whether the state is peaked on the area or the extrinsic curvature. We first consider a generic dependence:

$$\sigma = j_m^{-\alpha} \,, \tag{7.7}$$

where $\alpha \in \mathbb{R}$. From (7.6) we find:

$$\eta = j_m^{\alpha+1} + \frac{j_m^{\alpha}}{2} \,. \tag{7.8}$$

Recalling the relation existing between the spin and the area operator in LQG, it can be shown [145] that the (relative) spread in the non-commuting areas and embedding data is:

$$\frac{\Delta\zeta}{\langle\zeta\rangle} \sim j_m^{\frac{\alpha}{2}} \,, \quad \frac{\Delta A}{\langle A\rangle} \sim j_m^{-\left(\frac{\alpha}{2}+1\right)} \,. \tag{7.9}$$

The product between the two relative spreads (7.9) is independent of α and goes to zero in the large j_m limit. A good requirement to recover the semiclassical behavior is that the relative dispersions (7.9) vanish for $j_m \longrightarrow \infty$. Along with condition (7.7) and requiring that both η and j_m are large, this results in the range $\alpha \in [-1, 0]$ for the extrinsic coherent states (7.2) to behave semiclassically. In [18], the choice made was $\alpha = -\frac{1}{2}$ to peak the state both on the area and on

the extrinsic curvature in the large j_m limit. Crucially, notice that with this choice, when j_m is small, condition (7.4) is not a valid approximation. Therefore, we consider the case $\alpha > 0$ so that (7.4) is valid. From (7.9), this choice leads to a sharp area operator and an extrinsic spread curvature. In the language of [145], this corresponds to a large heat-kernel time. Physically, this means increasing the quantum spread associated with the boost angle operator between Σ_- and Σ_+. From (7.7) and (7.8) we obtain:

$$\frac{-j(j+1)}{2\sigma} + j\,\eta = -\frac{j_m^\alpha}{2}(j-j_m)^2 + \frac{j_m^{\alpha+2}}{2}\,, \tag{7.10}$$

where the last term is absorbed in the normalization factor of the amplitude. In the transition amplitude, every link of the boundary graph has an extrinsic coherent state (7.2) associated with it. Therefore, each boundary link has an independent (infinite) sum. The condition $\alpha > 0$ increases the factor j_m^α, so putting a small half-integer cut-off K around j_m allows a good amplitude approximation. This corresponds to the physical limit in which we recover the intrinsic coherent states from the extrinsic ones.

The semiclassical condition $\alpha \in [-1, 0]$ implies that states (7.2) become rapidly spread around j_m for large j_m. In such a regime, a possible approach to computing the amplitude numerically could consist in using importance sampling Monte Carlo to overcome the multiple independent sums over boundary links. However, one has to deal with the sign problem in such a case. Alternatively, one could use the analytical methods developed in [209, 206] suitable for the semiclassical regime.

7.3.4 Normal orientation

In this Section, we define the orientation of normals to the boundary tetrahedra in the triangulation described in Section 7.3.1. These were originally computed in [18]. We parametrize the Wigner matrix as in [218]:

$$D^j_{m,j}(n) = D^j_{m,j}(\phi, \theta, -\phi) = e^{-im\phi} d^j_{m,j}(\theta) e^{ij\phi}\,, \tag{7.11}$$

where $d^j_{m,j}(\theta)$ is the small Wigner matrix [218, 169]. The $\vec{n}$ vector can be parametrized with the polar angles as usual:

$$\vec{n} = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)\,.$$

Using the orientation of Figure 7.3, after some calculation [18] the following values are obtained:

$$\vec{n}_0 = (0, 0), \tag{7.12}$$

$$\vec{n}_k = \left(\arccos\left[-\sqrt{\tfrac{2}{3}}\right],\ \varphi_k\right), \tag{7.13}$$

with $k = 1, 2, 3$ and

$$\varphi_1 = 0, \quad \varphi_2 = \frac{2}{3}\pi, \quad \varphi_3 = -\frac{2}{3}\pi\,. \tag{7.14}$$

The required value for β in (7.1) turns out to be $\beta = 0$ for the equilateral faces, and $\beta = \varphi_k - \varphi_{k'}$ for the isosceles faces. This extra rotation along the $\hat{z}$ axis must match the triangles in the (x, y)

plane. The effect of such rotation is such that we can replace the $SU(2)$ element n in (7.11) with another element τ, whose third component is zero:

$$D^j_{m,j}(\tau) = D^j_{m,j}(\phi, \theta, 0) = e^{-im\phi} d^j_{m,j}(\theta) \,. \tag{7.15}$$

For the target on the same link, we add a parity transformation so that $\theta \longrightarrow \theta - \frac{\pi}{2}$. Finally, we

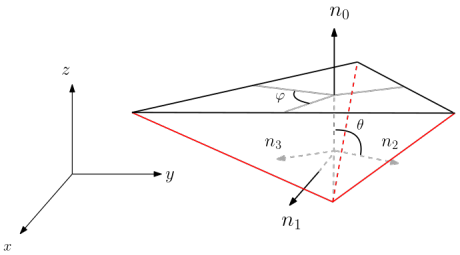
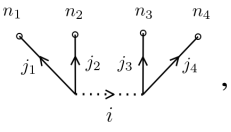


Figure 7.3: *Normals orientation in each isosceles tetrahedron.*

define the Livine-Speziale coherent intertwiner coefficient in the recoupling channel i with all outgoing links as:

$$\psi_i(\tau) = \sum_{m_a} \begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{(i)} \prod_{a=1}^{4} D^{j_a}_{m_a, j_a}(\tau)$$

$$= \tag{7.16}$$



,

where $D^j_{m,j}(\tau)$ has been defined in (7.15). The coefficient (7.16) encodes the orientation of the normals in the final amplitude. The definition of the 4jm Wigner symbol is reported in Appendix G.1.

7.4 Transition amplitude

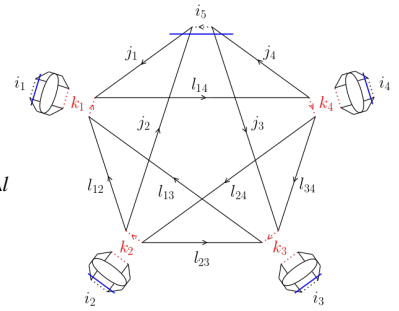
7.4.1 The EPRL vertex amplitude

We write the EPRL vertex amplitude using the graphical notation discussed in detail in [9]:

$$V_\gamma\left(j_f,\, i_e; \Delta l\right) = \tag{7.17}$$

$$= \sum_{\substack{j_f \le l_f \le j_f + \Delta l \\ k_a}} \left(\prod_{a=1}^{4} d_{k_a} B_4(j_f, l_f; i_a, k_a; \gamma) \right) \{15j\}_{l_f, k_a}$$

$$= \sum_{j_f \le l_f \le j_f + \Delta l}$$



,

where $a = 1 \dots 4$, $e = 1 \dots 5$, $q = 2 \dots 5$. The dependence of the amplitude on the Barbero-Immirzi parameter γ has been indicated using a label. The definition of the 15j Wigner symbol is reported in Appendix G.1, while the B_4 function is defined in Appendix E.3. We compute the EPRL vertices (7.17) with the numerical framework `sl2cfoam-next` [48]. We define the coherent amplitude as the vertex amplitude (7.17) contracted with coherent states coefficients (7.16) over all nodes except one[1]:

$$V^{coh}_{\gamma, n_f}\left(j_f, i_5; \Delta l\right) = \sum_{i_a} V \left(\prod_{a=1}^{4} d_{i_a} \psi_{i_a}(\tau_f) \right) . \tag{7.18}$$

The graphical notation of the coherent amplitude (7.18) is easily obtained from (7.17) and (7.16). We do not report it explicitly for the single vertex amplitude. Instead, we use it directly in the Black-to-White hole transition amplitude described in Section 7.4.2.

7.4.2 The Black-to-White hole transition amplitude

The Black-to-White hole transition amplitude is obtained by contracting the amplitude associated with the spinfoam described in Section 7.3.1 with the coherent boundary state described in Section 7.3.2, according to the usual procedure in covariant LQG [1]. We refer to the original article [18] for a description of all the necessary steps.

With the definition of the EPRL vertex amplitude (7.17), it is possible to write the Black-to-White hole transition amplitude in a suitable form for a numerical evaluation, transforming the original 24j Wigner symbol into the contraction of two (linear superposition of) 15j symbols

[1] The contraction over all nodes of the vertex is usually considered, but for the present context the definition (7.18) is more convenient

over a bulk intertwiner. Aside from the normals to the boundary tetrahedra, each link has an associated spin j and boost angle ζ. In [18], it was shown that when condition (7.4) is satisfied, the dependence on the parameters m, T is decoupled from the combinatorial structure of the two-complex. This greatly simplifies the numerical evaluation and leads to a factorization of the amplitude in the form of (a spin-sum over) a "weight function" w that multiplies the factor associated with the contraction of the two vertex amplitudes. We define the weight function as:

$$w_\alpha(j_a, j^\pm_{ab}, j_\pm, j_0, \zeta_\pm, \zeta_0) = c_\alpha\,(j_\pm, j_0)\left(\prod_{a=1}^{4} d_{j_a} e^{-\frac{j_0^\alpha}{2}(j_a - j_0)^2} e^{i\gamma\zeta_0 j_a}\right)\left(\prod_{ab,\pm} d_{j^\pm_{ab}} e^{-\frac{j_\pm^\alpha}{2}(j^\pm_{ab} - j_\pm)^2} e^{i\gamma\zeta_\pm j^\pm_{ab}}\right), \tag{7.19}$$

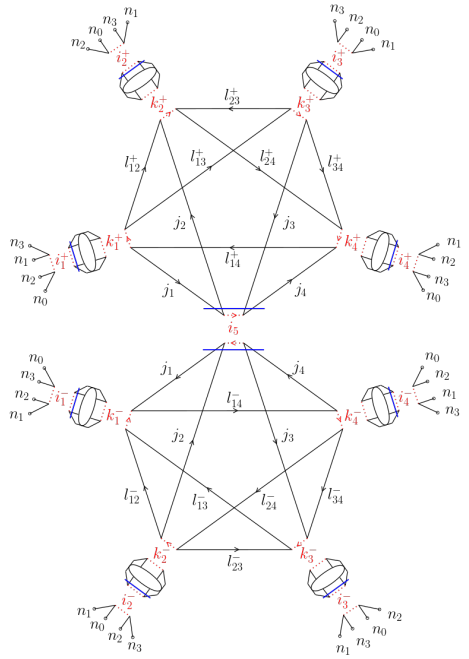
where $b = 2, 3, 4$, $a \neq b$. The data (j_0, ζ_0) label the 4 angular links, while $(j_\pm, \zeta_\pm)$ are associated with the radial links. The normalization factor inherited from the boundary state is:

$$c_\alpha\,(j_\pm, j_0) = \left(e^{\frac{j_0^{\alpha+2}}{2}}\right)^4 \left(e^{\frac{j_\pm^{\alpha+2}}{2}}\right)^{12}, \tag{7.20}$$

which corresponds to the last factor in (7.10) inherited by each link. We write the Black-to-White hole transition amplitude as:

$$W_\alpha(j_\pm, j_0, \zeta_\pm, \zeta_0; \Delta l) = \sum_{j^\pm_{ab}, j_a} w_\alpha \left(\sum_{i_5} d_{i_5} \prod_{\pm} V^{coh}_{\gamma, n^\pm_f}(j^\pm_{ab}, j_a, i_5; \Delta l)\right)$$

$$= \sum_{j^\pm_{ab}, j_a} w_\alpha \sum_{j^\pm_{ab} \leq l^\pm_{ab} \leq j^\pm_{ab} + \Delta l}$$



. (7.21)

The graphical notation in (7.21) emphasizes how the normals to the tetrahedra discussed in Section 7.3.4 are associated with the faces of the triangulation discussed in Section 7.3. The intertwiner i_5 is dual to tetrahedron τ_0 in Figure 7.2. In [219], we provide a Mathematica notebook to re-construct the full geometry (we thank Pietro Dona for the help with the notebook). Finally, we emphasize that the compact notation used in the first line of (7.21) does not specify if each link is a source or a target, but this is clarified with the graphical notation.

As mentioned in Section 7.3, the external geometry is entirely defined by the parameters m, T. Therefore, these two parameters entirely determine the amplitude (7.21). The relationship between these and the variables $j_0, j_\pm, \gamma, \zeta_\pm, \zeta_0$ is provided by the following relations:

$$j_0 = \frac{m^2\left(1 + e^{-\frac{T}{2m}}\right)^2}{2\gamma}\,, \tag{7.22}$$

$$j_\pm = \frac{j_0}{\sqrt{6}}\,, \tag{7.23}$$

$$\zeta_0 = \frac{T}{2m}\,, \tag{7.24}$$

$$\zeta_\pm = \mp\frac{32}{9}\sqrt{6}\,. \tag{7.25}$$

Despite the equality sign, in equation (7.22) it was used the well-known approximation $A = 8\pi\gamma\sqrt{j(j+1)} \approx 8\pi\gamma j$. A few comments are in order.

The discrete nature of spin has interesting consequences. For example, relation (7.23) implies that j_0 and $j_\pm$ cannot be both half-integer numbers. These are the terms on which the boundary coherent states (7.2) are peaked. Furthermore, triangular inequalities impose constraints on the allowed spins configurations.

7.5 Computing the amplitude

7.5.1 The numerical algorithm

The algorithm to calculate the black-to-white hole transition amplitude (7.21) as a function of T can be divided into three main steps. The strategy is similar to the one outlined in [9]. The core idea is to separate the computation of the EPRL vertex amplitudes (7.17) from the contraction along the intertwiners of each vertex, which is typically much less resource-demanding than the former. The code used for all calculations in this paper is public and available on `GitHub` [219].

The first step is to pre-calculate all the necessary EPRL vertex tensors (7.17). With the term "tensor," we refer to the multidimensional array consisting of the vertex amplitude computed for all the possible values of intertwiners. The flowchart is reported in 5. This is the most demanding step regarding computational resources and time complexity. The calculation of the vertex tensors has been performed with the `sl2cfoam-next` library [48] on the *Cedar*, *Graham* and *Narval* Compute Canada superclusters. We employed a hybrid parallelization scheme, distributing the workload on multiple processes and eventually exploiting various threads for each task. In [219], we also provide a code that automatically distributes the calculation of the vertex tensors to multiple machines. In Figure 7.4 we report the computational time of algorithm 5 for $j_0^{min} = 1$, $j_0^{max} = 5$, $\Delta l^{max} = 10$, $K_0 = 0.5$, $K_\pm = 0.5$. Each curve represents the seconds required to compute all the vertex amplitudes (7.17) centered around different spins configurations j_0, $j_\pm$. In the top panel, we show the results for $\gamma = 1$, while in the bottom one, we report the time for $\gamma = 5$. The two cases have approximately the same trend, even if the calculation for larger values of the Barbero-Immirzi parameter requires more time. The

Algorithm 5 Part 1: computing the EPRL vertices

1: At fixed Immirzi constant γ, choose the parameters α, K_0, $K_\pm$, j_0^{min}, j_0^{max} as described in Section 7.3.3
2: Set a maximum value Δl^{max}
3: **for** $j_0 \in \{j_0^{min}, j_0^{min} + \frac{1}{2}, \dots j_0^{max}\}$ **do**
4: Calculate $j_\pm$ from (7.23) and round to the nearest half-integer
5: **for** $\Delta l \in \{0, 1 \dots \Delta l^{max}\}$ **do**
6: Compute all the EPRL vertex amplitudes (7.17) with $j_a \in [j_0 - K_0,\ j_0 + K_0]$, $j_{ab} \in [j_\pm - K_\pm,\ j_\pm + K_\pm]$
7: Dump the vertices to disk
8: **end**
9: **end**

results in Figure 7.4 were estimated by distributing the computation of the vertices over 64 CPUs AMD Rome 7532 @ 2.40 GHz 256M cache L3.

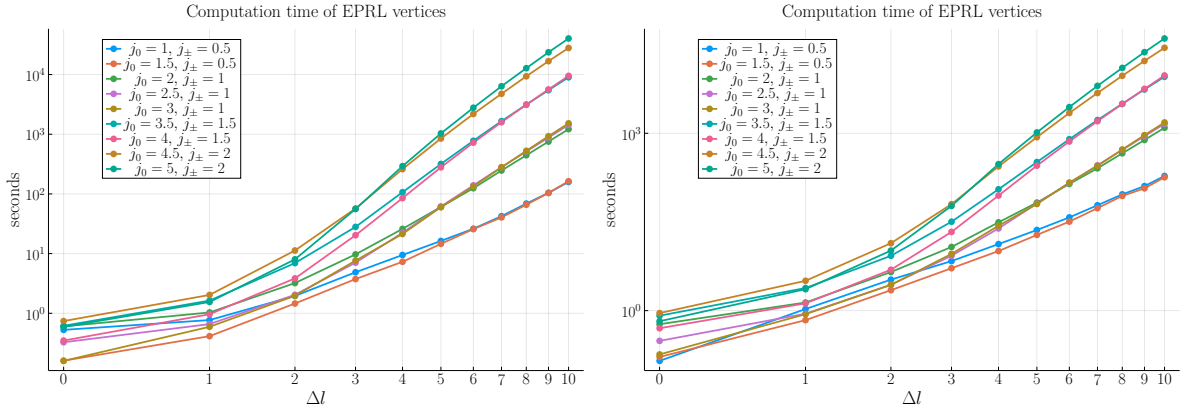


Figure 7.4: *Log-log plot of computational time required for algorithm 5.* ***Top:*** *case* $\gamma = 1$. ***Bottom:*** *case* $\gamma = 5$.

The second step consists in contracting the stored vertex tensors. The contraction is performed between the vertices and the coherent state coefficients (7.16) according to the spinfoam structure described in Section 7.3. This is illustrated in the flowchart 6. The final result of this step is a set of complex numbers, which correspond to the term in round brackets in (7.21) multiplying the weight factor w. This represents the pure "spinfoam contribution" to the amplitude, which is contracted with the weight factor of the boundary states.

The third and final step consists in computing the weight factor (7.19) and assembling the amplitude by retrieving all the pieces previously computed. As noticed in [18], the amplitude (7.21) is periodic in the extrinsic curvature angle ζ_0 with period $\frac{4\pi m}{\gamma}$ in T. This is a consequence of the discretization discussed in Section 7.3.1. Following the strategy of [204, 206, 209], we restrict the validity of the amplitude (7.21) to a single period over T. We choose a parameter

Algorithm 6 Part 2: contracting the EPRL vertices

for the spins configurations considered in algorithm 5 **do**
  Compute all the coherent state coefficients (7.16) with $j_a \in [j_0 - K_0,\ j_0 + K_0]$, $j_{ab} \in [j_\pm - K_\pm,\ j_\pm + K_\pm]$
  for $\Delta l \in \{0, 1 \dots \Delta l^{max}\}$ **do**
    Retrieve the amplitudes stored during algorithm 5 and load them into memory
    Contract the vertices with the coherent states to obtain two coherent amplitudes (7.18) as in (7.21)
    Contract the coherent amplitudes along i_5
    Dump the results to disk
  end
end

$N >> 1$ and divide the interval $\left[0, \frac{4\pi m}{\gamma}\right]$ into N equal sub-intervals with constant T:

$$0 \equiv T_0 < T_1 < T_2 \cdots < T_N \equiv \frac{4\pi m}{\gamma}\ ,\ N >> 1\ . \tag{7.26}$$

The weight factor is computed in each sub-interval, and the amplitude is assembled by retrieving and assembling all pieces. The value of m is computed using (7.22) disregarding the T dependence, as the term $e^{-\frac{T}{2m}}$ rapidly becomes negligible as a function of T. As shown in 7.6.1, neglecting this term results in a shift of the crossing time, but it does not alter the functional dependence on T. The flowchart is shown in 7. The final result of algorithm 7 is a set of

Algorithm 7 Part 3: Assembling the B-W amplitude

for the spins configurations considered in algorithm 5 **do**
  Compute m using (7.22)
  Choose a parameter $N >> 1$ and divide the first period in T according to (7.26)
  for each sub-interval **do**
    Compute and store the weight factor (7.19)
    Assemble the amplitude (7.21) retrieving the data stored in algorithm 6
  end
  Dump the amplitudes to disk
end

amplitudes:

$$\{W_{T_i}\}_{i=0}^{N} \equiv \{W_{T_0}, W_{T_1} \dots W_{T_N}\}\ ,\ N >> 1 \tag{7.27}$$

corresponding to the partition (7.26), which can be used to compute the physical observables depending on the amplitude (7.21) as a function of T. We discuss one example in Section 7.6. In figure 7.5, we display the results of the (rescaled) amplitude computed using the algorithm 7 and $N = 100$ in the partition (7.26). Each point corresponds to the squared absolute value of the elements in the partition (7.27). Notice that in Figure 7.5 the amplitude is rescaled so that the `julia` package `Plots.jl` displays the value correctly. A few comments are in order.

The exact value of the amplitude is recovered in the limit $\Delta l \longrightarrow \infty$. This parameter is introduced in the EPRL vertex amplitude (7.17) as a homogeneous truncation to approximate the unbounded convergent sums over the virtual spins l_f [42]. The role of this parameter has been deeply discussed in many papers focusing on numerical computations of spinfoam amplitudes [9, 47, 48, 72, 71, 133]. The amplitude becomes constant as m increases since the quantum fluctuations are suppressed as the spin grows. That is, approaching the semiclassical limit as discussed in 7.3.3. With the expression "quantum fluctuations," we refer to the terms in the sum (7.5) defining the extrinsic coherent state with $j \neq j_m$. In (7.21) we have such a sum for each link, where j_m corresponds to j_0 in the case of angular links and $j_\pm$ for the radial ones.

Higher orders in the vertex expansion are necessary to investigate larger T values and resolve (at least partially) the periodicity of amplitude (7.21) in T. An example of complete derivation of the black-to-white hole transition amplitude with a finer triangulation has been derived in [205]. Unfortunately, the level of complexity in the numerical evaluation of the amplitude grows very quickly as the triangulation refinement increases.

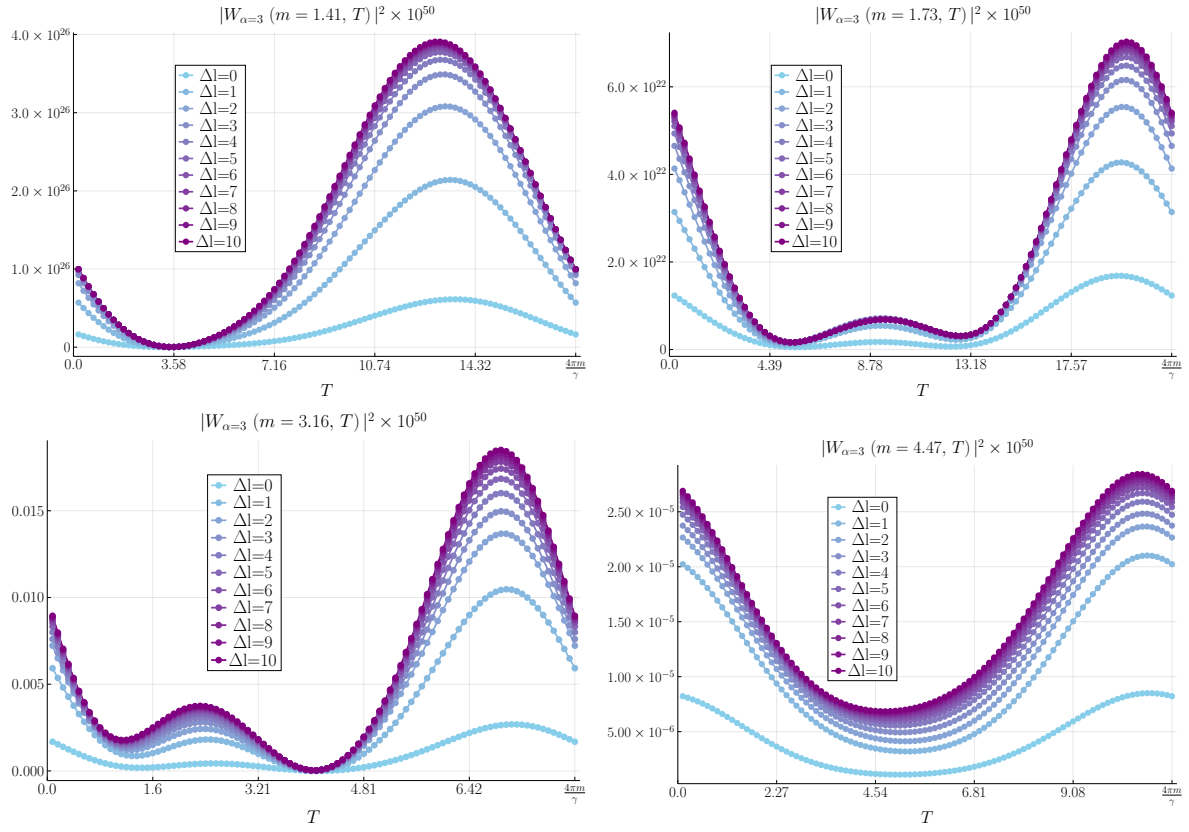


Figure 7.5: *Black-to-White hole transition amplitude* (7.21) *computed with the algorithm described in Section 7.5. In the partition* (7.27)*, it was used* $N = 100$*. The truncation parameter's value* $\Delta l = 10$ *reasonably approximates the amplitude.* ***Top:*** *case* $\gamma = 1$*.* ***Bottom:*** *case* $\gamma = 5$*.*

7.6 The crossing time

In this Section, we estimate the crossing time both numerically and analytically. It represents the characteristic time scale for the transition when it takes place. For an accurate and comprehensive physical description of this observable (as well as other time scales involved in the tunneling process), we refer to [204]. In the following, we do not explicitly indicate the dependence on the truncation parameter Δl, implying that the latter has been fixed to $\Delta l = 10$ providing a reasonable amplitude estimate. According to the probabilistic interpretation of the transition amplitude developed by Oeckl [220, 221], explicitly applied to the black-to-white hole transition in [222], we first define a conditional probability distribution:

$$P_\alpha(m|T) = \frac{\mu_\alpha(m,T)\,|W_\alpha(m,T)|^2}{\int_0^\infty dT\;\mu_\alpha(m,T)\,|W_\alpha(m,T)|^2}\;, \tag{7.28}$$

which is interpreted as the conditional probability for measuring T at a given mass m. The coefficient μ provides the measure for the identity resolution of the extrinsic boundary states. The necessity for this factor was first pointed out in [222]. It was explicitly computed in [209, 206] in the twisted geometry parametrization. For a single link, this reads:

$$\nu_\alpha(j) = \frac{j^{-\frac{\alpha}{2}}(1+2j)}{64\pi^{\frac{7}{2}}}\left(e^{-j^{\alpha+2}} - e^{-j^\alpha(1+j)^2}\right)\;, \tag{7.29}$$

where j is the spin attached to the link. The coefficient μ in (7.28) is defined as the product of a factor (7.29) for each boundary link. In the present context, there are just two types of links. That is four angular links with spin j_0 and 12 radial links with spins $j_\pm$.

We estimate the crossing time as the expectation value of T over the conditional probability distribution (7.28):

$$\tau_\alpha(m) = \int_0^\infty dT\;T\;P_\alpha(m|T)\;. \tag{7.30}$$

The crossing time (7.30) is well-defined since integrating (7.28) along T at fixed m gives a total (conditional) probability of 1, regardless of the constant factor multiplying the amplitude (7.21).

7.6.1 Crossing time estimate in large spins regime

As discussed in Section 7.3.3, for large values of spins, we can set the cut-off K equal to zero in the sum over spins in (7.5). From (7.29) and (7.22), it is easy to see that the measure coefficient μ in the conditional probability distribution (7.28) acts as the Heaviside step function:

$$\mu_\alpha(m,T) \approx \theta(T - T_*)\;, \tag{7.31}$$

where T_* is large enough so that j_0 in (7.22) is rounded to the lowest half-integer. Focusing on the first period of the amplitude in T, along with condition (7.31), this results in:

$$P_\alpha(m|T) \approx \begin{cases} 0 & \text{for } T \in [0, T_*] \\ \left(\frac{4\pi m}{\gamma} - T_*\right)^{-1} & \text{for } T \in \left[T_*, \frac{4\pi m}{\gamma}\right] \end{cases}\;. \tag{7.32}$$

It is immediate to compute the crossing time (7.30) with (7.32):

$$\tau(m) \approx \frac{m^2\pi^2}{\gamma^2\left(\frac{4\pi m}{\gamma} - T_*\right)} \,. \tag{7.33}$$

We have recovered the linear scaling of the crossing time as a function of the mass m. If we completely neglect the T dependence in the boundary data (7.22), this is equivalent to consider $T_* = 0$, which implies:

$$\tau(m) \approx \frac{2\pi m}{\gamma} \quad \text{for } T_* = 0 \,. \tag{7.34}$$

This is the same estimate originally obtained in [204, 206] in the semi-classical regime with $\alpha = -\frac{1}{2}$ using the stationary phase approximation for the amplitude. These two results considered together emphasize that the scaling of the crossing time as a function of T does not depend on the α parameter, which is used to peak to coherent states as discussed in Section 7.3.3. Using the range $\alpha > 0$ makes the calculation remarkably simpler. Since such a regime implies that the extrinsic curvature becomes more and more spread as the spin increases, we infer that the scaling of the crossing time only depends on the intrinsic geometry of the black-to-white hole scenario rather than the extrinsic geometry.

7.6.2 The crossing time computation

Finally, we compute the crossing time (7.6) using the numerical approach described in Section 7.5. We consider the parameters $\gamma = 1,5$, $\alpha = 3,\dots 6$, $K_0 = 0.5$, $K_\pm = 0.5$, $j_0^{min} = 1.5$, $j_0^{max} = 5$, $\Delta l^{max} = 10$, and $N = 100$ for the algorithm described in Section 7.5. This choice of parameters is such that approximation (7.5) is reliable. After obtaining the amplitudes (7.27), the conditional probability distribution and the crossing time (7.30) can be evaluated using the trapezoidal rule to compute the integrals over T. The result is shown in Figure 7.6 for different parameter α values, which balances the quantum spread of the boundary states as discussed in Section 7.3.3. The dashed curve represents the "semiclassical" (in the sense of just the intrinsic geometry) asymptotic estimate (7.34).

The crossing time tends to become closer and closer to the asymptotic estimate as a function of m as α increases. The quantum fluctuations emerge for very small values of the black hole mass.

7.7 Conclusions

In this paper, we presented and described an algorithm to calculate the full black-to-white hole transition amplitude numerically, using the covariant "spinfoam formulation" in the Lorentzian EPRL model and high-performance computing methods. We considered the triangulation originally introduced in [18], for which a complete numerical evaluation was still missing in the literature. We explicitly applied the algorithm to compute a relevant physical observable corresponding to the crossing time of the transition. We also discussed a straightforward analytical approach to estimate the same quantity alternative to the one currently present in the spinfoam literature, which is based on the stationary phase technique [209, 206, 204].

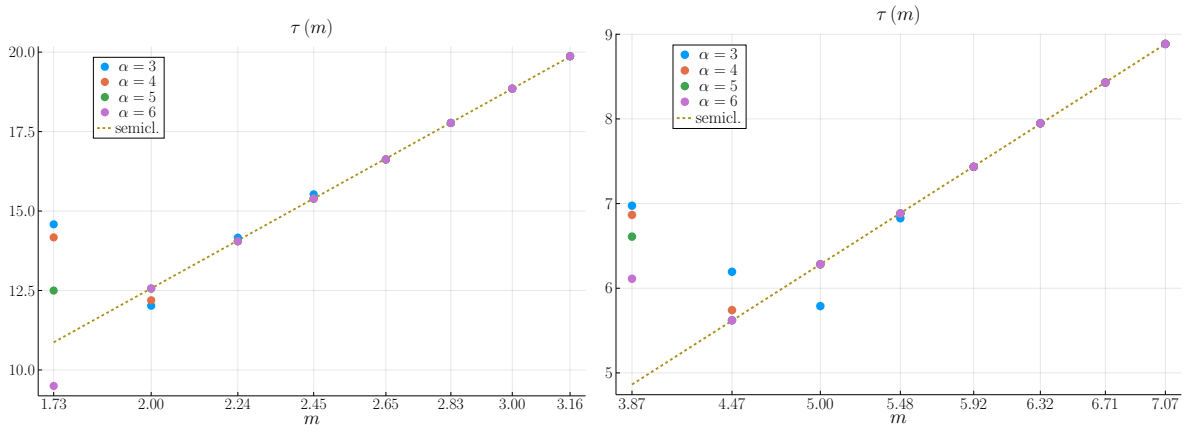


Figure 7.6: *Crossing time* (7.30) *evaluated with the amplitudes* (7.27)*, computed with the algorithm discussed in Section 7.5. The asymptotic estimate corresponds to* (7.34)*.* ***Top:*** *case* $\gamma = 1$*.* ***Bottom:*** *case* $\gamma = 5$*.*

Compared to the analytical calculation in the literature, we tuned the boundary state in this paper so that the limit for large spins corresponds to an infinite spread of the extrinsic curvature. The estimate of the crossing time (analytical and numerical) performed in this paper shows that the estimate is the same. Physically, this emphasizes that the crossing time of the black-to-white hole transition amplitude does not depend on extrinsic curvature. On the contrary, it appears to be a feature of intrinsic geometry. Therefore, this result adds new information despite being in excellent agreement with the previous estimates of the same physical observable in the literature, including estimates obtained by different communities [223, 224].

This work hopes to be the first step in connecting the usage of high-performance computing techniques in loop quantum gravity with the study of the quantum tunneling process between a black hole and a white hole. In recent years, remarkable advances have been made in developing computational methods for spinfoam calculations to study refined triangulation [138, 72, 32]. Numerical approaches based on (Markov Chain) Monte Carlo methods combined with the vertex decomposition (7.17) recently allowed computing EPRL spinfoam amplitudes and observables for highly non-trivial triangulations, potentially containing infrared bubbles [133, 71]. Two examples are the vertex renormalization (or "5-1 Pachner move") amplitude [138] and the star model [78]. The former contains a bubble with 10 internal faces. We hope that our work will provide a valuable and encouraging ground for progressing in the development of spinfoams more refined than the one considered in this paper. At the same time, we hope that the considerations about the independence of the crossing time scaling from the extrinsic geometry will encourage the investigations of new probabilistic interpretations of the tunneling process.

Conclusions

This Ph.D. thesis has explored a small part of the LQG frontier, attempting to push the boundaries of the research by developing new computational techniques based on HPC and enabling investigations into quantum gravitational phenomena. The collection of integrated articles presented here highlights the current and future role of scientific computing in advancing the covariant loop quantum gravity field. I believe the prospects for further advancements in HPC-assisted LQG research are promising. The ever-increasing computational power of modern supercomputers and the exciting perspectives of quantum computing, coupled with ongoing developments in algorithms and numerical methods, opens new avenues for exploration. These developments will enable the study of previously inaccessible physical regimes and the investigation of quantum phenomena on larger scales.

Compared to 3-4 years ago, we now have the numerical tools and algorithms to calculate rather complicated and extended EPRL amplitudes, once considered inaccessible. For example, concerning infrared divergences, one of the most noteworthy results obtained is the introduction of a robust scheme to investigate amplitudes associated with spinfoams containing bubbles numerically. Before the algorithm discussed in Chapter 2 was introduced, the situation was somewhat uncertain, with significantly different estimates regarding self-energy divergence [46, 102]. By resorting to Monte Carlo methods, as shown in Chapter 4, it is now possible to test the divergence of a spinfoam amplitude with high precision without compromising the computational effort. The results of Chapters 3 and 4 indicate that there is still much to discover about EPRL spinfoam amplitudes with many internal faces.

Specifically, for the interested reader in continuing my line of research, I found some questions exciting and could not answer during my Ph.D. For example, "which are the EPRL spinfoam diagrams containing infrared bubbles that are convergent?". In other words, can we establish a priori if an EPRL diagram with specific connectivity has a finite amplitude? As discussed in Chapter 4, the destructive interference between the oscillations due to the Lorentz group in EPRL is most probably responsible for the amplitude's finiteness in a few cases (as in the examples considered in Chapter 3). This is especially manifest if compared to the topological model, where the interferences are more suppressed, usually resulting in highly divergent amplitudes. Therefore, the question of understanding the actual mechanism leading to the divergence of the amplitude naturally arises. A possible systematic way of investigating it could be to apply the numerical methods developed in Chapters 2, 3, and 4 to study many spinfoam diagrams with different connectivity. At the same time, analytical studies using $SL(2,\mathbf{C})$ group theory could help solve this mystery. Another question is "how can we re-normalize divergent spinfoam diagrams?" This is the core of the large-volume infrared divergences "issue" in LQG. I do not have specific suggestions or intuition. The most natural approach seems to develop

and apply techniques inspired by renormalization in QFT. Of course, such a technique must be adapted to the specific and quite peculiar spinfoam formalism.

In spinfoam cosmology, the refinement approach based on Markov Chain Monte Carlo discussed in Chapter 5 could be applied to different graphs with richer connectivity than the star model. This would allow the exploration of possible correlations between universe regions using other cosmological models. This research direction has already been started in Chapter 6, which considers a regular triangulation of a 3-sphere, although it only employs a topological model. Extending the study of the 16-cell to the Lorentzian EPRL case is not straightforward in the deep quantum regime. Apart from the computational perspectives, it would be interesting to expand the conceptual frame of the cosmological spinfoam scenario further by incorporating matter fields.

We also mention that work is currently in progress towards using machine learning in spinfoams [225]. The idea is inspired by the recent results obtained in QCD [226, 227], where a Markov chain update scheme has been successfully employed for Monte Carlo sampling in lattice field theories using a machine-learned flow-based generative model. The current attempt to introduce machine learning algorithms in covariant LQG has been focused on replacing Markov Chain Monte Carlo methods described in Chapter 5 with Generative Flow Networks (GFlowNet) [228, 229, 230]. Although the initial expectancy was to speed up the original Monte Carlo-based approach, the latest results seem to indicate that this is not the case. Nevertheless, this new research direction is just at the primordial stages. I hope that computational scientists will address unexplored and visionary questions by using machine learning-inspired algorithms in the future of spinfoams.

Regarding the gravitational black-hole quantum tunneling, a compelling and satisfying project would be to investigate the black-to-white hole transition amplitude using a spinfoam more refined than the one considered in Chapter 7. I believe the current concrete proposal in the literature [205] is too complicated for the current status of numerical methods in LQG, even considering Monte Carlo methods and HPC resources. Nevertheless, one can construct many possible spinfoams that are more refined than the simple one studied in Chapter 7. Other prospects concerning the black-to-white hole scenario have already been mentioned in the conclusions of Chapter 7.

Finally, I dedicate a small paragraph regarding my overall considerations on covariant LQG. These maturated over more than three years devoted to doing scientific research in the field. On the one hand, the prospects for future research in this field are exciting, with HPC-assisted LQG opening up new frontiers for theoretical physics, cosmology, and our understanding of the universe. On the other hand, we have (yet) no solid empirical support for any theory of quantum gravity. The current limits of experimental physics are way too strict to provide viable lab experiments to test each view's unique features and predictions. Accessible energy scales are many orders of magnitude distant from the Planck scale. In this regard, the situation appears entirely different from the previous century, in which the experiments were crucial to building the theoretical standard model of particle physics. The lack of experimental data and direct empirical evidence contributed to the development of multiple theories (apparently non-compatible) describing the quantum gravity regime: string theory, LQG, asymptotic safety, causal dynamical triangulation, etc. There appear to be many distinct approaches and

strategies for each theory, usually employed by different communities. Despite this fragmented scenario, I was privileged to be part of the LQG community. The latter is full of intelligent, talented, and passionate scientists. I hope the "numerical spinfoam" community will expand and mature and hopefully find empirical support for the theory soon. I hope my modest contributions during the last few years could serve the purpose.

Bibliography

[1] C. Rovelli and F. Vidotto. *Covariant Loop Quantum Gravity*. Cambridge University Press, Cambridge, 2015.

[2] Alejandro Perez. *Introduction to Loop Quantum Gravity and Spin Foams*. *arXiv*, 2005.

[3] Abhay Ashtekar and Jerzy Lewandowski. Background independent quantum gravity: A status report. *Classical and Quantum Gravity*, 21:R53, 2004.

[4] Carlo Rovelli. *Quantum Gravity*. Cambridge University Press, Cambridge, 2004.

[5] Thomas Thiemann. *Modern Canonical Quantum General Relativity*. Cambridge Monographs on Mathematical Physics. Cambridge University Press, 2007.

[6] Abhay Ashtekar and Jorge Pullin, editors. *Loop Quantum Gravity: The First 30 Years*, volume 4 of *100 Years of General Relativity*. World Scientific, 2017.

[7] Seth K. Asante, Bianca Dittrich, and Sebastian Steinhaus. *Spin foams, Refinement limit and Renormalization*. *arXiv*, 2022.

[8] Rodolfo Gambini and Jorge Pullin. *Introduction to loop quantum gravity*. Oxford University Press, Oxford, 2010.

[9] Pietro Donà and Pietropaolo Frisoni. How-to compute EPRL spin foam amplitudes. *Universe*, 8(4):208, 2022.

[10] Pietropaolo Frisoni. *Introduction to Loop Quantum Gravity: Rovelli's lectures on LQG*. 2023.

[11] Carlo Rovelli and Lee Smolin. Discreteness of area and volume in quantum gravity. *Nuclear Physics*, B442:593–622, 1995.

[12] R. Arnowitt, S. Deser, and C. W. Misner. Dynamical structure and definition of energy in general relativity. *Physical Review*, 116:1322–1330, 1959.

[13] R. Arnowitt and S. Deser. Quantum theory of gravitation: General formulation and linearized theory. *Physical Review*, 113:745–750, 1959.

[14] Abhay Ashtekar and Parampreet Singh. Loop Quantum Cosmology: A Status Report. *Classical and Quantum Gravity*, 28:213001, 2011.

[15] Abhay Ashtekar, Miguel Campiglia, and Adam Henderson. Loop Quantum Cosmology and Spin Foams. *Physics Letters*, B681:347–352, 2009.

[16] Francesca Vidotto. Spinfoam Cosmology: quantum cosmology from the full theory. *Journal of Physics: Conference Series*, 314(012049), 2010.

[17] Eugenio Bianchi, Carlo Rovelli, and Francesca Vidotto. Towards Spinfoam Cosmology. *Physical Review*, D82:84035, 2010.

[18] Marios Christodoulou, Carlo Rovelli, Simone Speziale, and Ilya Vilensky. Planck star tunneling time: An astrophysically relevant observable from background-free quantum gravity. *Physical Review D*, 94, 2016.

[19] Alejandro Perez. *Black holes in loop quantum gravity*. 80(12):126901, 2017.

[20] Carlo Rovelli. Loop Quantum Gravity and Black Hole Physics. *arXiv.org*, gr-qc, 1996.

[21] Carlo Rovelli. Black Hole Entropy from Loop Quantum Gravity. *Physical Review Letters*, 77(16):3288–3291, 1996.

[22] Abhay Ashtekar, Tomasz Pawlowski, and Parampreet Singh. Quantum nature of the big bang: An analytical and numerical investigation. I. *Physical Review*, D73:124038, 2006.

[23] John W Barrett and Louis Crane. Relativistic spin networks and quantum gravity. *Journal of Mathematical Physics*, 39:3296–3302, 1998.

[24] Mariano Celada, Diego Gonzalez, and Merced Montesinos. Bf gravity. *Classical and Quantum Gravity*, 33(21):213001, 2016.

[25] Jonathan Engle, Etera Livine, Roberto Pereira, and Carlo Rovelli. LQG vertex with finite Immirzi parameter. *Nuclear Physics*, B799:136–149, 2008.

[26] Jonathan Engle, Roberto Pereira, and Carlo Rovelli. Flipped spinfoam vertex and loop gravity. *arXiv.org*, gr-qc, 2007.

[27] Jonathan Engle and Carlo Rovelli. The accidental flatness constraint does not mean a wrong classical limit. *Classical and Quantum Gravity*, 39(11):117001, 2022.

[28] Laurent Freidel and Kirill Krasnov. A New Spin Foam Model for 4d Gravity. *Classical and Quantum Gravity*, 25:125018, 2008.

[29] John W Barrett, Richard J Dowdall, Winston J Fairbairn, Frank Hellmann, and Roberto Pereira. Lorentzian spin foam amplitudes: graphical calculus and asymptotics. *Classical and Quantum Gravity*, 27:165009, 2010.

[30] Pietro Dona and Simone Speziale. Asymptotics of lowest unitary SL(2,c) invariants on graphs. *Physical Review D*, 102(8), 2020.

[31] Pietro Donà, Francesco Gozzini, and Giorgio Sarno. Numerical analysis of spin foam dynamics and the flatness problem. *Physical Review D*, 102(10):106003, 2020.

[32] Muxin Han, Zichang Huang, Hongguang Liu, and Dongxue Qu. Complex critical points and curved geometries in four-dimensional lorentzian spinfoam quantum gravity. *Physical Review D*, 106(4), 2022.

[33] Pietro Donà . Geometry from local flatness in lorentzian spin foam theories. *Physical Review D*, 107(6), 2023.

[34] Elena Magliaro and Claudio Perini. Regge gravity from spinfoams. *International Journal of Modern Physics D*, 22:1350001, 2013.

[35] Muxin Han. Cosmological Constant in LQG Vertex Amplitude. *Physical Review D*, 84:64010, 2011.

[36] Muxin Han. Four-dimensional spinfoam quantum gravity with a cosmological constant: Finiteness and semiclassical limit. *Physical Review D*, 104(10), 2021.

[37] Florian Conrady and Jeff Hnybida. A spin foam model for general lorentzian 4-geometries. *Classical and Quantum Gravity*, 27(18):185011, 2010.

[38] J C Baez, J D Christensen, T R Halford, and D C Tsang. Spin Foam Models of Riemannian Quantum Gravity. *General Relativity and Gravitation*, 19:4627–4648, 2002.

[39] J C Baez, J D Christensen, and G Egan. Asymptotics of 10j symbols. *General Relativity and Gravitation*, 19:6489, 2002.

[40] J Daniel Christensen, Etera R Livine, and Simone Speziale. Numerical evidence of regularized correlations in spin foam gravity. *Physics Letters B*, 670:403–406, 2009.

[41] J Daniel Christensen, Igor Khavkine, Etera R Livine, and Simone Speziale. Sub-leading asymptotic behaviour of area correlations in the Barrett-Crane model. *Classical and Quantum Gravity*, 27:35012, 2010.

[42] Simone Speziale. Boosting Wigner's nj-symbols. *Journal of Mathematical Physics*, 58(3), 2017.

[43] W Ruhl. *The Lorentz group and harmonic analysis*. W.A. Benjamin, Inc, New York, 1970.

[44] Giorgio Sarno, Simone Speziale, and Gabriele V. Stagno. 2-vertex Lorentzian spin foam amplitudes for dipole transitions. *General Relativity and Gravitation*, 50(4), 2018.

[45] Pietro Donà, Marco Fanizza, Giorgio Sarno, and Simone Speziale. Numerical study of the lorentzian engle-pereira-rovelli-livine spin foam amplitude. *Physical Review D*, 100, 2019.

[46] Pietro Donà. Infrared divergences in the EPRL-FK Spin Foam model. *Classical and Quantum Gravity*, 35(17):175019, 2018.

[47] Pietro Donà and Giorgio Sarno. Numerical methods for EPRL spin foam transition amplitudes and Lorentzian recoupling theory. *General Relativity and Gravitation*, 50(10), 2018.

[48] Francesco Gozzini. A high-performance code for eprl spin foam amplitudes. *Classical and Quantum Gravity*, 38(22):225010, Oct 2021.

[49] Seth K. Asante, José Diogo Simão, and Sebastian Steinhaus. Spin-foams as semiclassical vertices: Gluing constraints and a hybrid algorithm. *Physical Review D*, 107(4), 2023.

[50] Courtney Allen, Florian Girelli, and Sebastian Steinhaus. Numerical evaluation of spin foam amplitudes beyond simplices. *Physical Review D*, 105(6), 2022.

[51] Seth K. Asante, Bianca Dittrich, and José Padua-Arguelles. Effective spin foam models for Lorentzian quantum gravity. *Classical and Quantum Gravity*, 38(19):195002, 2021.

[52] Bianca Dittrich, Sebastian Mizera, and Sebastian Steinhaus. Decorated tensor network renormalization for lattice gauge theories and spin foam models. *New Journal of Physics*, 18(5):053009, May 2016.

[53] Benjamin Bahr and Sebastian Steinhaus. Investigation of the spinfoam path integral with quantum cuboid intertwiners. *Physical Review D*, 93(10), May 2016.

[54] Valentin Bonzom and Matteo Smerlak. Bubble divergences: Sorting out topology from cell structure. *Annales Henri Poincaré*, 13(1):185–208, 2011.

[55] Aristide Baratin, Sylvain Carrozza, Daniele Oriti, James Ryan, and Matteo Smerlak. Melonic phase transition in group field theory. *Letters in Mathematical Physics*, 104(8):1003–1017, May 2014.

[56] Marco Finocchiaro and Daniele Oriti. Renormalization of group field theories for quantum gravity: New computations and some suggestions. *Frontiers in Physics*, 8, 2021.

[57] Muxin Han, Zichang Huang, Hongguang Liu, Dongxue Qu, and Yidun Wan. Spinfoam on a lefschetz thimble: Markov chain monte carlo computation of a lorentzian spinfoam propagator. *Physical Review D*, 103(8), Apr 2021.

[58] Reinier van der Meer, Zichang Huang, Malaquias Correa Anguita, Dongxue Qu, Peter Hooijschuur, Hongguang Liu, Muxin Han, Jelmer J. Renema, and Lior Cohen. Experimental simulation of loop quantum gravity on a photonic chip. *npj Quantum Information*, 9(1), apr 2023.

[59] Lior Cohen, Anthony J. Brady, Zichang Huang, Hongguang Liu, Dongxue Qu, Jonathan P. Dowling, and Muxin Han. Efficient simulation of loop quantum gravity: A scalable linear-optical approach. *Physical Review Letters*, 126(2), 2021.

[60] Benjamin Bahr and Sebastian Steinhaus. Hypercuboidal renormalization in spin foam quantum gravity. *Physical Review D*, 95(12):126006, 2017.

[61] Bianca Dittrich, Sebastian Mizera, and Sebastian Steinhaus. Decorated tensor network renormalization for lattice gauge theories and spin foam models. *New Journal of Physics*, 18(5):053009, 2016.

[62] William J. Cunningham, Bianca Dittrich, and Sebastian Steinhaus. Tensor Network Renormalization with Fusion Charges—Applications to 3D Lattice Gauge Theory. *Universe*, 6(7):97, 2020.

[63] Seth K. Asante, Bianca Dittrich, and Hal M. Haggard. Effective spin foam models for four-dimensional quantum gravity. *Physical Review Letters*, 125(23), 2020.

[64] Seth K Asante, Bianca Dittrich, and José Padua-Argüelles. Effective spin foam models for lorentzian quantum gravity. *Classical and Quantum Gravity*, 38(19):195002, 2021.

[65] Johanna N Borissova and Bianca Dittrich. Towards effective actions for the continuum limit of spin foams. *Classical and Quantum Gravity*, 40(10):105006, apr 2023.

[66] J Engle, W Kaminski, and J Oliveira. Addendum to 'EPRL/FK asymptotics and the flatness problem'. *Classical and Quantum Gravity*, 38(11):119401, 2021.

[67] Seth K. Asante, Bianca Dittrich, and Hal M. Haggard. Effective Spin Foam Models for Four-Dimensional Quantum Gravity. *Physical Review Letters*, 125(23):231301, 2020.

[68] John W. Barrett, Richard J. Dowdall, Winston J. Fairbairn, Frank Hellmann, and Roberto Pereira. Lorentzian spin foam amplitudes: graphical calculus and asymptotics. *Classical and Quantum Gravity*, 27, 2010.

[69] Michael P Reisenberger and Carlo Rovelli. *Sum over surfaces* form of loop quantum gravity. *Physical Review*, D56:3490–3508, 1997.

[70] Pietro Donà, Marco Fanizza, Giorgio Sarno, and Simone Speziale. Numerical study of the Lorentzian Engle-Pereira-Rovelli-Livine spin foam amplitude. *Physical Review D*, 100(10), 2019.

[71] Pietropaolo Frisoni, Francesco Gozzini, and Francesca Vidotto. Numerical analysis of the self-energy in covariant loop quantum gravity. *Physical Review D*, 105(10), 2022.

[72] Pietropaolo Frisoni. Studying the EPRL spinfoam self-energy. In *The Sixteenth Marcel Grossmann Meeting*. WORLD SCIENTIFIC, 2023.

[73] Courtney Allen, Florian Girelli, and Sebastian Steinhaus. Numerical evaluation of spin foam amplitudes beyond simplices. *Physical Review D*, 105(6), 2022.

[74] Alejandro Perez. The Spin-Foam Approach to Quantum Gravity. *Living Reviews in Relativity*, 16, 2013.

[75] Abhay Ashtekar and Eugenio Bianchi. A short review of loop quantum gravity. *Rept. Prog. Phys.*, 84(4):042001, 2021.

[76] Fabio D'Ambrosio, Marios Christodoulou, Pierre Martin-Dussaud, Carlo Rovelli, and Farshid Soltani. End of a black hole's evaporation. *Physical Review D*, 103(10), 2021.

[77] Francesco Gozzini and Francesca Vidotto. Primordial fluctuations from quantum gravity. *Frontiers in Astronomy and Space Sciences*, 7, Feb 2021.

[78] Pietropaolo Frisoni, Francesca Vidotto, and Francesco Gozzini. Markov chain monte carlo methods for graph refinement in spinfoam cosmology. *Classical and Quantum Gravity*, 2023.

[79] Jerzy F Plebanski. On the separation of Einsteinian substructures. *Journal of Mathematical Physics*, 18:2511–2520, 1977.

[80] John C Baez. An introduction to spin foam models of BF theory and quantum gravity. *Lecture Notes in Physics*, 543:25–94, 2000.

[81] Sören Holst. Barbero's Hamiltonian derived from a generalized Hilbert-Palatini action. *Physical Review D*, 53:5966–5969, 1996.

[82] Wojciech Kaminski, Marcin Kisielowski, and Jerzy Lewandowski. Spin-Foams for All Loop Quantum Gravity. *Classical and Quantum Gravity*, 27:95006, 2010.

[83] Eugenio Bianchi, Daniele Regoli, and Carlo Rovelli. Face amplitude of spinfoam quantum gravity. *Classical and Quantum Gravity*, 27:185009, 2010.

[84] Pierre Martin-Dussaud. A primer of group theory for loop quantum gravity and spinfoams. *General Relativity and Gravitation*, 51(9), Sep 2019.

[85] Jonathan Engle and Roberto Pereira. Regularization and finiteness of the Lorentzian LQG vertices. *Physical Review*, D79:84034, 2009.

[86] R. L. Anderson, R. Raczka, M. A. Rashid, and P. Winternitz. Clebsch-gordan coefficients for the coupling of sl(2,c) principal-series representations. *Journal of Mathematical Physics*, 11:1050–1058, 1970.

[87] G. A. Kerimov and I. A. Verdiev. Clebsch-Gordan Coefficients of the SL(2,c) Group. *Reports on Mathematical Physics.*, 13:315–326, 1978.

[88] Pietro Donà, Marco Fanizza, Pierre Martin-Dussaud, and Simone Speziale. Asymptotics of SL (2 ,C) coherent invariant tensors. *Communications in Mathematical Physics*, November 2021.

[89] Pietropaolo Frisoni and Pietro Dona. Github repository *How-To-SpinFoam-Amplitude*. `https://github.com/PietropaoloFrisoni/HowToSpinFoamAmplitude`, 2022.

[90] Giorgio Sarno and Pietro Dona. Github repository *SL2Cfoam*. `https://github.com/qg-cpt-marseille/sl2cfoam`, 2017.

[91] H. T. Johansson and C. Forssén. Fast and accurate evaluation of wigner $3j$, $6j$, and $9j$ symbols using prime factorization and multiword integer arithmetic. *SIAM Journal on Scientific Computing*, 38(1):A376–A384, Jan 2016.

[92] *Various*. Github repository *klib*. `https://github.com/attractivechaos/klib`, 2010.

[93] Torbjörn Granlund et al. GNU multiple precision arithmetic library 4.1.2, December 2002. `http://swox.com/gmp/`.

[94] Laurent Fousse et al. "mpfr: A multiple-precision binary floating-point library with correct rounding". RR5753, INRIA. 2005, pp.15. ffinria-00070266f.

[95] Andreas Enge, Mickaël Gastineau, Philippe Théveny, and Paul Zimmermann. *mpc — A library for multiprecision complex arithmetic with exact rounding*. INRIA, 1.1.0 edition, January 2018. `http://mpc.multiprecision.org/`.

[96] Francesco Gozzini. Github repository *SL2Cfoam-next*. `https://github.com/qg-cpt-marseille/sl2cfoam-next`, 2021.

[97] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

[98] L Susan Blackford, Antoine Petitet, Roldan Pozo, Karin Remington, R Clint Whaley, James Demmel, Jack Dongarra, Iain Duff, Sven Hammarling, Greg Henry, et al. An updated set of basic linear algebra subprograms (blas). *ACM Transactions on Mathematical Software*, 28(2):135–151, 2002.

[99] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. Cuda, release: 10.2.89, 2020.

[100] Tim Besard, Christophe Foket, and Bjorn De Sutter. Effective extensible programming: Unleashing Julia on GPUs. *IEEE Transactions on Parallel and Distributed Systems*, 2018.

[101] Tim Besard, Valentin Churavy, Alan Edelman, and Bjorn De Sutter. Rapid software prototyping for heterogeneous and distributed platforms. *Advances in Engineering Software*, 132:29–46, 2019.

[102] Aldo Riello. Self-Energy of the Lorentzian EPRL-FK Spin Foam Model of Quantum Gravity. *Physical Review*, D88:24011, 2013.

[103] Pietropaolo Frisoni. Github repository *Markov Chain Monte Carlo in spinfoams*. `https://github.com/PietropaoloFrisoni/Star-spinfoam-model`, 2022.

[104] Matthew Fishman, Steven R. White, and E. Miles Stoudenmire. The ITensor software library for tensor network calculations, 2020.

[105] A. C. Aitken. Xxv.—on bernoulli's numerical solution of algebraic equations. *Proceedings of the Royal Society of Edinburgh*, 46:289–305, 1927.

[106] Robert Oeckl. A 'general boundary' formulation for quantum mechanics and quantum gravity. *Physics Letters*, B575:318–324, 2003.

[107] Carlo Rovelli, Daniele Colosi, Luisa Doplicher, Winston Fairbairn, Leonardo Modesto, and Karim Noui. Background independence in a nutshell. *arXiv.org*, gr-qc, 2004.

[108] Winston J Fairbairn and Catherine Meusburger. q-Deformation of Lorentzian spin foam models. *PoS*, QGQGS2011:17, 2011.

[109] Hal M Haggard, Muxin Han, Wojciech Kamiński, and Aldo Riello. Four-dimensional Quantum Gravity with a Cosmological Constant from Three-dimensional Holomorphic Blocks. *Physics Letters*, B752:258–262, 2016.

[110] Roberto Pereira. Lorentzian LQG vertex amplitude. *Classical and Quantum Gravity*, 25:85013, 2008.

[111] Etera R Livine and Simone Speziale. A new spinfoam vertex for quantum gravity. *Physical Review*, D76:84028, 2007.

[112] Etera R Livine and Simone Speziale. Consistently Solving the Simplicity Constraints for Spinfoam Quantum Gravity. *EuroPhysics Letters*, 81:50004, 2008.

[113] Jonathan Engle, Roberto Pereira, and Carlo Rovelli. The loop-quantum-gravity vertex-amplitude. *Physical Review Letters*, 99:161301, 2007.

[114] You Ding, Muxin Han, and Carlo Rovelli. Generalized spinfoams. *Physical Review D*, 83(12), 2011.

[115] Carlo Rovelli. Zakopane lectures on loop gravity. *arXiv.org*, gr-qc, 2011.

[116] Pietro Doná and Simone Speziale. Introductory lectures to loop quantum gravity. In Abdelhafid Bounames and Abnenacer Makhlouf, editors, *TVC 79. Gravitation : théorie et expérience*. Hermann, 2013.

[117] Bianca Dittrich. The continuum limit of loop quantum gravity - a framework for solving the theory, 2016.

[118] Valentin Bonzom, Razvan Gurau, and Vincent Rivasseau. Random tensor models in the largenlimit: Uncoloring the colored tensor models. *Physical Review D*, 85(8), Apr 2012.

[119] Dario Benedetti and Razvan Gurau. Phase transition in dually weighted colored tensor models. *Nuclear Physics B*, 855(2):420–437, Feb 2012.

[120] Sylvain Carrozza, Daniele Oriti, and Vincent Rivasseau. Renormalization of a su(2) tensorial group field theory in three dimensions. *Communications in Mathematical Physics*, 330(2):581–637, Mar 2014.

[121] Joseph Ben Geloun, Tim A. Koslowski, Daniele Oriti, and Antonio D. Pereira. Functional renormalization group analysis of rank-3 tensorial group field theory: The full quartic invariant truncation. *Physical Review D*, 97(12), Jun 2018.

[122] Valentin Bonzom and Matteo Smerlak. Bubble divergences from twisted cohomology. *Communications in Mathematical Physics*, 312(2):399–426, apr 2012.

[123] Claudio Perini, Carlo Rovelli, and Simone Speziale. Self-energy and vertex radiative corrections in LQG. *Physics Letters*, B682:78–84, 2009.

[124] Thomas Krajewski, Jacques Magnen, Vincent Rivasseau, Adrian Tanasa, and Patrizia Vitale. Quantum Corrections in the Group Field Theory Formulation of the EPRL/FK Models. *Physical Review*, D82:124069, 2010.

[125] Wolfram Research, Inc. Mathematica, Version 12.2. Champaign, IL, 2020.

[126] Eugenio Bianchi and You Ding. Lorentzian spinfoam propagator. *Physical Review*, D86:104040, 2012.

[127] Eugenio Bianchi and Hal M. Haggard. Discreteness of the volume of space from bohr-sommerfeld quantization. *Physical Review Letters*, 107(1), 2011.

[128] Laurent Freidel and Simone Speziale. Twisted geometries: A geometric parametrisation of SU(2) phase space. *Physical Review*, D82:84040, 2010.

[129] H. T. Johansson and C. Forssén. Fast and accurate evaluation of Wigner 3j, 6j, and 9j symbols using prime factorisation and multi-word integer arithmetic. *SIAM Journal on Scientific Computing*, 38(1):A376–A384, April 2015.

[130] J. Rasch and A. C. H. Yu. Efficient storage scheme for precalculated wigner 3j, 6j and gaunt coefficients. *SIAM Journal on Scientific Computing*, 25(4):1416–1428, 2004.

[131] Pietropaolo Frisoni and Francesco Gozzini. Github repository *Self-energy*. `https://github.com/PietropaoloFrisoni/Self-energy-spin-foam-amplitude`, 2021.

[132] Seth K. Asante, Bianca Dittrich, and Hal M. Haggard. Effective Spin Foam Models for Four-Dimensional Quantum Gravity. *Physical Review Letters*, 125(23):231301, December 2020.

[133] Pietro Donà , Pietropaolo Frisoni, and Edward Wilson-Ewing. Radiative corrections to the lorentzian engle-pereira-rovelli-livine spin foam propagator. *Physical Review D*, 106(6), 2022.

[134] Aleksandar Miković and Marko Vojinović. A finiteness bound for the EPRL/FK spin foam model. *Classical and Quantum Gravity*, 30(3):035001, 2013.

[135] Pietropaolo Frisoni and Pietro Dona. Github repository *Radiative correction in spinfoams*. `https://github.com/PietropaoloFrisoni/Spinfoam_radiative_corrections`, 2022.

[136] L Freidel. Group Field Theory: An Overview. *International Journal of Theoretical Physics.*, 44:1769–1783, 2005.

[137] Daniele Oriti. Group field theory as the 2nd quantization of loop quantum gravity, 2015.

[138] Pietro Donà and Pietropaolo Frisoni. Summing bulk quantum numbers with monte carlo in spin foam theories. *Physical Review D*, 107(10), 2023.

[139] Seth K Asante, Bianca Dittrich, and Hal M Haggard. Discrete gravity dynamics from effective spin foams. *Classical and Quantum Gravity*, 38(14):145023, 2021.

[140] Pietro Donà , Marco Fanizza, Giorgio Sarno, and Simone Speziale. Numerical study of the lorentzian engle-pereira-rovelli-livine spin foam amplitude. *Physical Review D*, 100(10), 2019.

[141] Pietropaolo Frisoni. Numerical approach to the black-to-white hole transition. *Physical Review D*, 107(12), 2023.

[142] Pietropaolo Frisoni and Pietro Dona. Github repository *Monte Carlo spinfoams*. `https://github.com/PietropaoloFrisoni/Monte_Carlo_spinfoams`, 2022.

[143] Pietro Dona, Muxin Han, and Hongguang Liu. Spinfoams and high performance computing. 2022.

[144] Laurent Freidel and David Louapre. Diffeomorphisms and spin foam models. *Nuclear Physics*, B662:279–298, 2003.

[145] Eugenio Bianchi, Elena Magliaro, and Claudio Perini. Coherent spin-networks. *Physical Review D*, 82, 2010.

[146] Eugenio Bianchi, Elena Magliaro, and Claudio Perini. LQG propagator from the new spin foams. *Nuclear Physics*, B822:245–269, 2009.

[147] Benjamin Bahr and Sebastian Steinhaus. Hypercuboidal renormalization in spin foam quantum gravity. *Physical Review D*, 95:126006, 2017.

[148] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[149] Francesca Vidotto. Many-nodes/many-links spinfoam: the homogeneous and isotropic case. *Classical and Quantum Gravity.*, 28(245005), 2011.

[150] Francesca Vidotto. Relational quantum cosmology. In *The Philosophy of Cosmology*, pages 297–316. Cambridge University Press, 2017.

[151] J B Hartle and S W Hawking. Wave Function of the Universe. *Physical Review*, D28:2960–2975, 1983.

[152] CHRISTIAN RÖKEN. FIRST-ORDER QUANTUM-GRAVITATIONAL CORRECTION TO FRIEDMANNIAN COSMOLOGY FROM COVARIANT, HOLOMORPHIC SPINFOAM COSMOLOGY. *International Journal of Modern Physics D*, 22(02):1350005, 2013.

[153] Eugenio Bianchi, Thomas Krajewski, Carlo Rovelli, and Francesca Vidotto. Cosmological constant in spinfoam cosmology. *Physical Review*, D83:104015, 2011.

[154] Frank Hellmann. On the Expansions in Spin Foam Cosmology. *Physical Review D*, 84:103516, 2011.

[155] Marcin Kisielowski, Jerzy Lewandowski, and Jacek Puchta. Feynman diagrammatic approach to spinfoams. *Classical and Quantum Gravity*, 29(1):015009, 2011.

[156] Etera R Livine and Mercedes Martí n-Benito. Classical setting and effective dynamics for spinfoam cosmology. *Classical and Quantum Gravity*, 30(3):035006, 2013.

[157] Marcin Kisielowski, Jerzy Lewandowski, and Jacek Puchta. One vertex spin-foams with the dipole cosmology boundary. *Classical and Quantum Gravity*, 30(2):025007, 2012.

[158] Julian Rennert and David Sloan. Anisotropic spinfoam cosmology. *Classical and Quantum Gravity*, 31(1):015017, 2013.

[159] Steffen Gielen, Daniele Oriti, and Lorenzo Sindoni. Cosmology from Group Field Theory Formalism for Quantum Gravity. *Physical Review Letters*, 111(3):31301, 2013.

[160] Steffen Gielen, Daniele Oriti, and Lorenzo Sindoni. Homogeneous cosmologies as group field theory condensates. *JHEP*, 06:13, 2014.

[161] Steffen Gielen, , and Lorenzo Sindoni and. Quantum cosmology from group field theory condensates: a review. *Symmetry, Integrability and Geometry: Methods and Applications*, 2016.

[162] Ilya Vilensky. Spinfoam cosmology with the proper vertex amplitude. *Classical and Quantum Gravity*, 34(22):225015, 2017.

[163] Steffen Gielen and Daniele Oriti. Quantum cosmology from quantum gravity condensates: cosmological variables and lattice-refined dynamics. *New Journal of Physics*, 16(12):123004, 2014.

[164] Benjamin Bahr, Sebastian Klöser, and Giovanni Rabuffo. Towards a cosmological subsector of spin foam quantum gravity. *Physical Review D*, 96(8), 2017.

[165] Eugenio Bianchi, Leonardo Modesto, Carlo Rovelli, and Simone Speziale. Graviton propagator in loop quantum gravity. *Classical and Quantum Gravity*, 23:6989–7028, 2006.

[166] Etera R Livine and Simone Speziale. Group Integral Techniques for the Spinfoam Graviton Propagator. *JHEP*, 11:92, 2006.

[167] Emanuele Alesci, Eugenio Bianchi, and Carlo Rovelli. {LQG} propagator: {III}. {T}he new vertex. *Classical and Quantum Gravity*, 26:215001, 2009.

[168] Christian P. Robert. The metropolis-hastings algorithm, 2016.

[169] A P Yutsin, I B Levinson, and V V Vanagas. *Mathematical Apparatus of the Theory of Angular Momentum*. Israel Program for Scientific Translation, Jerusalem, Israel, 1962.

[170] J. Rasch and A. C. H. Yu. Efficient storage scheme for precalculated wigner 3j, 6j and gaunt coefficients. *SIAM Journal on Scientific Computing*, 25(4):1416–1428, 2004.

[171] Tim Besard, Christophe Foket, and Bjorn De Sutter. Effective extensible programming: Unleashing julia on gpus. *IEEE Transactions on Parallel and Distributed Systems*, 30(4):827–841, Apr 2019.

[172] Pietropaolo Frisoni Francesco Gozzini and Francesca Vidotto. Primordial fluctuations from quantum gravity: 16-cell model, 2023.

[173] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, Cambridge, 2010.

[174] Eugenio Bianchi, Pietro Donà, and Ilya Vilensky. Entanglement entropy of bell-network states in loop quantum gravity: Analytical and numerical results. *Physical Review D*, 99(8), Apr 2019.

[175] Norbert Bodendorfer, Muxin Han, Fabian Haneder, and Hongguang Liu. Path integral renormalization in loop quantum cosmology. *Physical Review D*, 103(12), 2021.

[176] Gregory Popovitch. Parallel hashmap. url=https://greg7mdp.github.io/parallel-hashmap/, 2019.

[177] Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 27381–27394. Curran Associates, Inc., 2021.

[178] Rohit Chandra, Leo Dagum, David Kohr, Ramesh Menon, Dror Maydan, and Jeff McDonald. *Parallel programming in OpenMP*. Morgan kaufmann, 2001.

[179] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[180] Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

[181] B. P. Abbott et. al. and (Ligo Collaboration Virgo Collaboration). Observation of Gravitational Waves from a Binary Black Hole Merger. *Physical Review Letters*, 116:061102, 2016.

[182] S. W. Hawking. Particle Creation by Black Holes. *Communications in Mathematical Physics*, 43:199–220, 1975. [Erratum: Commun.Math.Phys. 46, 206 (1976)].

[183] J V Narlikar, K.M.V. Appa Rao, and Naresh Dadhich. High energy radiation from white holes. *Nature*, 251:591, 1974.

[184] Valeri P. Frolov and G.A. Vilkovisky. Quantum Gravity removes Classical Singularities and Shortens the Life of Black Holes. *ICTP preprint IC/79/69, Trieste.*, 1979.

[185] V.P. Frolov and G.A. Vilkovisky. Spherically symmetric collapse in quantum gravity. *Physics Letters B*, 106:307–313, 1981.

[186] Steven B. Giddings and William M. Nelson. Quantum emission from two-dimensional black holes. *Physical Review D*, 46:2486–2496, 1992.

[187] C R Stephens, G 't Hooft, and B F Whiting. Black hole evaporation without information loss. *Classical and Quantum Gravity*, 11:621–647, 1994.

[188] Pawel O Mazur and Emil Mottola. Gravitational vacuum condensate stars. *Proceedings of the National Academy of Sciences of the United States of America*, 101(26):9545–50, 2004.

[189] Leonardo Modesto. Disappearance of the black hole singularity in loop quantum gravity. *Physical Review D*, 70(12):124009, 2004.

[190] Abhay Ashtekar and Martin Bojowald. Black hole evaporation: A paradigm. *Classical and Quantum Gravity*, 22:3349–3362, 2005.

[191] S.D. Mathur. The fuzzball proposal for black holes: an elementary review. *Fortschritte der Physik*, 53(7-8):793–827, 2005.

[192] Sean A. Hayward. Formation and Evaporation of Nonsingular Black Holes. *Physical Review Letters*, 96:031103, 2006.

[193] V. Balasubramanian, D. Marolf, and Rozali. M. Information Recovery From Black Holes. *General Relativity and Gravitation*, 38:1529–1536, 2006.

[194] Leonardo Modesto. Black Hole Interior from Loop Quantum Gravity. *Advances in High Energy Physics*, 2008:1–12, 2008.

[195] S. Hossenfelder and L. Smolin. Conservative solutions to the black hole information problem. *Physical Review D*, 81:064009, 2010.

[196] Valeri P. Frolov. Information loss problem and a 'black hole' model with a closed apparent horizon. *Journal of High Energy Physics*, 2014(5), 2014.

[197] Cosimo Bambi, Daniele Malafarina, and Leonardo Modesto. Non-singular quantum-inspired gravitational collapse. *Physical Review D*, 88(4):044009, 2013.

[198] Rodolfo Gambini and Jorge Pullin. Loop quantization of the Schwarzschild black hole. *Physical Review Letters*, 110:211301, 2013.

[199] James M. Bardeen. Black hole evaporation without an event horizon, 2014.

[200] Samir D. Mathur. A model with no firewall, 2015.

[201] Frank Saueressig, Natalia Alkofer, Giulio D'Odorico, and Francesca Vidotto. Black holes in Asymptotically Safe Gravity, 2015.

[202] Carlos Barceló, Raúl Carballo-Rubio, Luis J Garay, and Gil Jannes. The lifetime problem of evaporating black holes: mutiny or resignation. *Classical and Quantum Gravity*, 32:035012, 2015.

[203] Carlos Barceló , Raúl Carballo-Rubio, Luis J Garay, and Gil Jannes. The lifetime problem of evaporating black holes: mutiny or resignation. *Classical and Quantum Gravity*, 32(3):035012, 2015.

[204] Marios Christodoulou and Fabio D'Ambrosio. Characteristic time scales for the geometry transition of a black hole to a white hole from spinfoams, 2018.

[205] Farshid Soltani, Carlo Rovelli, and Pierre Martin-Dussaud. End of a black hole's evaporation. II. *Physical Review D*, 104(6), 2021.

[206] Marios Christodoulou, Fabio D'Ambrosio, and Charalampos Theofilis. Geometry transition in spinfoams. 2023.

[207] Farshid Soltani. A spin foam framework for the black-to-white hole transition. In *The Sixteenth Marcel Grossmann Meeting*. WORLD SCIENTIFIC, 2023.

[208] Carlo Rovelli. Introduction to Loop Quantum Gravity - lectures on YouTube, 2018.

[209] Fabio D'Ambrosio. Semi-classical holomorphic transition amplitudes in covariant loop quantum gravity, 2020.

[210] R. Penrose. Gravitational collapse: The role of general relativity. *Riv. Nuovo Cim.*, 1:252–276, 1969.

[211] Carlo Rovelli and Francesca Vidotto. Planck stars. *International Journal of Modern Physics. D*, 23:1442026, 2014.

[212] Sina Kazemian, Mateo Pascual, Carlo Rovelli, and Francesca Vidotto. Diffuse emission from black hole remnants. *Classical and Quantum Gravity*, 40(8):087001, 2023.

[213] Hal M. Haggard and Carlo Rovelli. Quantum-gravity effects outside the horizon spark black to white hole tunneling. *Physical Review D*, 92(10), 2015.

[214] Thomas Thiemann and Oliver Winkler. Gauge field theory coherent states (GCS). II: Peakedness properties. *Classical and Quantum Gravity*, 18:2561–2636, 2001.

[215] Eugenio Bianchi, Elena Magliaro, and Claudio Perini. Coherent spin-networks. *Physical Review*, D82:24012, 2010.

[216] Bianca Dittrich and Simone Speziale. Area-angle variables for general relativity. *New Journal of Physics*, 10:83006, 2008.

[217] Carlo Rovelli and Simone Speziale. On the geometry of loop quantum gravity on a graph. *Physical Review*, D82:44018, 2010.

[218] Varshaloviic Dimitrij Aleksandroviic, A. N. Moskalev, and Khersonskii Valerij Kel'manoviic. *Quantum theory of angular momentum: irreducible tensors, spherical harmonics, vector coupling coefficients, 3nj symbols*. World scientific, 1988.

[219] Pietropaolo Frisoni. Github repository *BlackToWhiteHole*. `https://github.com/PietropaoloFrisoni/BlackToWhiteHole`, 2023.

[220] Robert Oeckl. A "general boundary" formulation for quantum mechanics and quantum gravity. *Physics Letters B*, 575(3-4):318–324, 2003.

[221] Robert Oeckl. General boundary quantum field theory: Foundations and probability interpretation. *Advances in Theoretical and Mathematical Physics*, 12(2):319–352, 2008.

[222] Robert Oeckl. A predictive framework for quantum gravity and black hole to white hole transition. *Physics Letters A*, 382(37):2622–2625, 2018.

[223] M. Ambrus and P. Hájíček. Quantum superposition principle and gravitational collapse: Scattering times for spherical shells. *Physical Review D*, 72:064025, 2005.

[224] Carlos Barceló , Raúl Carballo-Rubio, and Luis J Garay. Exponential fading to white of black holes in quantum gravity. *Classical and Quantum Gravity*, 34(10):105007, apr 2017.

[225] Joseph Bunao Pietropaolo Frisoni and Jared Wogan. Generative Flow Networks in spinfoams, 2023.

[226] Gurtej Kanwar, Michael S. Albergo, Denis Boyda, Kyle Cranmer, Daniel C. Hackett, Sébastien Racanière, Danilo Jimenez Rezende, and Phiala E. Shanahan. Equivariant flow-based sampling for lattice gauge theory. *Physical Review Letters*, 125(12), 2020.

[227] M. S. Albergo, G. Kanwar, and P. E. Shanahan. Flow-based generative models for markov chain monte carlo in lattice field theory. *Physical Review D*, 100(3), 2019.

[228] Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets, 2022.

[229] Yoshua Bengio. The GFlowNet Tutorial, 2022.

[230] Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations, 2022.

[231] Ilkka Mäkinen. Introduction to SU(2) Recoupling Theory and Graphical Methods for Loop Quantum Gravity, 10 2019.

[232] Pietro Donà, Marco Fanizza, Giorgio Sarno, and Simone Speziale. Su(2) graph invariants, regge actions and polytopes. *Classical and Quantum Gravity*, 35, 2018.

[233] M Carmeli, E Leibowitz, and N Nissani. *Gravitation: SL(2,C) Gauge Theory and Conservation Laws*. WORLD SCIENTIFIC, 1990.

[234] Vong Duc Dao and Van Hieu Nguyen. On the theory of unitary representations of the sl(2,c) group. *Acta Physica Hungarica*, 22:201–219, 1967.

[235] M.A. Rashid. Boost matrix elements of the homogeneous lorentz group. *Journal of Mathematical Physics*, 20:1514–1519, 1979.

[236] D. Basu and S. Srinivasan. A unified treatment of the groups so(4) and so(3,1). *Czechoslovak Journal of Physics B*, 27:629–635, 1977.

[237] Hirosi Ooguri. Topological lattice models in four-dimensions. *Modern Physics Letters*, A7:2799–2810, 1992.

[238] Carlo Rovelli and Edward Wilson-Ewing. Discrete symmetries in covariant loop quantum gravity. *Physical Review D*, 86(6), 2012.

[239] Marios Christodoulou, Miklos Langvik, Aldo Riello, Christian Röken, and Carlo Rovelli. Divergences and Orientation in Spinfoams. *Classical and Quantum Gravity*, 30:055009, 2013.

[240] Razvan Gurau. Colored group field theory. *Communications in Mathematical Physics*, 304(1):69–93, 2011.

[241] Avram Sidi. *Practical Extrapolation Methods: Theory and Applications*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2003.

Appendix A

Appendix Chapter 1

A.1 $SU(2)$ toolbox

The group $SU(2)$ is the group of 2×2 complex matrices with unit determinants that satisfy the unitarity condition

$$\det(u) = 1\ , \quad \text{and} \quad u^{-1} = u^{\dagger}\ ,\ \forall u \in SU(2)\ . \tag{A.1}$$

The group is homomorphic to the rotation group $SO(3)$ and is generated by the angular momentum algebra L_i with $i = 1, 2, 3$ satisfying the commutation relations

$$\left[L_i, L_j\right] = i\epsilon_{ijk}L_k\ . \tag{A.2}$$

In the fundamental representation $L_i = \sigma_i/2$ where σ_i are the standard Pauli matrices. The Casimir operator is $L^2 = \vec{L}\cdot\vec{L}$ and the unitary irreducible representations are labeled by a spin $j \in \mathbb{N}/2$ a half-integer and are $2j+1$ dimensional. The canonical basis for these representations diagonalizes the operator L_3

$$L^2\,|j,m\rangle = j(j+1)\,|j,m\rangle\ , \qquad L_3\,|j,m\rangle = m\,|j,m\rangle\ . \tag{A.3}$$

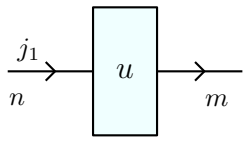
In this basis the matrix elements of the group are given by the Wigner matrices

$$D^j_{mn}(u) \equiv \langle j,m|\,u\,|j,n\rangle\ . \tag{A.4}$$

Their explicit expression and properties can be found in [218] and we will not report them.

In this work, we compute integrals of products of $SU(2)$ representation matrices in terms of $SU(2)$ invariants. We will introduce the minimal amount of tools needed and the graphical method to perform the calculations. We do not want to provide a complete introduction to recoupling theory and its graphical method that are worth books and reviews on their own [231, 84, 169]. We use a graphical notation that is completely analogous to the one introduced in Section 1.4.

We associate an oriented line to each $SU(2)$ representation matrix. We decorate the line with a spin label and a box containing the group element

$D^j_{mn}(u)$ = 

. (A.5)

We contract two representations summing over the magnetic indices by connecting the two lines. We compute the integral over $SU(2)$ using the unique invariant measure over the group (the Haar measure du [84]). The explicit form of the measure depends on the parametrization used for the group. We collect the boxes corresponding to the same group elements. In the following, we will always imply the integration of all the group elements in the boxes.

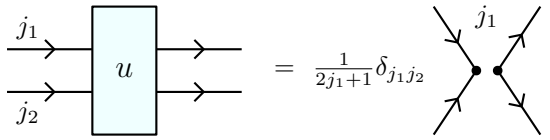
The integral of the product of two representation matrices is given by

$$\int \mathrm{d}u D^{j_1}_{m_1 n_1}(u) D^{j_2}_{m_2 n_2}(u) = \frac{1}{2j_1+1}\delta_{j_1 j_2}(-1)^{2j_1-m_1-n_1}\delta_{-m_1 m_2}\delta_{-n_1 n_2} = \frac{1}{2j_1+1}\delta_{j_1 j_2}\epsilon^{j_1}_{m_1 m_2}\epsilon^{j_1}_{n_1 n_2}\,, \tag{A.6}$$

where we defined the tensor $\epsilon^{j_1}_{m_1 m_2} \equiv (-1)^{j_1-m_1}\delta_{-m_1 m_2}$, the unique invariant tensor in the product of two j_1 representations. The ϵ tensor squares to

$$\sum_{m_2} \epsilon^{j_1}_{m_1 m_2}\epsilon^{j_1}_{m_2 m_3} = \sum_{m_2}(-1)^{j_1-m_1}\delta_{-m_1 m_2}(-1)^{j_1-m_2}\delta_{-m_2 m_3} = (-1)^{2j_1-m_1+m_3}\delta_{m_1 m_3} = (-1)^{2j_1}\delta_{m_1 m_3}\,, \tag{A.7}$$

and has the symmetry property $\epsilon^{j_1}_{m_1 m_2} = (-1)^{2j_1}\epsilon^{j_1}_{m_2 m_1}$. We use the graphical representation to write (A.6) as



. (A.8)

The invariance property of the tensor $\epsilon^{j_1}_{m_1 m_2}$ means

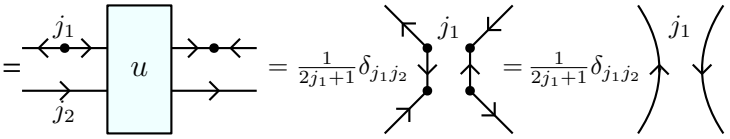
$$\sum_{n_1,n_2} D^{j_1}_{m_1 n_1}(u) D^{j_1}_{m_2 n_2}(u)\epsilon^{j_1}_{n_1 n_2} = \epsilon^{j_1}_{m_1 m_2}\,. \tag{A.9}$$

From (A.9) we can derive the property of Wigner matrices

$$\sum_{m_2 n_2} \epsilon^{j_1}_{n_1 m_2}\epsilon^{j_1}_{m_1 n_1} D^{j_1}_{m_2 n_2}(u) = D^{j_1}_{m_1 n_1}(u^{-1})\,. \tag{A.10}$$

Using this property we can also perform integrals where an inverse group element appears

$$\int \mathrm{d}u D^{j_1}_{n_1 m_1}(u^{-1}) D^{j_2}_{m_2 n_2}(u) =$$

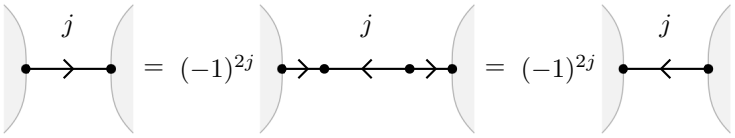


(A.11)

$$= \frac{1}{2j_1+1}\delta_{j_1 j_2}\delta_{m_1 m_2}\delta_{n_1 n_2}\,. \tag{A.12}$$

For simplicity, we will merge the ϵ tensors with the box in the following. At first sight, it could appear as an ambiguity since one line will have a group element u in the box, while the line with the opposite orientation u^{-1} and there is no indication of which is which. However, we are integrating over u. Therefore, the name we give the group element is irrelevant. The important information is contained in the relative polarity: one group element is the inverse of the other.

Using this convention and the square property (A.7), in any closed diagram, inverting the orientation of a line (without group elements) results into a phase $(-1)^{2j}$.



(A.13)

The integral of the product of three representation matrices is given by

$$\int \mathrm{d}u D^{j_1}_{m_1 n_1}(u) D^{j_2}_{m_2 n_2}(u) D^{j_3}_{m_3 n_3}(u) = \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \begin{pmatrix} j_1 & j_2 & j_3 \\ n_1 & n_2 & n_3 \end{pmatrix} . \tag{A.14}$$

The tensors appearing in (A.14) are the Wigner $(3jm)$ symbols, the unique invariant tensor (or three valent *intertwiner*) in the tensor product of three $SU(2)$ representations.

$$\sum_{n_1,n_2,n_2} D^{j_1}_{m_1 n_1}(u) D^{j_2}_{m_2 n_2}(u) D^{j_3}_{m_3 n_3}(u) \begin{pmatrix} j_1 & j_2 & j_3 \\ n_1 & n_2 & n_3 \end{pmatrix} = \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} . \tag{A.15}$$

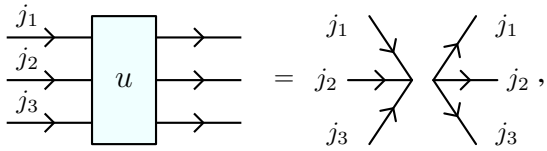
The $(3jm)$ has the following symmetry properties (see [84, 169, 218] for an exhaustive list)

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ n_1 & n_2 & n_3 \end{pmatrix} = \begin{pmatrix} j_2 & j_3 & j_1 \\ n_2 & n_3 & n_1 \end{pmatrix} = (-1)^{j_1+j_2+j_3} \begin{pmatrix} j_1 & j_3 & j_2 \\ n_1 & n_3 & n_2 \end{pmatrix} , \tag{A.16}$$

and vanishes unless the selection rules are satisfied

$$m_1 + m_2 + m_3 = 0 , \qquad |j_1 - j_2| \leq j_3 \leq j_1 + j_2 , \qquad j_1 + j_2 + j_3 \in \mathbb{N} . \tag{A.17}$$

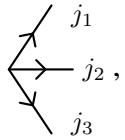
In the graphical representation (A.14) is



(A.18)

where for the $(3jm)$ symbol

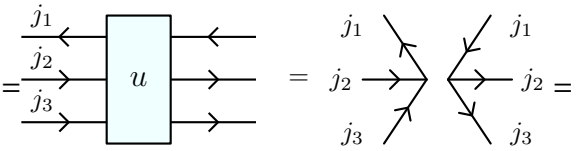
$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} =$$



(A.19)

we read the spins in clockwise order if all the arrows are outgoing and in anti-clockwise order if all the arrows are ingoing. In the standard $SU(2)$ graphical calculus, this is usually indicated with a sign next to the node [169, 218, 231]. For our calculations, this is unnecessary, and we avoid adding this extra layer of complexity. Similarly to (A.11) we have

$$\int \mathrm{d}u D^{j_1}_{n_1 m_1}(u^{-1}) D^{j_2}_{m_2 n_2}(u) D^{j_3}_{m_3 n_3}(u) =$$

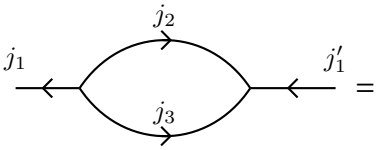


(A.20)

$$(-1)^{j_1-n_1} \begin{pmatrix} j_1 & j_2 & j_3 \\ -n_1 & n_2 & n_3 \end{pmatrix} (-1)^{j_1-m_1} \begin{pmatrix} j_1 & j_2 & j_3 \\ -m_1 & m_2 & m_3 \end{pmatrix} .$$

The $(3jm)$ satisfy the orthogonality relation

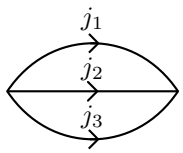
$$\sum_{m_2,m_3} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \begin{pmatrix} j_1' & j_2 & j_3 \\ m_1' & m_2 & m_3 \end{pmatrix} =$$



$$= \frac{1}{2j_1+1} \delta_{j_1 j_1'} \delta_{m_1 m_1'} \,, \tag{A.21}$$

and are normalized to 1

$$\sum_{m_1,m_2,m_3} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} =$$



$$= 1 \,. \tag{A.22}$$

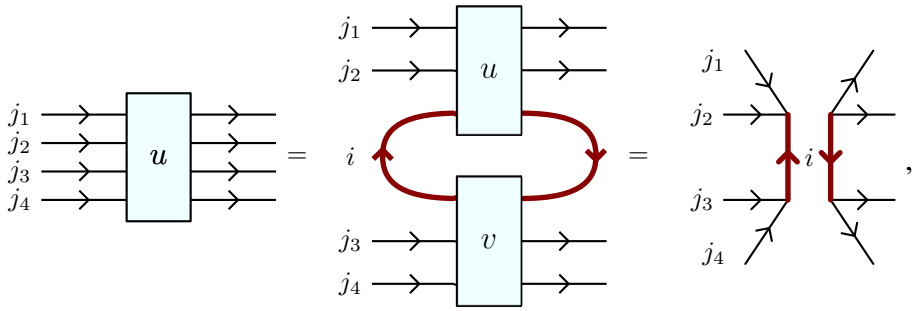
We also compute the integral of the product of four representation matrices with the tools we provided using a trick

$$\int \mathrm{d}u D^{j_1}_{m_1 n_1}(u) D^{j_2}_{m_2 n_2}(u) D^{j_3}_{m_3 n_3}(u) D^{j_4}_{m_4 n_4}(u) = \int \mathrm{d}u \mathrm{d}v D^{j_1}_{m_1 n_1}(u) D^{j_2}_{m_2 n_2}(u) \delta(uv^{-1}) D^{j_3}_{m_3 n_3}(v) D^{j_4}_{m_4 n_4}(v) \,, \tag{A.23}$$

where we doubled the number of integrals by inserting a delta function. The delta function can be expanded as a sum of Wigner functions [1]

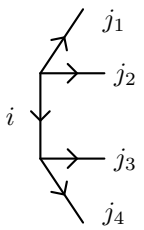
$$\delta(uv^{-1}) = \sum_i (2j+1) \sum_m D^i_{mm}(uv^{-1}) = \sum_i (2i+1) \sum_{m,n} D^i_{mn}(u) D^i_{nm}(v^{-1}) \,. \tag{A.24}$$

Using the graphical representation and the properties (A.15) and (A.20) we compute (A.23)



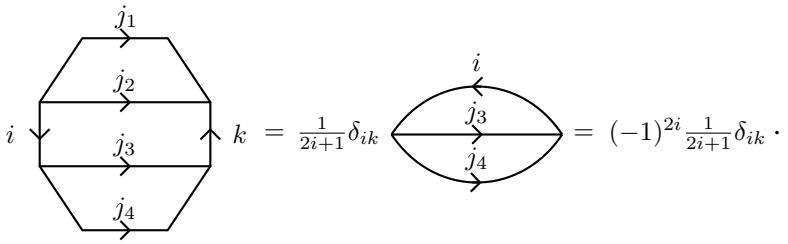
(A.25)

where we used a red thick line to imply a summation weighted by the dimensional factor $(2i+1)$ and we define the $(4jm)$ symbols (invariant tensor or four valent intertwiner) as



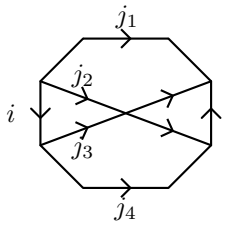
$$= \begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{(i)} = \sum_m (-1)^{i-m} \begin{pmatrix} j_1 & j_2 & i \\ m_1 & m_2 & m \end{pmatrix} \begin{pmatrix} i & j_3 & j_4 \\ -m & m_3 & m_4 \end{pmatrix} \,. \tag{A.26}$$

The way we grouped representations together in (A.23) is completely arbitrary. The definition (A.26) corresponds to the choice of coupling (also called recoupling basis) of the representation of spins j_1 and j_2 (ore equivalently spins j_3 and j_4). The orthogonality condition (A.21) of the $(3jm)$ symbols imply the normalization of the $(4jm)$ symbols



(A.27)

The contraction of two $(4jm)$ symbols in different recoupling basis forms another notable $SU(2)$ invariant called the $\{6j\}$ symbol.



$$= (-1)^{j_2+j_3+i+k} \begin{Bmatrix} j_1 & j_2 & i \\ j_4 & j_3 & k \end{Bmatrix} . \tag{A.28}$$

The $\{6j\}$ symbol in terms of $(3jm)$ symbols can be written in a canonical from as

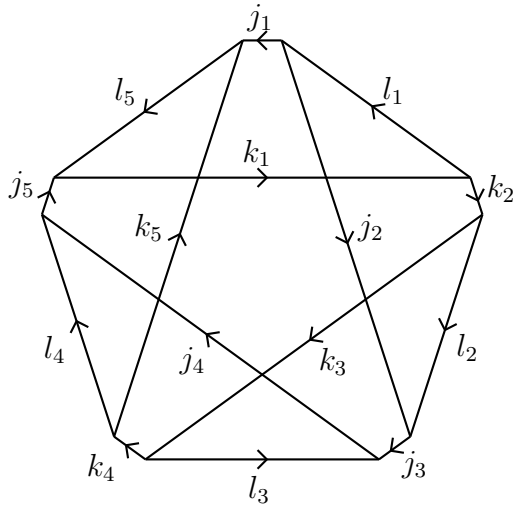
$$\begin{Bmatrix} j_1 & j_2 & j_3 \\ j_4 & j_5 & j_6 \end{Bmatrix} = \sum_{m_1 \dots m_6} (-1)^{\sum_{i=1}^{6}(j_i - m_i)} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & -m_3 \end{pmatrix} \begin{pmatrix} j_1 & j_5 & j_6 \\ -m_1 & m_5 & m_6 \end{pmatrix} \times$$
$$\times \begin{pmatrix} j_4 & j_5 & j_3 \\ m_4 & -m_5 & m_3 \end{pmatrix} \begin{pmatrix} j_4 & j_2 & j_6 \\ -m_4 & -m_2 & -m_6 \end{pmatrix} . \tag{A.29}$$

For a numerical evaluation, it is not convenient to write the $\{6j\}$ symbol as in (A.29). It is much more efficient to rely on libraries that compute and store Wigner $\{6j\}$ symbols optimally using recursion and symmetry properties, such as `wigxjpf` and `fastwixj` [91, 170].

Another higher-order invariant that appears in our calculations is the irreducible $\{15j\}$ symbol of the first kind (following the classification of [169]). We can write it both graphically and

in terms of $\{6j\}$ symbols as:

$$\left\{ \begin{array}{ccccc} j_1 & j_2 & j_3 & j_4 & j_5 \\ l_1 & l_2 & l_3 & l_4 & l_5 \\ k_1 & k_2 & k_3 & k_4 & k_5 \end{array} \right\} =$$



$$= (-1)^{\sum_{i=1}^{5} j_i + l_i + k_i} \sum_x (2x+1) \left\{ \begin{array}{ccc} j_1 & k_1 & x \\ k_2 & j_2 & l_1 \end{array} \right\} \left\{ \begin{array}{ccc} j_2 & k_2 & x \\ k_3 & j_3 & l_2 \end{array} \right\} \times \left\{ \begin{array}{ccc} j_3 & k_3 & x \\ k_4 & j_4 & l_3 \end{array} \right\} \left\{ \begin{array}{ccc} j_4 & k_4 & x \\ k_5 & j_5 & l_4 \end{array} \right\} \left\{ \begin{array}{ccc} j_5 & k_5 & x \\ j_1 & k_1 & l_5 \end{array} \right\} . \quad \text{(A.30)}$$

It must be emphasized that the $\{15j\}$ symbol (A.30) is not the most convenient choice from a numerical point of view. In fact, it is possible to choose the recoupling scheme in order to obtain reducible $\{15j\}$ symbols (see [232] for an example), whose evaluation is much faster. However, since this aspect is not the most critical part of the performance, we prefer to have a pleasantly symmetrical symbol and sacrifice some efficiency. This also simplifies computations of spinfoams transition amplitudes with many vertices, since the basis choice in the recoupling on one edge affects both vertices it connects. Therefore, choosing a symmetric $\{15j\}$ symbol as in (A.30), we are sure that the recoupling is consistent in every vertex.

A.2 $SL(2,\mathbb{C})$ toolbox

The group $SL(2,\mathbb{C})$ is the group of 2×2 complex matrices with unit determinant. The group is homomorphic to the proper Lorentz group (the Lorentz group part that preserve the sign of the time component) [233, 43].

The algebra of $SL(2,\mathbb{C})$ is generated by spatial rotations and boosts L_i and K_i satisfying the commutation relations

$$\Big[L_i, L_j\Big] = i\epsilon_{ijk} L_k \,, \qquad \Big[L_i, K_j\Big] = i\epsilon_{ijk} K_k \,, \qquad \Big[K_i, K_j\Big] = -i\epsilon_{ijk} L_k \,. \quad \text{(A.31)}$$

In the spinorial representation $L_i = \sigma_i/2$ and $K_i = i\sigma_i/2$ where σ_i are the standard Pauli matrices. The two Casimir operators are $K^2 - L^2$ and $\vec{K} \cdot \vec{L}$. The unitary irreducible representations in the principal series are labeled by ρ a real number and k a half-integer. In these representations the Casimirs assume the values

$$\Big(K^2 - L^2\Big) |\rho, k\rangle = (\rho^2 - k^2 + 1) |\rho, k\rangle \,, \qquad \vec{K} \cdot \vec{L} |\rho, k\rangle = \rho k |\rho, k\rangle \,. \quad \text{(A.32)}$$

The generic unitary representation (ρ, k) is infinite dimensional since the group is non-compact. However, we can decompose the representation (ρ, k) in an infinite number of $SU(2)$ representations that diagonalize L^2 with different values of the spin j

$$(\rho, k) = \bigoplus_{j \geq k} j \ . \tag{A.33}$$

The definition of the EPRL model is based on the canonical basis of (ρ, k). In this basis we diagonalize L^2 and L_3

$$L^2 \, |\rho, k; j, m\rangle = j(j+1) \, |\rho, k; j, m\rangle \ , \qquad L_3 \, |\rho, k; j, m\rangle = m \, |\rho, k; j, m\rangle \ . \tag{A.34}$$

with $j \geq k$ and $m = -j, \ldots, j$.

The Cartan parametrization [43, 42] of the group $SL(2,\mathbb{C})$ is given by the map

$$g = u e^{\frac{r}{2}\sigma_3} v^{-1} \ , \tag{A.35}$$

where $u, v \in SU(2)$, $r \in [0, \infty)$ is the rapidity and σ_3 is the diagonal Pauli matrix the generator of boosts along the z axis. The Haar measure with respect to this parametrization is [43, 42]

$$\mathrm{d}g = \frac{1}{4\pi} \sinh^2 r \, \mathrm{d}r \, \mathrm{d}u \, \mathrm{d}v \ . \tag{A.36}$$

Using the Cartan parametrization (A.35) the matrix elements of a group element g in the canonical basis reads

$$D^{\rho,k}_{jmln}(g) \equiv \langle \rho, k; j, m| \, g \, |\rho, k; l, n\rangle = D^{\rho,k}_{jmln}(u e^{\frac{r}{2}\sigma_3} v^{-1}) = \sum_{a,a'} D^j_{ma}(u) D^{\rho,k}_{jaa'}(e^{\frac{r}{2}\sigma_3}) D^l_{a'n}(v^{-1}) \ . \tag{A.37}$$

The subgroup $SU(2) \subset SL(2,\mathbb{C})$ is generated by $\vec{L}$ and its matrix elements are given by $SU(2)$ Wigner matrices (A.4)

$$D^{\rho,k}_{jmln}(u) = \langle \rho, k; j, m| \, u \, |\rho, k; l, n\rangle = \delta_{jl} D^j_{mn}(u) \ \text{ where } \ u \in SU(2) \ . \tag{A.38}$$

Moreover, $e^{\frac{r}{2}\sigma_3}$ is diagonal, therefore $D^{\rho,k}_{jmln}(e^{r\sigma_3}) = \delta_{aa'} D^{\rho,k}_{jala}(e^{r\sigma_3}) \equiv \delta_{aa'} d^{\rho,k}_{jla}(r)$ where $d^{\rho,k}_{jla}$ are called reduced matrix elements of $SL(2,\mathbb{C})$. Summarizing

$$D^{\rho,k}_{jmln}(g) = \sum_a D^j_{ma}(u) \, d^{\rho,k}_{jla}(r) \, D^l_{an}(v^{-1}) \ . \tag{A.39}$$

The expression for $d^{\rho,k}_{jlm}(r)$ was given in [234, 235, 236, 43, 42]

$$\begin{aligned} d^{\rho,k}_{jlm}(r) = (-1)^{j-l} & \sqrt{\frac{(i\rho - j - 1)! \, (j + i\rho)!}{(i\rho - l - 1)! \, (l + i\rho)!}} \frac{\sqrt{(2j+1)(2l+1)}}{(j+l+1)!} e^{(i\rho - k - m - 1)r} \\ & \sqrt{(j+k)!(j-k)!(j+m)!(j-m)!(l+k)!(l-k)!(l+m)!(l-m)!} \\ & \sum_{s,t} (-1)^{s+t} e^{-2tr} \frac{(k+s+m+t)!(j+l-k-m-s-t)!}{t!s!(j-k-s)!(j-m-s)!(k+m+s)!(l-k-t)!(l-m-t)!(k+m+t)!} \\ & {}_2F_1 \left[\{l - i\rho + 1, k + m + s + t + 1\}, \{j + l + 2\}; 1 - e^{-2r} \right] \end{aligned} \tag{A.40}$$

where ${}_2F_1$ is the Gauss hypergeometric function. The phase used in (A.40) is the same introduced in [42], which ensures the reality of the booster function (1.23). The reduced matrix elements (A.40) satisfy the following relation:

$$\overline{d^{\rho,k}_{jlm}(r)} = (-1)^{j-l} d^{\rho,k}_{jl-m}(r) \,. \tag{A.41}$$

As a consequence, the matrices (A.38) have the property:

$$\overline{D^{\rho,k}_{jmln}(g)} = (-1)^{j-l+m-n} D^{\rho,k}_{j-ml-n}(g) \,. \tag{A.42}$$

We can write the $SL(2,\mathbb{C})$ matrix elements of g^{-1} as:

$$D^{\rho,k}_{ln\,jm}(g^{-1}) = (-1)^{j-l+m-n} D^{\rho,k}_{j-ml-n}(g) = \sum_a (-1)^{j-l+m-n}\, D^j_{-ma}(u)\, d^{\rho,k}_{jla}(r)\, D^l_{a-n}(v^{-1}) \,, \tag{A.43}$$

where in the first equality we used (A.42) (in addition to the $SL(2,\mathbb{C})$ irrep properties) and in the second one (A.39). Since there are no phases depending on the summed index, we conclude that the orientation of the $(4\,jm)$ spins in the booster function (1.23) is irrelevant. This justifies the fact that we draw the latter without arrows.

A.3 Approximation of a convergent series

In this appendix, we provide further details on the extrapolation scheme used in Section 1.7. This is analogous to the more general Aitken's delta-squared process [105], which accelerates the rate of convergence of a sequence providing a good approximation technique . Consider the series $S = \sum_n^\infty a_n$ and cut-offed sum $S_N = \sum_n^N a_n$. By definition, the series is the limit of S_N for an infinite cut-off

$$S = \lim_{N\to\infty} S_N = \lim_{N\to\infty} \sum_n^N a_n = \sum_n^\infty a_n \,. \tag{A.44}$$

Suppose that the sequence a_n is positive and, from a certain point onwards, increasing such that

$$\lim_{N\to\infty} c_N \equiv \lim_{N\to\infty} \frac{a_N}{a_{N-1}} = \lim_{N\to\infty} \frac{S_N - S_{N-1}}{S_{N-1} - S_{N-2}} \equiv L < 1 \,, \tag{A.45}$$

where the ratios increase to L. The series S is convergent by the ratio test since the ratios are increasing:

$$c_N = \frac{a_N}{a_{N-1}} < \frac{a_k}{a_{k-1}} \,, \qquad \forall k > N \,. \tag{A.46}$$

Hence we have $a_{N+1} = a_N \frac{a_{N+1}}{a_N} > a_N c_N$, $a_{N+2} > a_{N+1} c_N > a_N c_N^2$, and in general $a_{N+m} > a_N c_N^m$ for $m > 0$. We can provide a bound on the series observing that

$$S - S_N = \sum_{n=N+1}^\infty a_n = \sum_{m=1}^\infty a_{N+m} > \sum_{m=1}^\infty a_N c_N^m = a_N \frac{c_N}{1-c_N} \,. \tag{A.47}$$

Similarly, we have that $\frac{a_k}{a_{k-1}} < L \;\forall k > N$ by definition of L and monotonicity of the ratios.

$$S - S_N = \sum_{n=N+1}^\infty a_n = \sum_{m=1}^\infty a_{N+m} < \sum_{m=1}^\infty a_N L^m = a_N \frac{L}{1-L} \,. \tag{A.48}$$

Summarizing, we have an estimate from above and below of the value of the series as

$$S_N + a_N \frac{c_N}{1 - c_N} < S < S_N + a_N \frac{L}{1 - L} \,. \tag{A.49}$$

If the ratios are decreasing instead of increasing, we obtain an estimate analog to (A.49) but with inequalities operators inverted. Let us focus on (A.49) since it is the case relevant for the cut-off approximation presented in Section 1.7. We rewrite (A.49) in terms of cut-offed sums as

$$\frac{S_N S_{N-2} - S_{N-1}^2}{S_N - 2S_{N-1} + S_{N-2}} < S < \frac{S_N - S_{N-1} L}{1 - L} \,. \tag{A.50}$$

Often, in real-world physical applications, the analytical expression of a_n is very complicated. We cannot compute S but can still calculate the cut-offed sums S_N with N as large as our numerical computational resources allow. What is the best approximation of S we can find? We will assume that we know S is convergent (so that the question is well-posed) and that the ratios c_N increase. We want to use the inequalities (A.50). We can numerically compute the left-hand side of the inequality. What about the right-hand side? The convergence of S ensures that $\lim_{N\to\infty} c_N = L$ exists, however in general we cannot compute the value of L. At the moment, there is no clear strategy on how to compute L. In this work, we will consider two possibilities. None of them is optimal, and we leave improvements to future work. Notice that no matter what approximation we decide to adopt to compute L the right inequality of (A.50) will not hold anymore.

For example, we can approximate L with the largest available ratio $L \approx c_N$. In particular, if we insist and substitute in (A.50) the approximation $L \approx c_N$, the right quantity becomes equal to the left one. We have to content ourselves with a lower-bound estimate of the amplitude given

$$S \gtrapprox \frac{S_N S_{N-2} - S_{N-1}^2}{S_N - 2S_{N-1} + S_{N-2}} \,. \tag{A.51}$$

The estimate (A.51) is similar to the strategy used in [72, 71]. We clarified that it is a lower bound. Another possibility is to use the sequence of ratios c_N computed numerically to estimate the value of L. This extrapolation is slightly dangerous since its accuracy depends on how large we can take the cut-off N. Of course, this is in addition to the lower bound (A.51).

In the following, we provide a concrete toy model example. Consider the series

$$S = \sum_{n=1}^{\infty} \frac{1}{n+1} (9/10)^n = \frac{10}{9} \log(10) - 1 \approx 1.558 \,. \tag{A.52}$$

This series is exactly summable in terms of the log function. Nevertheless, we want to approximate the series pretending not to know how to sum it, ignoring that any analytical calculation is straightforward, and relying only on numerical tools. Let us assume that the largest possible cut-off we can access is $N = 15$. We can compute the cut-offed sums

$$S_{15} \approx 1.480 \,, \qquad S_{14} \approx 1.467 \,, \qquad S_{13} \approx 1.452 \,. \tag{A.53}$$

We can immediately apply (A.51) to obtain

$$S \gtrapprox \frac{S_{15}S_{13} - S_{14}^2}{S_{15} - 2S_{14} + S_{13}} \approx 1.549 \,. \tag{A.54}$$

The largest cut-offed sum is 5% off the actual value while the lower bound approximation (A.54) is closer, being only 0.6% off. The numerical values for the ratios are

N	3	4	5	6	7	8	9	10	11	12	13	14	15
c_N	0.675	0.720	0.750	0.771	0.788	0.800	0.810	0.818	0.825	0.831	0.836	0.840	0844

We extrapolate the limit at infinity of the ratios L fitting the data using the first few terms of an inverse power law and keeping the constant term. It is a cheap and dirty way of extrapolating, and one should be more careful. However, it is more than enough for our purposes. We use Wolfram's Mathematica built-in `Fit` method to perform the fit and find $L \approx 0.889$. If we substitute it in (A.50), keeping in mind the approximations we are making, we find the estimate

$$S \lessapprox \frac{S_{15} - S_{14}L}{1 - L} \approx 1.583 \,, \tag{A.55}$$

which is 1.6% larger than the actual value. Combining the two estimates, we obtain a range for the series $S \in [1.549, 1.583]$.

Appendix B

Appendix Chapter 2

B.1 Boundary observables

A useful check of the consistency of the geometrical picture and the numerical calculation can be obtained by looking at the geometry of the two boundary tetrahedra. Their geometrical properties are jointly fixed by choosing the boundary data and by the dynamics. Here we show the resulting expectation values of some geometrical boundary operators, that is:

$$\langle O \rangle = \frac{\langle W|O|\Psi\rangle}{\langle W|\Psi\rangle} \tag{B.1}$$

where the bra W contains the propagator, namely the dynamics, while the ket Ψ turns out to be the tensor product of the *in* and *out* states of the LQG Hilbert space. With the term "propagator," we refer to the square matrices (they are such since the self-energy triangulation has two boundary tetrahedra) in which the element a, b corresponds to the amplitude (2.13) with $a = i_+, b = i_-$. Computing the expectation values (B.1) of geometric operators describing boundary tetrahedra is a significant numerical investigation, especially considering that computations carried out with the EPRL model are still in their primordial stages [70, 45, 77, 57, 32, 72], and, as far as we know, at present time there are no such numerical computations with spinfoams containing a bubble.

Since in the calculation of (B.1), the dimensional factor of the boundary intertwiners is relevant, in this section, it is convenient to define a "normalized" amplitude as:

$$W_N(j, i_\pm, K) \equiv W(j, i_\pm, K)\,\sqrt{d_{\{i_\pm\}}}\,, \tag{B.2}$$

where $d_{\{i_\pm\}} = \prod_{\pm}(2i_\pm + 1) = (2i_+ + 1)(2i_- + 1)$.

Angles

The angle operator measures the shape of the tetrahedra in twisted geometry:

$$A_{ab}|j, i_\pm\rangle = \cos(\theta_{ab})_{i_\pm}|j, i_\pm\rangle \tag{B.3}$$

which is interpreted as the cosine of the external dihedral angle between the faces a and b of the tetrahedron defined on the node associated with the intertwiner $i_\pm$. The spin-network basis

states diagonalize the dihedral angle θ_{ab} between faces a and b. The equation for the dihedral angle $\cos(\theta_{ab})$ in terms of intertwiner spin $i_\pm$ reads:

$$\cos(\theta_{ab})_{i_\pm} = \frac{i_\pm(i_\pm + 1) - j_a(j_a + 1) - j_b(j_b + 1)}{2\sqrt{j_a(j_a + 1)j_b(j_b + 1)}}. \tag{B.4}$$

We consider the expectation value (B.1) of the angle operator (B.3) in any of the two (equal) boundary regular tetrahedra of the triangulation using the spin network state. According to the recoupling basis (j_1, j_2), we focus on the angle between faces 1 and 2. The expectation value on node + can be computed as:

$$\frac{\langle W|A_{12}|W\rangle}{\langle W|W\rangle} = \frac{\sum\limits_{i_\pm} \left[W_N(j, i_\pm, K)\right]^2 \cos(\theta_{12})_{i_+}}{\sum\limits_{i_\pm} \left[W_N(j, i_\pm, K)\right]^2}, \tag{B.5}$$

where we considered the case in which all boundary spins are equal to j. Carrying out the numerical computation of (B.5), we obtain a value that is in agreement with the geometric value of the external angle of a regular tetrahedron up to the tenth significant digit, as shown in Figure B.1. We repeated the calculation for increasing values of boundary spins j up to $j = 5$, also varying Δl, finding exactly the same value and the unaltered precision. This is consistent

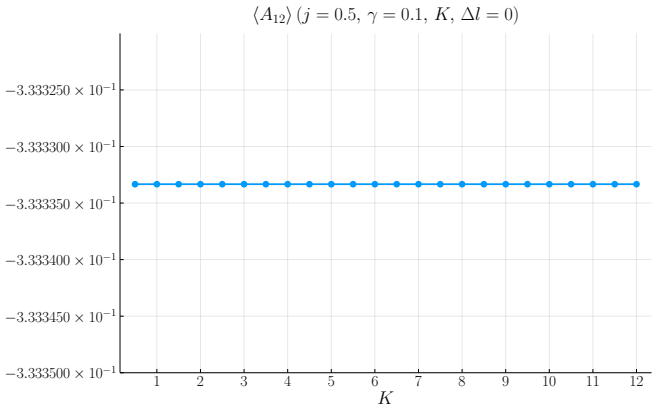


Figure B.1: *The dynamic expectation value of the cosine of the external dihedral angle operator* (B.5). *We find an excellent agreement with the value* $-0.\bar{3}$.

with the fact that we are looking at the only angle which is completely sharp, while the others turn out to be spread.

Volumes

The volume operator of a tetrahedron reads:

$$V_{abc} = \frac{\sqrt{2}}{3}(8\pi G\hbar\gamma)^{\frac{3}{2}} \sqrt{\left|\vec{J}_a \cdot \vec{J}_b \times \vec{J}_c\right|}, \tag{B.6}$$

where $\vec{J}_a$ is the vector of $SU(2)$ generators on link a. The operator (B.6) is not diagonal in the spin network states. The dynamic expectation value (see [1] for the derivation) on node + turns out to be:

$$\frac{\langle W|V|W\rangle}{\langle W|W\rangle} = \frac{\sum\limits_{i_\pm, i'_+} \sqrt{d_{\{i_+\}} d_{\{i'_+\}}} W_N(j, i_\pm, K) W_N(j, i'_+, K) v\,(i_+, i'_+, j)}{\sum\limits_{i_\pm} \left[W_N(j, i_\pm, K)\right]^2} \tag{B.7}$$

where $v\,(i_+, i'_+, j)$ is the matrix element:

$$v\,(i_+, i'_+, j) = \frac{\sqrt{2}}{3} (8\pi G\hbar\gamma)^{\frac{3}{2}} \sum_v \langle j, i'_+|v\rangle \sqrt{|q_v|} \langle v|j, i_+\rangle, \tag{B.8}$$

in which q_v are the eigenvalues of the volume, while $|v\rangle$ corresponds to the eigenvector relative to the eigenvalue q_v. In Figure B.2 we report the numerical values obtained from (B.7). We see that the scaling of the expectation value of the boundary volume as a function of the spin j corresponds to that existing between the volume of a regular tetrahedron and the area of one of its faces, that is $V \propto j^{3/2}$. As for the angle operator, we repeated the calculation by varying the number of shells, finding the same trend.

Finally, we repeated the calculation with coherent boundary states verifying that the classical contribution dominates the expectation value for the angle operator and volume.

The picture emerges that the boundary observables have an expected value that coincides with the classical one and is not affected by divergence in the bulk of the spinfoam. Namely, even if the amplitude (2.13) is divergent, the expectation values (B.5), (B.7) are finite and fully consistent with the geometry of a regular tetrahedron.

B.2 Internal contributions

We add here a tentative analysis of the kind of configurations that contribute to the divergence. For this aim, it is convenient to rewrite equation (2.13) as

$$W\,(j, i, \gamma; K, \Delta l) = \sum_{j_{ab} \leq K} \omega(j_{ab}; j, i, \gamma, \Delta l). \tag{B.9}$$

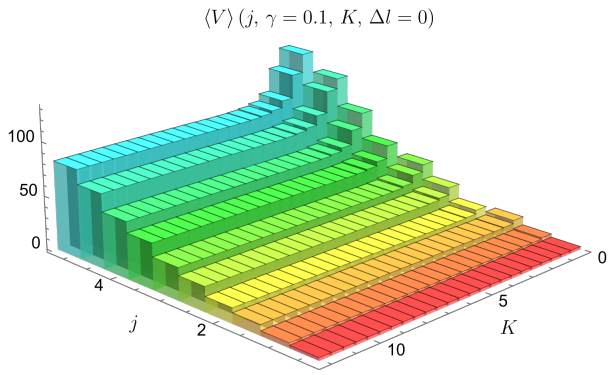


Figure B.2: *The dynamic expectation value of the volume operator* (B.7). *We neglected all the constant factors in* (B.7) *and* (B.8). *The volumes scale as* $j^{\frac{3}{2}}$.

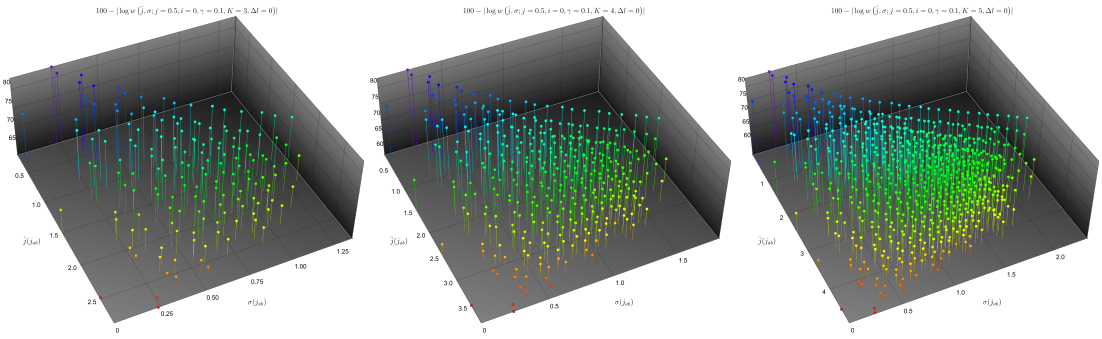


Figure B.3: *The largest individual contributions to the divergent amplitude* (2.13) *come from "semi-classical" bulk spins configurations with* $\bar{j} \sim j$ *and low* σ*. Triangular inequalities result in a triangular shape in the plane* $\left(\bar{j}, \sigma\right)$.

Now, consider the average and the standard deviation of the sextuplet of internal spins $\{j_{ab}\}$:

$$\bar{j}(j_{ab}) = \frac{1}{6}\sum_{a\neq b} j_{ab}, \quad \sigma(j_{ab}) = \sqrt{\frac{1}{6}\sum_{a\neq b}(j_{ab} - \bar{j}_{ab})^2} \tag{B.10}$$

and define

$$w(\bar{j}, \sigma; j, i, \gamma, K, \Delta l) = \sum_{\substack{\bar{j}(j_{ab})=\bar{j} \\ \sigma(j_{ab})=\sigma}} \omega(j_{ab}; j, i, \gamma, \Delta l). \tag{B.11}$$

That is, we write the contributions to the divergence as a function of the average and standard deviation of the internal spins, adding the contributions with different sextuplets with the same average and standard deviation. The function $100 - \left|\log w(\bar{j}, \sigma; j, i, \gamma, K, \Delta l\right|)$ is plotted for fixed $j, i, \gamma, \Delta l$ and various values of K in Figure B.3, where the factor 100 is only for visual purposes. Increasing Δl increases the value of each point while rising K also implies adding more configurations, as shown in Figure 2.4. Figure B.3 indicates that the largest individual contributions to the amplitude (2.13) come from "semiclassical" bulk spins configurations with $\bar{j} \sim j$ and low σ, as one might expect. At the same time, the divergence comes from the proliferation of smaller contributions with arbitrarily high spin and deviations.

We also note that the divergence of the self-energy, manifestly shown in Figure 2.9, occurs in the limit $\Delta l \to \infty$ only. In fact, in [46], it is shown that the sum over the internal spins of the amplitude (2.13) is convergent in the simplified model $\Delta l = 0$. Even if an analytical proof is not yet available, it seems likely that the amplitude has a finite limit $K \to \infty$ for each finite Δl since the internal spin configurations in the limit become more and more suppressed. Since also the limit $\Delta l \to \infty$ at fixed K is finite, we can schematize the role of the two limits with the following four-node diagram:

$$\begin{array}{ccc} W(K, \Delta l) & \xrightarrow{K\to\infty} & \text{convergence?} \\ {\scriptstyle \Delta l\to\infty}\downarrow & & \downarrow{\scriptstyle \Delta l\to\infty} \\ \text{convergence} & \xrightarrow{K\to\infty} & \text{divergence.} \end{array}$$

Appendix C

Appendix Chapter 3

C.1 Topological BF SU(2) spinfoam theory

Calculations with the topological BF SU(2) spinfoam theory are straightforward analytically and numerically. Therefore they provide a convenient test for the techniques we use and the choice of parameters we make with the EPRL theory.

The analytical calculation reduces to integrating group-valued delta functions, and the numerical calculations can be comfortably done on a standard laptop using optimized libraries to compute Wigner symbols [91, 170].

This theory's boundary data, face, and edge amplitudes are the same as the EPRL spinfoam theory (3.3). The vertex amplitude is simpler and consists of a $SU(2)$ invariant $A_v(j_f, i_e) = \{15j\}$ (A.30). The theory is affected by large spin divergences, and the renormalization is well studied [144, 54] and is related to residual diffeomorphism gauge symmetry. We can rewrite the amplitude (3.3) in a more suitable way for analytical calculations. The amplitude can be cast in terms of integrals (with the invariant measure) over a copy of $SU(2)$ for each edge of $SU(2)$ irreducible representations of spin j_f for each face. Boundary edges are contracted with 4-valent intertwiners. Bulk (closed) faces the $SU(2)$ character $\operatorname{Tr} D^{(j_f)}$ of the product of all the group elements of the edges composing the face g_f. The sum over the spin associated with the face j_f can be performed exactly

$$\sum_{j_f} A_f(j_f) \operatorname{Tr} D^{(j_f)}(g_f) = \sum_{j_f} (2j_f + 1) \operatorname{Tr} D^{(j_f)}(g_f) = \delta(g_f) \,, \tag{C.1}$$

in terms of the delta function over the group. The group elements are usually interpreted as the $SU(2)$ holonomy providing the parallel transport between reference frames associated with the spinfoam edges. As expected from a topological theory, the delta function on spinfoam faces forces holonomy around each bulk face to be trivial. In this form, the amplitude calculation reduces to integrating out delta functions. Divergences of the amplitudes manifest as delta functions evaluated at the identity. We regularize them with a cutoff of K.

$$\delta(\mathbb{1}) \approx \sum_{j_f=0}^{K} (2j_f + 1) \operatorname{Tr} D^{(j_f)}(\mathbb{1}) = \sum_{j_f=0}^{K} (2j_f + 1)^2 = \frac{8}{3}k^3 + 6k^2 + \frac{13}{3}k + 1 = O(K^3) \,. \tag{C.2}$$

We compute the degree of divergence analytically of four diagrams in this paper 6F(a), 4F(a), 4F(c), and 4F(e).

The BF amplitude of the melonic self-energy diagram 6F(a) is

$$A^{(BF)}_{6F(A)}(j_b, i_b;\, K) = \sum_{m_1 m_2 m_3 m_4} \begin{pmatrix} j_b & j_b & j_b & j_b \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{i_b} \begin{pmatrix} j_b & j_b & j_b & j_b \\ n_1 & n_2 & n_3 & n_4 \end{pmatrix}^{i_b} \int \prod_{e=1}^{6} dg_e D^{j_1}_{m_1 n_1}(g_1 g_2 g_6)\, D^{j_2}_{m_2 n_2}(g_1 g_3 g_6)\, D^{j_3}_{m_3 n_3}(g_1 g_4 g_6)\, D^{j_4}_{m_4 n_4}(g_1 g_5 g_6) \\ \delta\left(g_2 g_3^{-1}\right) \delta\left(g_2 g_4^{-1}\right) \delta\left(g_2 g_5^{-1}\right) \delta\left(g_3 g_4^{-1}\right) \delta\left(g_3 g_5^{-1}\right) \delta\left(g_4 g_5^{-1}\right) . \tag{C.3}$$

Performing all the integrals one by one and using the invariance of the intertwiners, we get

$$A^{(BF)}_{6F(A)}(j_b, i_b;\, K) = \frac{1}{2i_b + 1} \delta(\mathbb{1})^3 = O(K^9) . \tag{C.4}$$

The degree of divergence of the melonic self-energy diagram is 9.

Similarly, we can look at the BF amplitude of the diagram 4F(a),

$$A^{(BF)}_{4F(A)}(j_b, i_b;\, K) = \sum_{m_1 m_2 m_3 m_4} \begin{pmatrix} j_b & j_b & j_b & j_b \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{i_b} \begin{pmatrix} j_b & j_b & j_b & j_b \\ n_1 & n_2 & n_3 & n_4 \end{pmatrix}^{i_b} \int \prod_{e=1}^{6} dg_e D^{j_b}_{m_1 n_1}(g_1 g_2 g_3 g_4 g_6)\, D^{j_b}_{m_2 n_2}(g_1 g_2 g_3 g_4 g_6)\, D^{j_b}_{m_3 n_3}(g_1 g_2 g_3 g_4 g_6)\, D^{j_b}_{m_4 n_4}(g_1 g_2 g_3 g_4 g_6) \\ \delta(g_2)\, \delta(g_4)\, \delta\left(g_3 g_5^{-1}\right) \delta\left(g_2 g_5 g_4 g_5^{-1}\right) . \tag{C.5}$$

The result of the integrals is

$$A^{(BF)}_{4F(A)}(j_b, i_b;\, K) = \frac{1}{2i_b + 1} \delta(\mathbb{1}) = O(K^3) . \tag{C.6}$$

The degree of divergence of the diagram 4F(a) is 3. Notice that the divergence of this diagram is subdominant to the melonic self-energy one.

In total analogy, the BF amplitude of the diagram 4F(c) is

$$A^{(BF)}_{4F(C)}(j_b, i_b;\, K) = \sum_{m_1 m_2 m_3 m_4} \begin{pmatrix} j_b & j_b & j_b & j_b \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{i_b} \begin{pmatrix} j_b & j_b & j_b & j_b \\ n_1 & n_2 & n_3 & n_4 \end{pmatrix}^{i_b} \int \prod_{e=1}^{6} dg_e D^{j_b}_{m_1 n_1}(g_1 g_6)\, D^{j_b}_{m_2 n_2}\left(g_1 g_3 g_5 g_3^{-1} g_6\right) D^{j_b}_{m_2 n_2}\left(g_1 g_2 g_5 g_4^{-1} g_6\right) D^{j_b}_{m_2 n_2}\left(g_1 g_2 g_5^{-1} g_4^{-1} g_6\right) \\ \delta\left(g_2 g_3^{-1}\right) \delta\left(g_2 g_4^{-1}\right) \delta\left(g_3 g_4^{-1}\right) \delta(g_5) . \tag{C.7}$$

The result of the integrals is

$$A^{(BF)}_{4F(C)}(j_b, i_b;\, K) = \frac{1}{2i_b + 1} \delta(\mathbb{1}) = O(K^3) , \tag{C.8}$$

so the degree of divergence of the diagram 4F(c) is 3.

Finally the BF amplitude of the diagram 4F(e) is

$$A^{(BF)}_{4F(D)}(j_b, i_b;\, K) = \sum_{m_1 m_2 m_3 m_4} \begin{pmatrix} j_b & j_b & j_b & j_b \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{i_b} \begin{pmatrix} j_b & j_b & j_b & j_b \\ n_1 & n_2 & n_3 & n_4 \end{pmatrix}^{i_b} \int \prod_{e=1}^{6} dg_e \int D^{j_b}_{m_1 n_1}(g_1 g_6)\, D^{j_b}_{m_2 n_2}(g_1 g_2 g_6)\, D^{j_b}_{m_3 n_3}\left(g_1 g_2^{-1} g_6\right) D^{j_b}_{m_4 n_4}\left(g_1 g_3 g_4 g_5 g_3^{-1} g_6\right) \delta(g_2)\,\delta(g_4)\,\delta(g_5)\,\delta\left(g_2 g_3 g_4 g_5 g_4^{-1} g_5^{-1} g_3^{-1}\right) . \tag{C.9}$$

The result of the integrals is

$$A^{(BF)}_{4F(D)}(j_b, i_b;\, K) = \frac{1}{2i_b + 1}\delta(\mathbb{1}) = O(K^3) . \tag{C.10}$$

Once again, a degree of divergence of 3 for the diagram 4F(e).

We compare the analytic calculation of the degree of divergence of the diagrams 4F(a), 4F(c), and 4F(e) with the numerical estimation done using the same technique we employ for the EPRL model. The computation is straightforward and can be done in seconds using the *BF* vertex of `sl2cfoam-next`. We fix the homogeneous cutoff to $K \lessapprox 10$ as a prototype for the EPRL calculation. In all three cases, we can fit the amplitudes with a cubic polynomial (see Figure C.1), finding a perfect agreement with the analytic calculations. Furthermore, it suggests that using a cutoff of order 10 is not a limiting factor in our analysis.

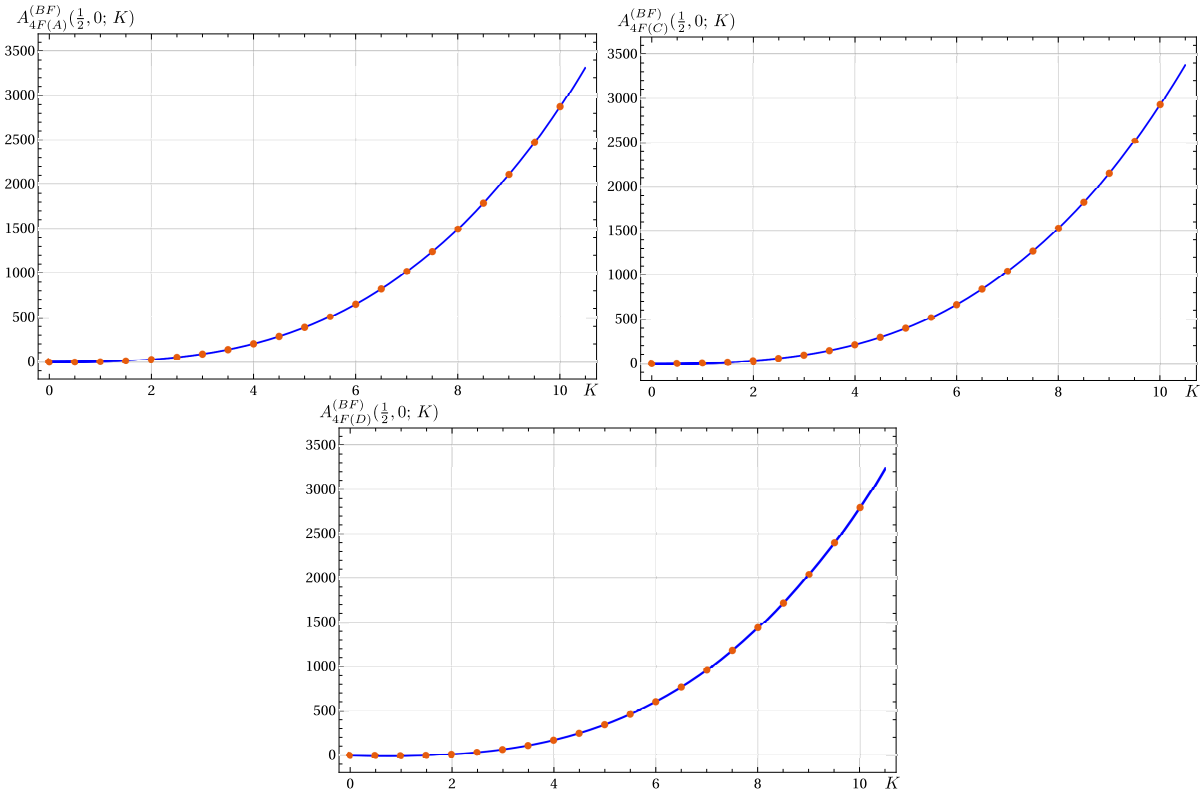


Figure C.1: *Plot of the numerical values of the BF amplitude of the spinfoam diagram 4F(a), 4F(c), and 4F(e) (left to right) with boundary spins* $j_b = \frac{1}{2}$ *(orange dots) and the corresponding fits (blue band). The leading order of the fits are* $A_{4F(A)}(\frac{1}{2},0;K) \approx (2.657, 2.667)k^3$, $A_{4F(C)}(\frac{1}{2},0;K) \approx (2.666, 2.667)k^3$, $A_{4F(D)}(\frac{1}{2},0;K) \approx (2.666, 2.667)k^3$. *For all cases, it is compatible with the analytical value of* $8/3k^3$ *of* (C.2).

C.2 Other Diagrams with Two Vertices

Here we present some diagrams with two vertices and two boundary edges, but that is ruled out by (at least) one of the other two conditions that (i) there are exactly four boundary faces, and each boundary face contains both boundary edges, and (ii) the propagators are trivial and preserve the order of the faces. The examples we give here are not exhaustive (there are many more diagrams besides the ones we discuss here) and are only meant to survey other possible contributions to the self-energy calculation. Whether these diagrams (and others) should be included or not depends on the symmetries of the spinfoam model. We note that we imposed these conditions as a simplifying assumption to reduce the number of diagrams we consider as a first step. To calculate the full self-energy of the propagator (to two vertices), it may be necessary to include the diagrams that do not satisfy these conditions.

These two conditions are quite different. The first condition tracks the boundary faces through the spinfoam and requires that each boundary face traverse the diagram and exit on the outgoing edge. Another way to understand this condition is that it requires all boundary faces to connect the spinfoam's initial and final boundary edges. This condition is not guaranteed to be satisfied because the boundary face could turn around and exit on the ingoing edge. A

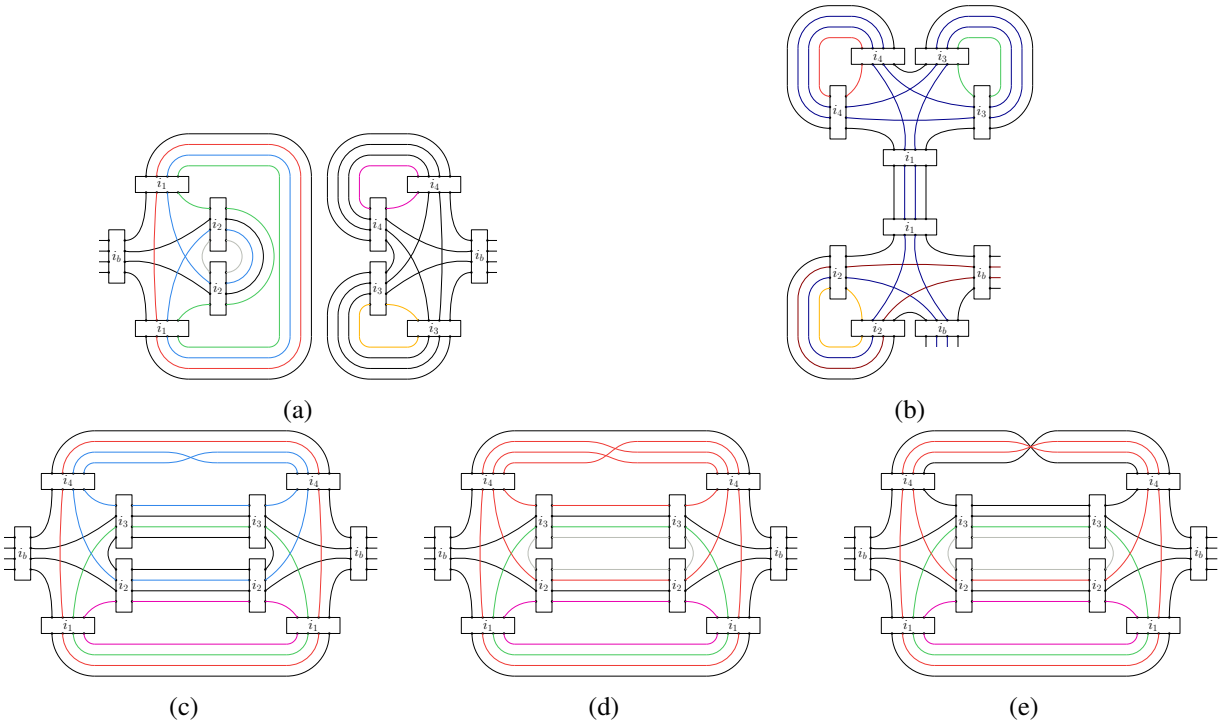


Figure C.2: *Some examples of spinfoam diagrams we did not consider.*

diagram with this property (a face that enters and leaves the diagram on the same edge) is only possible if two faces of the edge have the same spin, for example, $j_1 = j_2$. For this reason, this type of diagram can only contribute to the self-energy calculation for specific edge propagators, namely those with faces with at least one repeated spin. On the other hand, for an edge whose four faces all have different spins, these diagrams cannot contribute. Although it may seem unusual to have diagrams that contribute to some choices of boundary spins but not others (given the same boundary graph), if all possible paths are to be summed over, there appear to be more paths possible in the presence of symmetries in the spin labels of the propagator.

In Fig. C.2, the diagrams (A) and (B) are two examples of spinfoams that violate condition (i). These two diagrams fall into two subcategories: diagram (A) is disconnected, while diagram (B) is connected. In diagram (A), the spinfoam is completely disconnected, and it could contribute to the self-energy only for propagators where the four faces can be paired up, in this case, $j_1 = j_4$ and $j_2 = j_3$. In contrast, diagram (B) is connected, and for this spinfoam, only two of the faces of the propagator need to be paired up. In this case, $j_2 = j_3$ (we highlight them in dark red and blue in the picture).

As an aside, note that (either) half of diagram (A) may be relevant for no-boundary calculations in cosmology, where one wishes to calculate the transition from the 'no-geometry vacuum' state (with no quanta of geometry) to an excited state corresponding to a cosmological space-time.

The second condition that the order of the faces in the propagator is preserved ensures that there are no 'twists' in the internal propagators. We imposed this condition for simplicity because it drastically reduces the number of diagrams to consider. Taking the geometric perspective that the propagator represents a quantum tetrahedron, a cyclic permutation of three

faces can be seen as a rotation. In comparison, the interchange of two faces is a reflection (or a parity transformation). Since there is no preferred background, it may seem natural that rotations should be a symmetry of the theory [237]. However, arguments exist that parity transformations should not be allowed [238, 239]. Note that any number of interchanges in the order of the faces corresponds to either a rotation or a parity transformation: an even number of interchanges of faces gives a rotation. In contrast, an odd number is a parity transformation (perhaps composed of a rotation).

Examples of these two possibilities are given in Fig. C.2, where diagram (C) contains one interchange of faces corresponding to a parity transformation. At the same time, there is a cyclic permutation of three faces in diagram (D) corresponding to a rotation. Note that 'twisted' diagrams often have fewer internal faces than similar spinfoams without twists in the internal propagators. As a technical aside, note that the edge's intertwiner refers to a specific recoupling choice, so tracking how the recoupling scheme transforms under a rotation or parity transformation is essential.

Also, note that a propagator that completely reverses the order of the faces from (j_1, j_2, j_3, j_4) to (j_4, j_3, j_2, j_1) can be obtained through rotations and corresponds to a change in whether the propagator is being viewed from 'above' or from 'below'; an example is given in diagram (E) in Fig. C.2. It may seem appropriate that this transformation be allowed for propagators in spinfoams, even if different rotations and reflections are not.

The spinfoam diagrams that must be included in the calculation depend on the symmetries of the spinfoam model. For example, the spinfoam model derived from colored group field theory has very rigid combinatorics (thereby ensuring a one-to-one correspondence between any given spinfoam diagram and the dual simplicial complex) that does not allow any permutations in the faces of the propagator [240]. As a result, the number of spinfoams that need to be computed (with a given number of vertices) for any given boundary state will be much smaller for colored group field theories, thereby significantly simplifying the calculation. On the other hand, a model that allows permutations of the faces in the propagators will have many more diagrams contribute when calculating the self-energy and presumably also when calculating different amplitudes.

To summarize, in this paper, we imposed some conditions on the self-energy diagrams we considered to reduce the number of spinfoams we need to calculate, as otherwise, a large number of diagrams need to be computed. We stress that these conditions are used only to simplify the task for a first exploration of self-energy diagrams beyond the melonic diagram shown in Figure 6F(a) (that has already been studied in some detail [102, 71, 72, 46]). Although determining the right conditions to be imposed on the self-energy diagrams will ultimately depend on the symmetries of the theory (these symmetries could be defined directly at the level of the spinfoam model or instead in a group field theory from which one can derive the spinfoam expansion), it is possible that the conditions imposed here may be too strict, in which case it would be necessary to relax these conditions, depending on the symmetries of the spinfoam model.

Appendix D

Appendix Chapter 4

D.1 Vertex Amplitudes

In this Appendix, we report the definition of the Topological BF SU(2) spinfoam vertex amplitude.

$$A_v^{BF}(j_1, j_2, j_3, j_4, j_5, j_6, j_7, j_8, j_9, j_{10}; i_1, i_2, i_3, i_4, i_5) = \left\{ \begin{array}{ccccc} i_1 & j_3 & i_4 & j_6 & i_2 \\ j_4 & j_{10} & j_8 & j_5 & j_1 \\ j_7 & i_5 & j_9 & i_3 & j_2 \end{array} \right\} . \tag{D.1}$$

The SU(2) invariant in (D.1) is a $\{15j\}$ symbol of the first kind. According to the conventions of [169], we write it in terms of Wigner's $\{6j\}$ symbols.

$$\begin{aligned} \left\{ \begin{array}{ccccc} j_1 & j_2 & j_3 & j_4 & j_5 \\ l_1 & l_2 & l_3 & l_4 & l_5 \\ k_1 & k_2 & k_3 & k_4 & k_5 \end{array} \right\} = (-1)^{\sum_{i=1}^5 j_i + l_i + k_i} \sum_x (2x+1) \left\{ \begin{array}{ccc} j_1 & k_1 & x \\ k_2 & j_2 & l_1 \end{array} \right\} \left\{ \begin{array}{ccc} j_2 & k_2 & x \\ k_3 & j_3 & l_2 \end{array} \right\} \\ \times \left\{ \begin{array}{ccc} j_3 & k_3 & x \\ k_4 & j_4 & l_3 \end{array} \right\} \left\{ \begin{array}{ccc} j_4 & k_4 & x \\ k_5 & j_5 & l_4 \end{array} \right\} \left\{ \begin{array}{ccc} j_5 & k_5 & x \\ j_1 & k_1 & l_5 \end{array} \right\} . \end{aligned} \tag{D.2}$$

In this work we also use the in-line notation for

$$\{6j\}(j_1, j_2, j_3, j_4, j_5, j_6) = \left\{ \begin{array}{ccc} j_1 & j_2 & j_3 \\ j_4 & j_5 & j_6 \end{array} \right\} . \tag{D.3}$$

In the booster function decomposition, the Lorentzian EPRL vertex amplitude is defined as:

$$\begin{aligned} &A_v(j_1, j_2, j_3, j_4, j_5, j_6, j_7, j_8, j_9, j_{10}; i_1, i_2, i_3, i_4, i_5) = \\ &\sum_{l_f = j_f}^{\infty} \sum_{k_e} \left\{ \begin{array}{ccccc} i_1 & j_3 & k_4 & l_6 & k_2 \\ j_4 & l_{10} & l_8 & l_5 & j_1 \\ l_7 & k_5 & l_9 & k_3 & j_2 \end{array} \right\} (2k_2+1)(2k_3+1)(2k_4+1)(2k_5+1) \\ &\quad B_4^\gamma(j_5, j_6, j_7, j_1, l_5, l_6, l_7, j_1; i_2, k_2)\, B_4^\gamma(j_8, j_9, j_2, j_5, l_8, l_9, j_2, l_5; i_3, k_3) \\ &\quad B_4^\gamma(j_{10}, j_3, j_6, j_8, l_{10}, j_3, l_6, l_8; i_4, k_4)\, B_4^\gamma(j_4, j_7, j_9, j_{10}, j_4, l_7, l_9, l_{10}; i_5, k_5) \ . \end{aligned} \tag{D.4}$$

The booster functions are one-dimensional integrals over the rapidity parameter r of the reduced matrix elements, in the γ-simple unitary representation of $SL(2,\mathbb{C})$.

$$B_4^\gamma(j_1,j_2,j_3,j_4,l_1,l_2,l_3,l_4;i,k) = \sum_{p_f} \begin{pmatrix} l_1 & l_2 & l_3 & l_4 \\ p_1 & p_2 & p_3 & p_4 \end{pmatrix}^{(k)} \left(\int_0^\infty \mathrm{d}r \frac{1}{4\pi} \sinh^2 r \bigotimes_{f=1}^4 d^{\gamma j_f, j_f}_{l_f j_f p_f}(r) \right) \begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ p_1 & p_2 & p_3 & p_4 \end{pmatrix}^{(i)} . \tag{D.5}$$

The expression for $d^{\gamma j,j}_{jlm}(r)$ has been written in [43, 42]:

$$\begin{aligned} d^{(\gamma j,j)}_{jlp}(r) =& (-1)^{\frac{j-l}{2}} \frac{\Gamma(j+i\gamma j+1)}{|\Gamma(j+i\gamma j+1)|} \frac{\Gamma(l-i\gamma j+1)}{|\Gamma(l-i\gamma j+1)|} \frac{\sqrt{2j+1}\sqrt{2l+1}}{(j+l+1)!} \\ & \times \left[(2j)!(l+j)!(l-j)! \frac{(l+p)!(l-p)!}{(j+p)!(j-p)!} \right]^{1/2} e^{-(j-i\gamma j+p+1)r} \\ & \times \sum_s \frac{(-1)^s e^{-2sr}}{s!(l-j-s)!} {}_2F_1[l+1-i\gamma j, j+p+1+s, j+l+2, 1-e^{-2r}] . \end{aligned} \tag{D.6}$$

where ${}_2F_1$ is the Gauss hypergeometric function. In (D.5) we used a short notation for the $(4jm)$ Wigner symbols. They are the contraction of two Wigner $(3jm)$ symbols (the unique intertwiner of three $SU(2)$ representation) labeled by the virtual spin k:

$$\begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ p_1 & p_2 & p_3 & p_4 \end{pmatrix}^{(k)} = \sum_{p_k} (-1)^{k-p_k} \begin{pmatrix} j_1 & j_2 & k \\ p_1 & p_2 & p_k \end{pmatrix} \begin{pmatrix} k & j_3 & j_4 \\ -p_k & p_3 & p_4 \end{pmatrix} . \tag{D.7}$$

The large quantum number limit of the booster functions (D.5) has been studied in [88]. They possess an appealing geometrical interpretation in terms of boosted tetrahedra.

D.2 Monte Carlo summation

Suppose we want to compute the sum

$$S = \sum_{j \in \mathcal{J}} a_j , \tag{D.8}$$

where $\mathcal{J}$ is a finite subset of $(\mathbb{N}/2)^F$ and j is a multi-index. So far, we are just muddling the waters with a complicated notation to give a nod to the spinfoam application. If $V_\mathcal{J} = |\mathcal{J}|$ is the cardinally of the set, we can always map it in the interval of natural numbers between 1 and $V_\mathcal{J}$. The sum (D.8) is a fancy way to represent a sum over an integer index from 1 to $V_\mathcal{J}$.

If $V_\mathcal{J}$ is very large, the numerical computation of S can result in a highly resource-hungry task. Therefore, we want to use (discrete) Monte Carlo techniques to approximate its value.

We define a uniform probability density function over the set $\mathcal{J}$ as $R_\mathcal{J}$. We assume that $R_\mathcal{J}$ is normalized to 1. The probability associated to every element of $\mathcal{J}$ with $R_\mathcal{J}$ is $1/V_\mathcal{J}$. The hypothesis that the set $\mathcal{J}$ is finite ensures that $R_\mathcal{J}$ exists.

The fundamental step towards the implementation of Monte Carlo is to interpret the sum (D.8) as the expectation value of the terms of the sum a_j using $R_{\mathcal{J}}$

$$S = V_{\mathcal{J}} \sum_{j\in\mathcal{J}} \frac{a_j}{V_{\mathcal{J}}} = V_{\mathcal{J}}\, E\left[a(R_{\mathcal{J}})\right] \ . \tag{D.9}$$

We use a discrete uniform probability distribution because we assume we do not know in advance which term of the sum a_j contributes the most. This information would allow sampling from a more efficient probability distribution using importance sampling Monte Carlo. Alternatively, it would be possible to use Markov Chain Monte Carlo methods, such as the Metropolis-Hastings algorithm. This approach has been applied recently in [78] to compute observables in spinfoams with many boundary degrees of freedom.

We approximate the expectation value in (D.9) using a sample of the set $\mathcal{J}$. We use the probability distribution $R_{\mathcal{J}}$ and randomly extract N_{mc} elements from the set $\mathcal{J}$. We denote this set as $\mathcal{J}^{mc}$. Strictly speaking, $\mathcal{J}_{mc}$ is not a subset of $\mathcal{J}$ since it can contain elements more than once. The Monte Carlo estimate of S is given by

$$S^{mc} = \frac{V_{\mathcal{J}}}{N_{mc}} \sum_{j\in\mathcal{J}^{mc}} a_j \ . \tag{D.10}$$

The law of large numbers ensures that the average of a large number of samples becomes closer and closer to the expected value as more samples are performed. Since S^{mc} is a sample of the sum S we have that

$$S^{mc} \xrightarrow{N_{mc}\to\infty} S \ . \tag{D.11}$$

The sum S^{mc} (D.10) is the Monte Carlo estimate of S. We consider a "large" number of samples N_{mc} as we are interested in a numerical approximation of (D.9).

The amount of computational resources necessary to compute S^{mc} scales with the number of samples N_{mc} and not the size of the original set $V_{\mathcal{J}}$. The number of samples N_{mc} is a calculation parameter that we can tune. We must find a balanced value for N_{mc} to have a good approximation and save resources.

D.3 Aitken extrapolation

The Aitken's delta-squared process or Aitken extrapolation [105] is a numerical recipe used to accelerate a sequence's convergence rate. Suppose you have a convergent sequence S_n with $S = \lim_{n\to\infty} S_n$. If S_n converges linearly, namely

$$\lim_{n\to\infty} \frac{|S_n - S|}{|S_{n-1} - S|} = \lambda \ , \tag{D.12}$$

with $0 < \lambda < 1$. Linear convergence means the sequence is closer to its limit by almost the same amount with every step. In this case, we can approximate S starting from the approximate relation

$$\frac{S_n - S}{S_{n-1} - S} \approx \frac{S_{n-1} - S}{S_{n-2} - S} \ , \tag{D.13}$$

and solve for S to find

$$S \approx A[S_n] \equiv \frac{S_n S_{n-2} - S_{n-1}^2}{S_n - 2S_{n-1} + S_{n-2}} \,. \tag{D.14}$$

The sequence $A[S_n]$ is the Aitken extrapolation of S_n and converge to S faster than linear (see the original paper [105] or the book [241] for a proof) meaning that

$$\lim_{n\to\infty} \frac{|A[S_n] - S|}{|S_n - S|} = 0 \,. \tag{D.15}$$

The requirement of linear convergence of S_n is equivalent to asking that for n large enough

$$S_n \approx S + C\lambda^n \,, \tag{D.16}$$

for some constant C and for $|\lambda| < 1$. This allows us to estimate λ without knowing the limit S, as is often the case. In practice, if we knew the value of the limit S, we would not need to extrapolate S_n. The limit of differences

$$\lim_{n\to\infty} \frac{S_n - S_{n-1}}{S_{n-1} - S_{n-2}} = \lambda \,. \tag{D.17}$$

If the rate convergence of S_n is of higher order (quadratic or more), we do not need to extrapolate as the convergence is already very fast. However, we should look for a different extrapolation technique if the convergence is sub-linear. We refer to the Appendix of [9] for an explicit and straightforward example.

D.4 Diagrams of the melonic self-energy and vertex renormalization spinfoam amplitudes

This Appendix reports the self-energy and vertex renormalization spinfoam amplitudes' wiring diagrams, highlighting the internal faces' combinatorics. These are shown in Figure D.1 and Figure D.2. The notation for each intertwiner is used in equations (4.10) and (4.11). We do not explicitly label the spins so as not to clutter the picture.

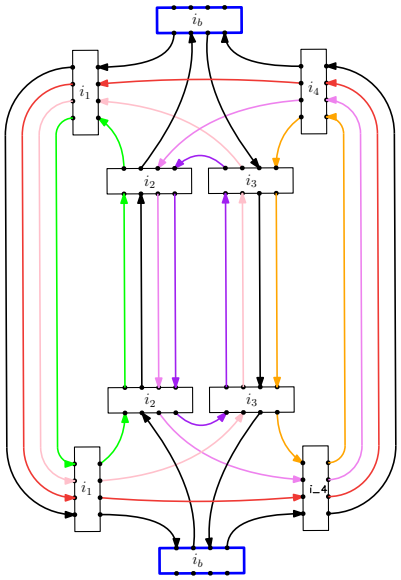


Figure D.1: *Wiring of the 2-complex of the self-energy spinfoam diagram. The internal faces are highlighted with different colors. Boundary intertwiners have a blue box corresponding to the integrals removed to regularize the amplitude.*

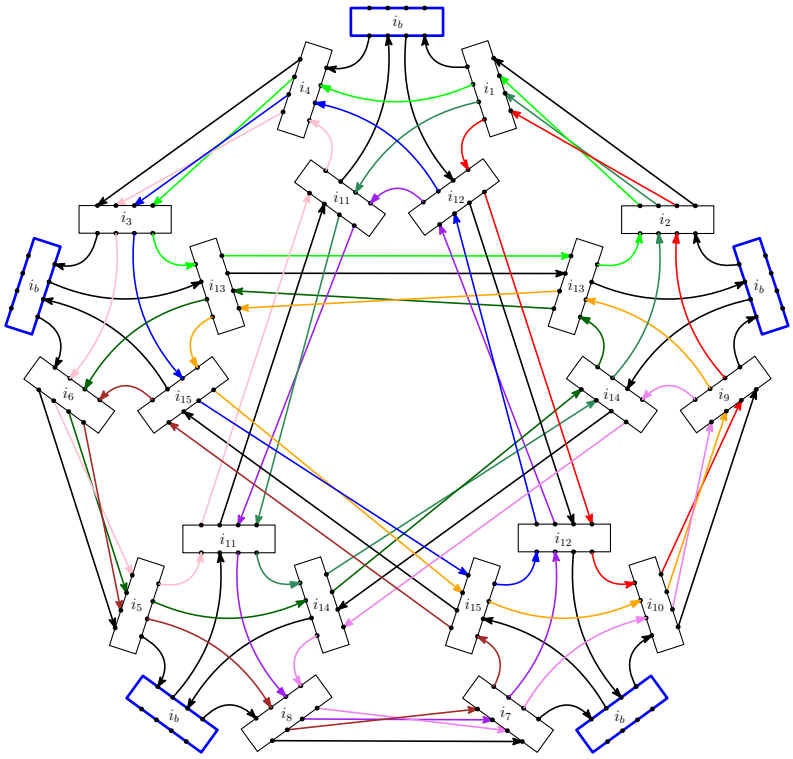


Figure D.2: *Wiring of the 2-complex of the vertex renormalization spinfoam diagram. The internal faces are highlighted with different colors. Boundary intertwiners have a blue box corresponding to the integrals removed to regularize the amplitude.*

Appendix E

Appendix Chapter 5

E.1 Discrete truncated normal distribution

We report in this appendix the definition of truncated normal distribution rounded to integers. For simplicity, we write the equations for a one-dimensional variable. The probability density function of a normal distribution $\mathcal{N}(x, \sigma)$ with mean zero and standard deviation σ is defined as:

$$\mathcal{N}(x, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} , \tag{E.1}$$

where $x \in R$. We can define the probability distribution function of a normal distribution with mean zero, and standard deviation σ rounded to integers as:

$$\mathcal{N}_d(n, \sigma) = \Phi(n + 0.5, \sigma) - \Phi(n - 0.5, \sigma) , \tag{E.2}$$

where $n \in N$ and $\Phi(x, \sigma)$ is the cumulative distribution function of a normal distribution with mean zero and standard deviation σ, defined as:

$$\Phi(x, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2\sigma^2}} dt . \tag{E.3}$$

For convenience, let us also define:

$$\Phi(a, b; \sigma) = \Phi(b, \sigma) - \Phi(a, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{a}^{b} e^{-\frac{t^2}{2\sigma^2}} dt . \tag{E.4}$$

The cumulative distribution function of a discrete (integer) Gaussian is written as:

$$\Phi_d(n_1, n_2; \sigma) = \mathcal{N}_d(n_1, \sigma) + \mathcal{N}_d(n_1 + 1, \sigma) + \cdots + \mathcal{N}_d(n_2, \sigma) . \tag{E.5}$$

With the above definitions, we can define the probability distribution function of a truncated normal distribution rounded to integers $\mathcal{N}_{d,t}(n, n_1, n_2; \sigma)$ between n_1 and n_2 as:

$$\mathcal{N}_{d,t}(n, n_1, n_2; \sigma) = \frac{\mathcal{N}_d(n, \sigma)}{\Phi_d(n_1, n_2; \sigma)} . \tag{E.6}$$

E.2 Metropolis-Hastings parameters

We report in the tables below the parameters used in the Metropolis-Hastings algorithm. These parameters are used for all calculations in this paper except for the data in Figures 5.17 and 5.6.

BF 4-simplex				
j	N_{MC}	b	σ	C
0.5	10^6	10^2	0.97	5
1.0	10^6	10^2	0.94	5
1.5	10^6	10^2	0.90	5
2.0	10^6	10^2	0.90	5
2.5	10^7	10^2	0.90	5
3.0	10^7	10^2	0.90	5
3.5	10^7	10^2	0.90	5
4.0	10^7	10^2	0.90	5
4.5	10^7	10^3	0.90	5
5.0	10^7	10^3	0.90	5
5.5	10^7	10^3	0.90	5
6.0	10^7	10^3	0.90	5

EPRL 4-simplex				
j	N_{MC}	b	σ	C
0.5	10^7	10^3	0.97	5
1.0	10^7	10^3	0.94	5
1.5	10^7	10^3	0.90	5
2.0	10^7	10^3	0.90	5
2.5	10^7	10^3	0.90	5
3.0	10^7	10^3	0.90	5
3.5	10^7	10^3	0.90	5
4.0	10^7	10^3	0.90	5
4.5	10^7	10^3	0.90	5
5.0	10^7	10^3	0.90	5
5.5	10^7	10^3	0.90	5
6.0	10^7	10^3	0.90	5

BF star				
j	N_{MC}	b	σ	C
0.5	10^6	10^3	0.40	32
1.0	10^6	10^3	0.39	32
1.5	10^6	10^3	0.37	32
2.0	10^6	10^3	0.35	32
2.5	10^6	10^3	0.35	32
3.0	$3\cdot10^6$	10^3	0.35	32
3.5	$3\cdot10^6$	10^3	0.35	32
4.0	$5\cdot10^6$	10^3	0.35	32
4.5	$5\cdot10^6$	10^3	0.35	32
5.0	$5\cdot10^6$	10^3	0.35	32
5.5	$7\cdot10^6$	10^3	0.35	32
6.0	$7\cdot10^6$	10^3	0.35	32

EPRL star				
j	N_{MC}	b	σ	C
0.5	10^7	10^3	0.40	32
1.0	$3\cdot10^7$	10^3	0.39	32
1.5	$3\cdot10^7$	10^3	0.35	32
2.0	$3\cdot10^7$	10^3	0.35	32
2.5	$3\cdot10^7$	10^3	0.35	32
3.0	$5\cdot10^7$	10^3	0.37	32
3.5	$5\cdot10^7$	10^3	0.37	32
4.0	$5\cdot10^7$	10^3	0.38	32
4.5	$5\cdot10^7$	10^3	0.39	32
5.0	$5\cdot10^7$	10^3	0.40	32
5.5	$8\cdot10^7$	10^3	0.40	32
6.0	$8\cdot10^7$	10^3	0.40	32

TABLES: Parameters used in the Metropolis-Hastings algorithm. From left to right: j is the spin attached to the links of the star spinfoam, N_{MC} is the number of Monte Carlo iterations, b is the number of burn-in iterations, σ is the standard deviation of the truncated normal proposal distribution and C corresponds to the number of Markov chains that we averaged to improve the statistic and measure the statistical fluctuations.

A general difference we observed between the BF and EPRL models is a greater statistical fluctuation in the expectation values of operators for EPRL as j increases, even though the dimension of the intertwiners' space is the same. To reduce the statistical fluctuations in EPRL,

we tried both to increase the number of Markov chains to be averaged by order of magnitude (in the code [103], the latter are automatically parallelized on the available cores) and to increase the number of Monte Carlo iterations N_{MC}. We found good precision in both cases, and for this paper, we decided to use the data obtained with the second approach, as shown in the tables.

Notice the role that the Metropolis-Hastings parameters play in the sampling process. While increasing the number of chains to be averaged has the effect of improving the accuracy in the determination of the operator's mean value (and the corresponding standard deviation), increasing the number of Monte Carlo iterations N_{MC} implies reducing the standard deviation of the statistical sampling. Therefore, satisfying statistical precision is achieved when these two parameters are sufficiently high and balanced. While we did not find relevant differences by modifying the number of burn-in iterations, we set the optimal standard deviation of the proposal distribution σ by requiring an acceptance rate of intertwiners draws $[i_n]$ around 30% in the sampling algorithm.

E.3 Booster functions

The booster functions [47], [45], also known as B4 functions [42], are the non compact residuals of the $SL(2,C)$ integrals. These functions encode all the details of the EPRL model, such as the $Y-$map. We define them as[1]:

$$B_4\left(j_f, l_f; i, k\right) \equiv \frac{1}{4\pi} \sum_{\{p_f\}} \begin{pmatrix} j_f \\ p_f \end{pmatrix}^{(i)} \left(\int_0^\infty \mathrm{d}r \sinh^2 r \prod_{f=1}^{4} d^{(\gamma j_f, j_f)}_{j_f l_f p_f}(r) \right) \begin{pmatrix} l_f \\ p_f \end{pmatrix}^{(k)} . \tag{E.7}$$

where $d^{(\rho,k)}(r)$ are the boost matrix elements for γ-simple irreducible representations of $SL(2,C)$ in the principal series and γ is the Immirzi parameter. In their most general formulation, the booster functions turn out to be the $SL(2,C)$ analogs of the usual Clebsch-Gordan coefficients for the rotation group $SU(2)$. The semi-classical limit of booster functions was discussed in [30]. The general explicit form of the boost matrix elements can be found in the literature [43, 42]. In the case of simple irreducible representations, these turn out to be [42]:

$$\begin{aligned} d^{(\gamma j, j)}_{jlp}(r) = \quad & (-1)^{\frac{j-l}{2}} \frac{\Gamma(j+i\gamma j+1)}{|\Gamma(j+i\gamma j+1)|} \frac{\Gamma(l-i\gamma j+1)}{|\Gamma(l-i\gamma j+1)|} \frac{\sqrt{2j+1}\sqrt{2l+1}}{(j+l+1)!} \left[(2j)!(l+j)!(l-j)! \frac{(l+p)!(l-p)!}{(j+p)!(j-p)!} \right]^{1/2} \\ & \times e^{-(j-i\gamma j+p+1)r} \sum_s \frac{(-1)^s e^{-2sr}}{s!(l-j-s)!} \, {}_2F_1[l+1-i\gamma j, j+p+1+s, j+l+2, 1-e^{-2r}] \,, \end{aligned} \tag{E.8}$$

where ${}_2F_1[a,b,c,z]$ is the hypergeometric function.

***

[1] In this Appendix, we do not indicate the dependence on multiple variables with the curly brackets in order not to weigh down the notation

Appendix F

Appendix Chapter 6

F.1 Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm is a Markov Chain Monte Carlo (MCMC) technique used to generate samples from a complex or high-dimensional target probability distribution. To discuss the algorithm, we consider a multidimensional discrete variable x on a state space χ, and a quantity O which can be computed as:

$$O = \frac{\sum\limits_{x\in\chi} W_\chi^2(x)\, o(x)}{\sum\limits_{x\in\chi} W_\chi^2(x)} \,, \tag{F.1}$$

where $W_\chi^2(x)$ is the unnormalized target distribution on χ. We refer to the normalized target distribution as follows:

$$f_\chi(x) \equiv \frac{W_\chi^2(x)}{\sum\limits_{x\in\chi} W_\chi^2(x)} \,, \tag{F.2}$$

Suppose $W_\chi(x)$ can be computed (up to a multiplying constant). In that case, the Metropolis-Hastings algorithm allows constructing an ergodic Markov chain on the space χ with length N_{mc}: $[x]_1, [x]_2, \ldots, [x]_{N_{mc}}$ such that each state is indirectly sampled from the normalized target distribution (F.2). We use the notation $[x]_s$ to denote the s-th state along the chain. A positive proposal distribution g_χ on the space χ is required to transit from each state to the next one.

We use the random walk Metropolis-Hastings, in which the sampler locally explores the neighborhood of the current state $[x]_s$ of the Markov chain, proposing a candidate state $[x]'$ sampling from a chosen probability distribution (usually a uniform or a normal distribution). That is, given the current state $[x]_s$, the algorithm suggests a candidate state depending on $[x]_s$. As proposal distribution, in this paper, we consider a truncated normal distribution rounded to integers. Before writing its expression, we first define the cumulative distribution function of a one-dimensional normal distribution with zero mean and standard deviation σ:

$$\Phi_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2\sigma^2}}\, dt \,. \tag{F.3}$$

If the space χ is multidimensional, g_χ is a multivariate distribution. We sample each component of x from a one-dimensional distribution independent of the others. We write the expression of

a one-dimensional normal distribution rounded to integers, truncated between n_1 and n_2, with standard deviation σ as:

$$g_{n_1,n_2,\sigma}(n) = \frac{\Phi_\sigma(n+0.5) - \Phi_\sigma(n-0.5)}{\sum_{k=n_1}^{n_2} [\Phi_\sigma(k+0.5) - \Phi_\sigma(k-0.5)]} . \tag{F.4}$$

For convenience, let us also define the truncated coefficients:

$$C_{n_1,n_2,\sigma}([x]) = \prod_{i=1}^{N} \left\{ \sum_{n=n_1-x_i}^{n_2-x_i} [\Phi_\sigma(n+0.5) - \Phi_\sigma(n-0.5)] \right\} , \tag{F.5}$$

where we defined with x_i the i-th component of the draw $[x]$. To build the Markov Chain, we compute the ratio between the target distribution (F.2) times the truncated coefficients (F.5) at the proposal state and the same quantity at the current state. Then, we accept the proposal state with a probability equal to this state. Otherwise, we stay at the current point. The initial steps of the algorithm are usually removed as burn-in iterations during the thermalization phase. The detailed implementation of the Metropolis-Hastings algorithm applied to spinfoams is described in Section 6.4.2 of the main text.

After storing a Markov chain with the desired length N_{mc}, we can compute (F.1) applying Monte Carlo on the multidimensional sum over x, using the chain itself as a statistical sample. When the number of samples N_{mc} is large enough, we obtain a reasonably good estimate of the original quantity:

$$O_{mc} = \frac{1}{N_{mc}} \sum_{s=1}^{N_{mc}} o([x]_s) \approx O \quad \text{for } N_{mc} \gg 1 . \tag{F.6}$$

The Monte Carlo estimate (F.6) is exactly equal to (F.1) only in the (ideal) limit of an infinite number of samples. The convergence is faster with respect to the standard version of Monte Carlo [138] (in which the draws are sampled randomly) because each draw is generated from the distribution (F.2).

There is a non-zero correlation between $[x]_s$ and $[x]_{s+d}$ where $d \geq 1$. This is because each proposed state depends on the previous one (as the process is Markovian). For each quantity (F.6) we can compute the autocorrelation function with lag d:

$$R_O(d) = \frac{\sum_{s=d+1}^{N_{mc}} (o([x]_s) - O_{N_{mc}})(o([x]_{s-d}) - O_{N_{mc}})}{\sum_{s=1}^{N_{mc}} (o([x]_s) - O_{N_{mc}})^2} . \tag{F.7}$$

Since the Markov Chain converges to a stationary distribution, the autocorrelation F.7 should decrease as the lag d increases.

Finally, we report in table F.1 the parameters used in the MCMC algorithm 3 for the calculations considered in this paper.

MH - parameters				
j	N_{mc}	b	σ	C
0.5	10^6	10	0.40	30
1.0	10^6	10	0.35	30
1.5	10^6	10	0.35	30
2.0	10^6	10	0.35	30
2.5	10^6	10	0.35	30
3.0	10^6	10	0.35	30
3.5	10^6	10	0.32	30
4.0	10^6	10	0.30	30

Table F.1: *Parameters used in the MCMC algorithm 3. From left to right: j is the boundary spin of spinfoam amplitude, N_{mc} is the number of iterations over the chain, "b" is the number of burn-in iterations, σ is the standard deviation of the proposal distribution and "C" corresponds to the number of Markov chains that we stored.*

F.2 $SU(2)$ recoupling coefficients

In this Appendix, we report some definitions of standard invariant Wigner symbols. We refer to [169, 218] for the analytical definitions and graphical representations of the general expressions which appear in the recoupling theory of $SU(2)$ representations.

We explicitly define the $4jm$ symbol, which is less common than the other Wigner symbols usually considered in the literature. It can be obtained by contracting two $3j$ Wigner symbols using an intertwiner i. Three inequivalent recouplings correspond to how four spins of $SU(2)$ representations can be coupled in pairs. Each different recoupling corresponds to a (different) $4jm$ symbol for the given set of spins j_1, j_2, j_3, j_4. We choose to couple spins j_1, j_2 with j_3, j_4:

$$\begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{(i)} \equiv \sum_{m_i} (-1)^{i-m_i} \begin{pmatrix} j_1 & j_2 & i \\ m_1 & m_2 & m_i \end{pmatrix} \begin{pmatrix} i & j_3 & j_4 \\ -m_i & m_3 & m_4 \end{pmatrix} . \tag{F.8}$$

The $4jm$ Wigner symbol (F.8) has the following orthogonality property:

$$\sum_{m_1,m_2,m_3} \begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{(i)} \begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{(i')} = \frac{\delta_{i,i'}}{d_i} \frac{\delta_{j_4,l_4}\delta_{m_4,n_4}}{d_{j_4}} , \tag{F.9}$$

Where the triangular inequality constrains the intertwiner i:

$$\max\{|j_1 - j_2|, |j_3 - j_4|\} \leq i \leq \min\{j_1 + j_2, j_3 + j_4\} . \tag{F.10}$$

To perform $SU(2)$ integrations and switch to intertwiners on spinfoam boundaries, we use the following property of the 4jm Wigner symbol:

$$\int_{SU(2)} du\, D^{j_1}_{n_1m_1}(u) D^{j_2}_{m_2n_2}(u) D^{j_3}_{m_3n_3}(u) D^{j_4}_{m_4n_4}(u) = \sum_i d_i \begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ n_1 & n_2 & n_3 & n_4 \end{pmatrix}^{(i)} \begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{(i)} , \tag{F.11}$$

where D^{j}_{nm} represents an $SU(2)$ Wigner matrix. We use the in-line notation for the 6j Wigner symbols

$$\{6j\}(j_1, j_2, j_3, j_4, j_5, j_6) = \left\{ \begin{array}{ccc} j_1 & j_2 & j_3 \\ j_4 & j_5 & j_6 \end{array} \right\} \tag{F.12}$$

And the 9j Wigner symbol:

$$\{9j\}(j_1, j_2, j_3, j_4, j_5, j_6, j_7, j_8, j_9) = \left\{ \begin{array}{ccc} j_1 & j_2 & j_3 \\ j_4 & j_5 & j_6 \\ j_7 & j_8 & j_9 \end{array} \right\} . \tag{F.13}$$

Finally, we report the definition of the 21j Wigner symbol split into 6j and 9j Wigner symbols:

$$\begin{aligned} \{21j\}\,(j, i_1, i_2, i_3, i_4, b_1, b_2, b_3, p_1, p_2) = \sum_{g_1} \sum_{g_2} \Big[\sum_{l} \{6j\}\,(j, i_3, j, j, i_4, l) \cdot \{6j\}\,(i_4, j, l, b_1, j, j) \cdot \\ \cdot \{6j\}\,(l, j, g_2, b_2, j, b_1) \cdot \{9j\}\,(l, j, i_3, j, i_2, j, g_2, g_1, j) \cdot d_l \cdot (-1)^{2l+i_3+3i_4+b_1+b_2+g_2} \Big] \cdot \\ \cdot \{6j\}\,(i_2, j, j, i_1, j, g_1) \cdot \{6j\}\,(j, p_1, j, j, i_1, g_1) \cdot \{6j\}\,(j, b_3, p_2, j, g_2, b_2) \cdot \{6j\}\,(j, p_2, p_1, j, g_1, g_2) \cdot \\ \cdot d_{g_1} d_{g_2} \cdot (-1)^{2g_1+3i_1+j+i_2+3g_2+2b_2+b_3+2p_1} \ . \end{aligned} \tag{F.14}$$

Appendix G

Appendix Chapter 7

G.1 Wigner symbols

In this paper, we use the definition of the 3j Wigner symbol provided in [218]. It has the following orthogonality property:

$$\sum_{m_1,m_2} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & n_3 \end{pmatrix} = \frac{\delta_{j_3 l_3} \delta_{m_3 n_3}}{2j_3 + 1} \ . \tag{G.1}$$

The 3j Wigner symbol vanishes if triangular inequalities are not satisfied. We define the 4jm Wigner symbol as the contraction of two 3j symbols over an internal spin k:

$$\begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{(k)} = \tag{G.2}$$

$$= \sum_{m_i} (-1)^{k-m_i} \begin{pmatrix} j_1 & j_2 & k \\ m_1 & m_2 & m_i \end{pmatrix} \begin{pmatrix} k & j_3 & j_4 \\ -m_i & m_3 & m_4 \end{pmatrix} .$$

We also use the synthetic notation:

$$\begin{pmatrix} j_f \\ m_f \end{pmatrix}^{(k)} \equiv \begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{(k)} \ . \tag{G.3}$$

With the definitions (G.2) and (7.11) we have:

$$\int dn \, D^{j_1}_{m_1,n_1}(n) \, D^{j_2}_{m_2,n_2}(n) \, D^{j_3}_{m_3,n_3}(n) \, D^{j_4}_{m_4,n_4}(n)$$

$$= \sum_k d_k \begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{(k)} \begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ n_1 & n_2 & n_3 & n_4 \end{pmatrix}^{(k)} . \tag{G.4}$$

A valuable property of Wigner matrices is the following:

$$D^j_{j,m}\left(n^{-1}\right) = (-1)^{j-m} D^j_{-m,-j}(n) \ . \tag{G.5}$$

We use the irreducible 15j Wigner symbol of the first kind, following the conventions of [169]. Its definition in terms of Wigner's 6j symbols turns out to be:

$$\{15j\} = (-1)^{\sum_{i=1}^{5} j_i+l_i+k_i} \sum_s d_s \begin{Bmatrix} j_1 & k_1 & s \\ k_2 & j_2 & l_1 \end{Bmatrix} \begin{Bmatrix} j_2 & k_2 & s \\ k_3 & j_3 & l_2 \end{Bmatrix} \begin{Bmatrix} j_3 & k_3 & s \\ k_4 & j_4 & l_3 \end{Bmatrix} \begin{Bmatrix} j_4 & k_4 & s \\ k_5 & j_5 & l_4 \end{Bmatrix} \begin{Bmatrix} j_5 & k_5 & s \\ j_1 & k_1 & l_5 \end{Bmatrix} . \tag{G.6}$$

G.2 Booster functions

The booster functions are the non-compact remnants of the $SL(2,\mathbb{C})$ group [42, 45]. For the physical interpretation of the booster functions and their semiclassical limit, we refer to [30]. We define them as follows:

$$B_4\left(j_f, l_f; i, k\right) \equiv \frac{1}{4\pi} \sum_{p_f} \begin{pmatrix} j_f \\ p_f \end{pmatrix}^{(i)} \left(\int_0^\infty \mathrm{d}r \sinh^2 r \prod_{f=1}^{4} d^{(\gamma j_f, j_f)}_{j_f l_f p_f}(r) \right) \begin{pmatrix} l_f \\ p_f \end{pmatrix}^{(k)} , \tag{G.7}$$

where γ is the Barbero-Immirzi parameter and $d^{(\rho,k)}(r)$ are the matrix elements for γ-simple irreducible representations of $SL(2,\mathbb{C})$. The explicit form of the boost matrix elements can be found in [43, 42]. We report below the case of simple irreducible representations:

$$\begin{aligned} d^{(\gamma j, j)}_{jlp}(r) =& (-1)^{\frac{j-l}{2}} \frac{\Gamma\left(j + i\gamma j + 1\right)}{\left|\Gamma\left(j + i\gamma j + 1\right)\right|} \frac{\Gamma\left(l - i\gamma j + 1\right)}{\left|\Gamma\left(l - i\gamma j + 1\right)\right|} \frac{\sqrt{2j+1}\sqrt{2l+1}}{(j+l+1)!} \left[(2j)!(l+j)!(l-j)! \frac{(l+p)!(l-p)!}{(j+p)!(j-p)!} \right]^{1/2} \\ & \times e^{-(j-i\gamma j+p+1)r} \sum_s \frac{(-1)^s e^{-2sr}}{s!(l-j-s)!} {}_2F_1[l+1-i\gamma j, j+p+1+s, j+l+2, 1-e^{-2r}] . \end{aligned} \tag{G.8}$$