We adapt the Metropolis-Hastings algorithm, along with recently developed computational methods, to study the quantum regime in spinfoams with many degrees of freedom. This approach fills a gap in the current numerical methods for computing spinfoam observables in the low-spin regime of covariant Loop Quantum Gravity. We test the method by studying a refinement of the 4-simplex amplitude, where the boundary is refined from 5 to 20 boundary tetrahedra. We investigate the BF and Lorentzian Engle-Pereira-Rovelli-Livine (EPRL) models, computing boundary geometrical operators, correlation functions and entanglement entropy. We are able to confirm that the transition amplitudes are stable against this refinement, even in the physically relevant EPRL case: the boundary geometry does not change on average, but the new degrees of freedom modify the quantum fluctuations of the boundary and the correlations between different spatial patches. The expectation values obtained numerically are compatible with their geometrical interpretation and, interestingly, the correlations between neighbouring patches decay sharply when computed across spinfoam vertices.

I. INTRODUCTION

The last years have seen a steady development of the use of sophisticated numerical methods in the covariant formulation of Loop Quantum Gravity, referred to as spinfoam. The evaluation of spinfoam amplitudes in the Engle-Pereira-Rovelli-Livine model (henceforth EPRL) [1–4], the introduction of effective models [5, 6] and the numerical study of cuboid renormalization [7] 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. 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 [8, 9]. These methods become essential when simulations involve a large number of degrees of freedom. In [8] 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, and found good agreement with the semiclassical results obtained via analytical methods. The problem of computing expectation values and correlations in spinfoams with many degrees of freedom in the full quantum regime, i.e. in a context in which the usual semiclassical approximations are not applicable, on the other hand, remains open.

This paper addresses this problem by introducing a technique that can be applied in a regime in which the spin quantum numbers are small, but the number of degrees of freedom is large. We combine the Metropolis-Hastings algorithm [10] with some recently developed high performance computing techniques in covariant LQG [11]. Using these, we study a spinfoam with 6 vertices in the bulk and 20 nodes on the boundary. This corresponds to a cellular decomposition obtained from one elementary 4-simplex by splitting each of the 5 boundary tetrahedra into 4 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. In the following, we refer to it as the “star” spinfoam.

We restrict the calculation to the symmetric sector where the spins of all boundary links have the same value. In this sector, the spinfoam degrees of freedom are given by the boundary intertwiners, which encode the shapes of the boundary tetrahedra. We compute the amplitude as a function of these variables numerically, and use the Monte Carlo sampling to study expectation values of different boundary operators and
their correlations. For the sake of completeness, we investigate both the BF and EPRL models. We compare the results obtained in this larger graph with the same results for the 4-simplex. The boundary state that we study was introduced in [12] can be interpreted as the quantum nothing-to-geometry transition amplitude. It avoids the problem of dealing with non-positive-definite sampling probabilities. This state is rich enough to describe a boundary geometry that is regular on average but allows quantum fluctuations.

A key open question in the spinfoam approach to quantum gravity is the convergence of the amplitudes under refinement of the 2 complex of the spinfoam. In this regard, we are able to 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 on the refined graph. 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 provides 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 II we write the expressions of the LQG vertex amplitude in the topological BF and Lorentzian EPRL model employed in the algorithm. In Section III we define the boundary state that we use in our analysis, then we describe the spinfoam considered in this work. In Section IV we discuss the Markov chain Monte Carlo method applied to the spin sums over the boundary degrees of freedom. In Section V we present the numerical results of certain geometrical operators (boundary angles and volumes) and the related correlations. In Section VI we 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 [13]1. All the computations were performed on Compute Canada’s Cedar, Graham, Beluga and Narval clusters (www.computecanada.ca). The computational resources employed for this paper can be quantified in approximately 7 days of computations distributed over 600 CPUs.

II. THE VERTEX AMPLITUDE

In this Section we write the explicit expression for the four-dimensional vertex amplitudes of the SU(2) BF and Lorentzian EPRL models. These can be interpreted as the “elementary building blocks” for the Monte Carlo sampling algorithm (see Section IV), as the spinfoam model (discussed in Section III A) can be easily decomposed into such elementary amplitudes. The BF and EPRL vertices can be efficiently computed with the sl2cfoam-next library [11], which is the improved version of the original sl2cfoam computational library [14–17]. For an explicit tutorial on how to use the library and compute vertex amplitudes, see also [4].

A. BF vertex amplitude

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.

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).
following the convention of [18]. The definition in terms of \{6j\} symbols is:

\[
V_{BF}(j_{ab}; i_a) = (-1)^{\sum_{k=1}^{5} j_{k} + i_{k}} \sum_{s} d_s \begin{bmatrix}
  i_{1} & j_{25} & s \\
  i_{5} & j_{14} & j_{15}
\end{bmatrix} \begin{bmatrix}
  j_{35} & i_{5} & j_{45} \\
  j_{24} & i_{3} & j_{34}
\end{bmatrix} \begin{bmatrix}
  j_{12} & i_{1} & j_{25} \\
  j_{34} & i_{2} & j_{23}
\end{bmatrix} \times \begin{bmatrix}
i_{4} & j_{35} & s \\
i_{3} & j_{24} & j_{34}
\end{bmatrix} \begin{bmatrix}
  j_{13} & i_{2} & j_{23} \\
  j_{45} & i_{4} & j_{45}
\end{bmatrix}
\]

(1)

where \(a, b = 1 \ldots 5\), \(a \neq b\) and \(d_s = 2s + 1\). In order 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 (1) can be computed efficiently with libraries such as \texttt{wigxjpf} and especially \texttt{fastwigxj} [19, 20].

**B. EPRL vertex amplitude**

The EPRL vertex amplitude is built from the topological \(SL(2,C)\) spinfoam vertex amplitude once that the simplicity constraints have been imposed [21, 22]. We use the form of amplitude originally derived in [23], which results in a linear combination of \{15j\} symbols weighted by one booster functions \(B_{4}\) per edge (see Appendix [C] for explicit formulas). We write the vertex amplitude according to the graphical notation discussed in detail in [4]:

\[
V_{EPRL}^{\gamma}(j_{ab}, i_{a}; \Delta l) = \sum_{l_{eq} = j_{eq}} \bigg( \prod_{e} d_e B_{4}^{e}(j_{1e}, l_{eq}; i_{e}, k_{e}) \bigg) V_{BF}(j_{1e}, l_{eq}; k_{e}, i_{a})
\]

(2)

where \(e, q = 2 \ldots 5\), \(e \neq q\). We introduced the \(\Delta l\) parameter in order 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 [3, 24, 25]. In this paper we consider \(\gamma = 1.2\) as value of Barbero-Immirzi parameter. The 4 spins associated with the gauge-fixed edge are \((j_{12}, j_{13}, j_{14}, j_{15})\), as the elimination of a redundant \(SL(2,C)\) integration along one edge in the EPRL vertex (2) is necessary to ensure that the corresponding amplitude is well defined [26]. As in (1) we neglected the dimensional factors attached to the boundary intertwiners and spins.

\[
V_{EPRL}^{\gamma}(j_{ab}, i_{a}; \Delta l) = \sum_{l_{eq} = j_{eq}} \bigg( \prod_{e} d_e B_{4}^{e}(j_{1e}, l_{eq}; i_{e}, k_{e}) \bigg) V_{BF}(j_{1e}, l_{eq}; k_{e}, i_{a})
\]

(3)
III. THE BOUNDARY STATE

We use the boundary state $|\psi_0\rangle$ originally introduced in [12], which studied the simplest possible graph, i.e. the boundary of a single vertex amplitude. This is the complete graph with five nodes. In general, let $\Gamma$ be a graph with $L$ links and $N$ nodes. The LQG Hilbert space for the graph is:

$$H_\Gamma = L^2[SU(2)^L/SU(2)^N].$$

The spin network basis in $H_\Gamma$ is made by the states $|\{j_l\},\{i_n\}\rangle$ (from now on, we omit the $\Gamma$ subscript), where $\{j_l\}$ is a set of half-integer spins and $\{i_n\}$ an intertwiner set, $n = 1 \ldots N$, $l = 1 \ldots L$. 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 symmetry-reduced space as:

$$|j,\{i_n\}\rangle \equiv |j,i_1\ldots i_N\rangle = |j,i_1\rangle \otimes \cdots \otimes |j,i_N\rangle,$$

suppressing the curly brackets for the spin label $j$, as there is one common spin attached to all the links. In the following we use the EPRL Lorentzian model to explicitly show the calculations, since the procedure with the topological model BF is the same. The only difference between the EPRL and BF model is in the coefficients of the state $|\psi_0\rangle$ in the basis (5).

We define the state $|\psi_0\rangle$ in the Hilbert space (4) by

$$\langle j,\{i_n\}|\psi_0\rangle \equiv A(j,\{i_n\}).$$

where $A(j,\{i_n\})$ is the LQG Lorentzian amplitude of the state in the spin network basis. The amplitude (6) can be interpreted as the amplitude associated to the transition nothing-to- $|j,\{i_n\}\rangle$. Hence $|\psi_0\rangle$ gives the natural state that is projected out of the empty state by the LQG dynamics. The amplitude function depend on the common spin $j$ on the links and on all the $N$ intertwiner indices. If we consider a spinfoam made by a single 5-valent vertex, the amplitude (6) with $N=5$ can be computed numerically with the expression (2) at a finite value of $\Delta l$, for a fixed Immirzi parameter $\gamma$. The state $|\psi_0\rangle$ is therefore defined as:

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

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 a total of $(2j + 1)^N$ boundary basis elements that enter the sum (7). Following the geometrical interpretation of the covariant LQG phase space in terms of twisted geometries [27] 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 of freedom model the “shape” of the boundary tetrahedra, and these are relational observables at given value $j$. They are directly linked to the boundary 3d dihedral angles, as discussed in Section V A.

A. The triangulation and graph

The 2-complex of the star is composed by 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 by five 4-simplices glued on 5 internal tetrahedra, each 4-simplex showing 4 tetrahedra on its boundary. The triangulation of the star graph is showed in Figure 1 along with the boundary spin network.
The triangulation of the star model is non-regular, since there are segments that are shared 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. In Appendix A we give a diagrammatic expression for the EPRL star spinfoam amplitude in terms of vertex amplitudes (2).

In this paper we consider the analysis of the star model boundary graph up to the value $j = 6$ for the spins associated with the boundary links. There are several reasons for which we perform the computations up to this value:

• This numerical approach is intended to be applied in the full quantum regime, i.e., when the spin quantum numbers are small. The value $j = 6$ is therefore fit for this purpose, and the analysis of numerical results do not require increasing the boundary spins.

• The computational complexity represented by increasing $j$ strongly depends on the type of the considered operator, as well as on the Metropolis-Hastings parameters (see Appendix B for details). The value $j = 6$ allows to compute all the operators and correlation functions that we consider with a stable precision up to 3 significant digits.

• In order to perform the sampling algorithm it is necessary to pre-compute the elementary vertex amplitudes (see Section II) at 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 II the computational time exponentially increases with the truncation parameter $\Delta l$. Although we found an excellent approximation even with $\Delta l = 0$ for most of the operators and correlations considered in this paper, we decided to select $\Delta l = 20$ in the computation of the EPRL vertex amplitude (2) for $j = 0 \ldots 6$. The calculation, distributed over several machines and hundreds of CPUs, took about 5 days to complete. The computed amplitudes are available at the public repository [13], along with the corresponding BF counterparts [1].

B. Expectation values

We consider local geometrical operators acting on single boundary nodes of $\mathcal{H}$. For each operator, we specify the matrix elements in the basis states (5). We start defining the normalized expectation value on

\[ \langle \psi | O | \psi \rangle \]

\[ \text{with the exception of the dihedral angle operator (see Section V A), for which } \Delta = 2 \text{ determines a slight systematic shift in the expectation values, for all the other operators we found no differences between } \Delta = 2 \text{ and } \Delta = 20. \]
Normalized correlations are defined as:
\[ \langle O_k \rangle \equiv \frac{1}{Z} \langle \psi_0 | O_k | \psi_0 \rangle . \] (8)

The normalization factor is computed as:
\[ Z \equiv \langle \psi_0 | \psi_0 \rangle = \sum_{\{i_n\}} A (j, \{i_n\})^2 . \] (9)

From (7) we write:
\[ \langle \psi_0 | O_k | \psi_0 \rangle = \frac{1}{Z} \sum_{\{i_n\}} \sum_{\{i'_n\}} A (j, \{i_n\}) A_T (j, \{i'_n\}) \langle j, \{i'_n\} | O_k | j, \{i_n\} \rangle . \] (10)

By using the orthogonality of the spin-network states (5) we find:
\[ \langle j, \{i'_n\} | O_k | j, \{i_n\} \rangle = \delta_{i'_n, i_n} \cdots \delta_{i'_1, i_1} \delta_{i'_m, i_m} ; \] (11)

therefore we conclude:
\[ \langle O_k \rangle = \frac{1}{Z} \sum_{\{i_n\}} \sum_{i'_k = 0}^{2j} A (j, \{i_n\}) A (j, \{i_n\}, i'_k) \langle j, i'_k | O_k | j, i_k \rangle , \] (12)

where \( A (j, \{i_n\}, i'_k) \) is defined as:
\[ A (j, \{i_n\}, i'_n) = A (j, i_1 \cdots i'_k \cdots i_N) , \] (13)

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 \( \langle O_k O_m \rangle \), which turns out to be:
\[ \langle O_k O_m \rangle = \frac{1}{Z} \sum_{\{i_n\}} \sum_{i'_k = 0}^{2j} \sum_{i''_m = 0}^{2j} A (j, \{i_n\}) A_T (\{j\}, \{i_n\}, i'_k, i''_m) \langle j, i'_k | O_k | j, i_k \rangle \langle j, i''_m | O_m | j, i_m \rangle , \] (14)

where the meaning of \( A_T (\{j\}, \{i_n\}, i'_k, i''_m) \) is transparent by looking at (13). That is, 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 (12) and (14) become respectively:
\[ \langle D_k \rangle = \frac{1}{Z} \sum_{\{i_n\}} A^2 (j, \{i_n\}) \langle j, i_k | D_k | j, i_k \rangle , \] (15)
\[ \langle D_k D_m \rangle = \frac{1}{Z} \sum_{\{i_n\}} A^2 (j, \{i_n\}) \langle j, i_k | D_k | j, i_k \rangle \langle j, i_m | D_m | j, i_m \rangle . \] (16)

Normalized correlations are defined as:
\[ C (O_k, O_m) = \frac{\langle O_k O_m \rangle - \langle O_k \rangle \langle O_m \rangle}{(\Delta O_k)(\Delta O_m)} , \] (17)

where the quantum spread is:
\[ \Delta O_k = \sqrt{\langle O_k^2 \rangle - \langle O_k \rangle^2} . \] (18)

The fact that the connected correlation function (17) between the nodes \( k \) and \( m \) is non-vanishing turns out to be a necessary condition in order to have correlated fluctuations between the shapes of the tetrahedra dual to nodes \( k \) and \( m \). [28][32].
IV. MONTE CARLO OVER INTERTWINER SPACE

The computation of the expectation value of an operator requires to sum (or integrate, for a continuous spectrum) over all possible eigenstates of the quantum system. Numerically, this rapidly becomes intractable as the number of degrees of freedom increases. In our case, for a graph with $N$ boundary intertwiners there are $(2j + 1)^N$ values to compute and to sum. Suppose that the amplitude function can be computed in $10^{-6}$ seconds on a reference hardware (the real time is many orders of magnitude larger, especially with many vertices). For the star spinfoam described in Section III A with 20 boundary tetrahedra, a spin $j = 2$ computation would take 3 years. Obviously, we cannot use blind summation if we want to approach this problem numerically. Clearly 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 is a technique that it is used to compute expectation values of random variables, i.e. integrals (or sums for discrete variables) of the form:

$$E[f(X)] = \int f(X) dP[X] ,$$

with $dP$ the probability measure induced by the random variable $X$. The application of this method to quantum mechanics, with the modulus squared of the amplitude function being the probability measure, is called Quantum Monte Carlo. The most efficient methods use a Markov chain to speed up the convergence of the procedure. We obtained the best results by adapting the Metropolis-Hastings algorithm [10] to the discrete sums over the boundary intertwiners. We consider the discrete probability density function proportional to the squared amplitude $A$:

$$p = \frac{A^2(j, \{i_n\})}{\sum_{\{i_n\}} A^2(j, \{i_n\})} ,$$

proceeding by running a stochastic sampling routine that extract draws$^3$ of intertwiners $[i_1 \ldots i_N]$ from their whole configuration space, according to the Markov chain. In the following, we refer to such routine as random walk. Specifically, we employ the Metropolis-Hastings algorithm$^4$ and we consider the star spinfoam amplitude in Figure 15. At each step of the random walk, we perform the contraction of the elementary vertex amplitudes over the 5 bulk intertwiners, and the $N = 20$ boundary intertwiners of the spinfoam are determined by the sampler in the random walk over the intertwiner space. For low spins, it is sufficient to perform the contraction with HPC techniques exploiting solely the CPU. We found the best performance using the LoopVectorization Julia package. We provide an explicit flowchart$^3$ which shows the steps in order to implement the random walk over the intertwiner space and build the Markov chain, which can be used to compute operators.

Algorithm 1 Numerical algorithm to perform random walk over the intertwiner space

1: Pre-compute the vertex amplitude at fixed boundary spins $j$ for all the possible intertwiners $i_1 \ldots i_5$.
2: Choose the total number of Monte Carlo iterations $N_{MC}$ and set the remaining Metropolis-Hastings parameters
3: Load the computed vertex amplitudes into memory
4: Set an initial configuration of boundary intertwiners $[i_1 \ldots i_N]$ and compute $A(j, [i_1 \ldots i_N])$
5: for $n = 1 \ldots N_{MC}$ do
6:  Generate a candidate draw $[i_1c \ldots i_Nc]$ from the current one
7:  Assemble the spinfoam amplitude $A(j, [i_1c \ldots i_Nc])$
8:  Compute $p = \min\{1, \frac{A^2(j, [i_1c \ldots i_Nc])}{A^2(j, [i_1 \ldots i_N])} \cdot \frac{C}{C_{prop}} \}$
9:  Generate a uniform random number $r$ between 0 and 1
10: If $p > r$ accept the candidate intertwiner draw and set $[i_1 \ldots i_N] = [i_1 \ldots i_Nc]$
11: If $p < r$ reject the candidate intertwiner draw and the current one remains the same
12: Save the current intertwiner draw
13: Dump to disk all the intertwiners draws

---

$^3$ Since intertwiners are subject to triangular inequalities, we sample from a truncated distribution, which involves the introduction of truncated coefficients $C$ and $C_{prop}$.

$^4$ We refer to the original paper [10] or to the numerous texts available for a description of the algorithm.
The final result of the algorithm is a Markov chain with length $N_{MC}$ constituted by intertwiner draws. The computationally most expensive part of the algorithm is undoubtedly the calculation of the spinfoam amplitude for each step of the Markov chain, which requires the use of high performance computing techniques to obtain appreciable results. The offloading of tensor contractions on the GPU with parallelization on the GPU cores [33], exploiting the recent tensor network techniques [34], will be implemented in future works. In fact, the best improvement is obtained for large values of the spins [35], making this approach more suitable for a study of the semiclassical limit of spinfoams rather than the quantum regime. We show a benchmark of the random walk sampling algorithm in Figure 2 for increasing values of the total number of iterations $N_{MC}$ in the Markov chain. The sampling in Figure 2 have been carried out in a few seconds 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 burnin parameter equal to $10^3$. In the code available at the repository [13] the Markov chains are automatically parallelized on the number of available CPUs, eventually distributing the computation on multiple machines. As discussed in Section V, building more Markov chains is useful for improving accuracy and estimating the error committed due to the statistical fluctuations of the random walk.

![Random walk benchmark](image)

FIG. 2: Benchmark of the sampling and storage process of intertwiners draws in the random walk over the 20-dimensional intertwiners’ space of the star spinfoam amplitude. Computation time asymptotically scales as $\sim j^{4.5}$.

After storing the intertwiner draws we can use them to compute expectation values of operators. The computation is stochastic in nature and the correct result is found only in the limit of an infinite number of samples. However, the soundness of the procedure comes from known theorems on Monte Carlo summation and we can estimate the error done by comparing many different “runs”\(^5\). The parameters used for the sampling of the draws employed for the computation of the operators are discussed in detail in Appendix B.

### A. Expectation values with Monte Carlo

We can summarize the introduction of the Monte Carlo method with the following substitution in the formulas of the expectation values of the operators:

$$\sum_{\{i_n\}} A^2(j, \{i_n\}) f (\{i_n\}) \approx \sum_{[i_n]} f ([i_n]).$$  \hspace{1cm} (21)

In the right side of equation (21), the sum over the intertwiners is intended as the sum over the stored draws $[i_n] \equiv [i_1 \ldots i_N]$ in which the intertwiners have a fixed value compatible with triangular inequalities. That is, we are no longer considering all the independent summations over the intertwiners. This hugely reduces the computational cost, making the computation feasible. To get an idea of the gain obtained with respect to

\(^5\) that is, each Markov chain previously sampled and stored in the random walk phase
the 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 a number of Monte Carlo iterations \( N_{MC} \sim 10^6 \). The exact sum would require performing \((2j + 1)^{20} \sim 10^{22}\) sums. Therefore, the Metropolis-Hastings algorithm adapted to the spinfoam formalism allows to reduce the computation complexity of about 16 orders of magnitude.

With \([21]\), the normalization factor \([9]\) becomes:

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

We can easily find the expression for the expectation values of operators by multiplying and dividing for \( A(j, \{i_n\}) \) and then using \([21]\), remembering \([22]\). In fact, \([12]\) becomes:

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

Equation \([14]\) becomes:

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

Equation \([15]\) and \([16]\) we obtain:

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

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

Notice that in the case of diagonal operator, it is not necessary 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.

V. NUMERICAL RESULTS: OPERATORS

In this Section we describe the numerical values obtained for the expectation value of local geometric operators \([12]\) with the boundary state \([7]\). We also compute the quantum spread \([18]\) and correlation functions \([14]\) between different nodes. For each geometrical operator, we discuss the results obtained both with the BF model and the EPRL model. In order to estimate the statistical fluctuations due to the Monte Carlo sampling, we compare the results of operator’s expectation values over different runs. That is, we store multiple Markov chains according to algorithm \([1]\) computing operators for each one of them. In fact, 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 \([25]\) of an operator \(D_k\) for each chain \(\langle D_k \rangle_1 \ldots \langle D_k \rangle_C\) and then consider the corresponding average and standard deviation:

\[
\mu_{\langle D_k \rangle} = \frac{\sum_{c=1}^{C} \langle D_k \rangle_c}{C} ,
\]

\[
\sigma_{\langle D_k \rangle} = \sqrt{\frac{\sum_{c=1}^{C} (\mu_{\langle D_k \rangle} - \langle D_k \rangle_c)^2}{C}} .
\]

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

\[
G(x; \mu_{\langle D_k \rangle}, \sigma_{\langle D_k \rangle}) = \frac{1}{\sigma_{\langle D_k \rangle} \sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{x - \mu_{\langle D_k \rangle}}{\sigma_{\langle D_k \rangle}} \right)^2} .
\]
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 Table I. Crucially, notice that for the EPRL model the Markov chains have a length $N_{MC}$ greater than about an order of magnitude compared to those used for sampling in BF, a characteristic that considerably reduces the standard deviation in (29), as discussed in the Appendix B.

A. The dihedral angle operator

The simplest operator to compute in the intertwiner basis 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 chosen for the invariant $SU(2)\{15j\}$ symbol, which appears both in the EPRL (2) and BF amplitude (1). The external dihedral angle of the tetrahedron dual to the node $n$ in the symmetry-reduced space basis states (5) is [12]:

$$\langle j, i_n | \cos(\theta) | j, i_n \rangle = \frac{i_n(i_n + 1) - 2j(j + 1)}{2j(j + 1)}.$$  \hfill (30)

The dihedral angle operator (30) is diagonal in the spin-network basis, therefore we can compute it very fast with equation (25). For this reason, since in our case all boundary tetrahedra are equal and regular, we can improve the statistic with a further average. Namely, we can compute the expectation value of the dihedral angle (30) and the corresponding quantum spread (18) for all 20 nodes of the spinfoam independently at fixed boundary spin $j$, then averaging the results, which are shown in Figure 3. The results show that the expectation value of the boundary dihedral angle (30) is peaked to the value corresponding to an equilateral tetrahedron, which is the same result originally obtained in [12] with the simplest possible triangulation of a 3-sphere. This indicates that in the evolution from 1 to 6 vertices according to the process described in Section III A, the spatial metric of the boundary state still averages to that of the 3-boundary of a regular 4-simplex, i.e. to that of a 3-sphere. This is not a trivial consequence of the symmetry reduction (5), but turns out to be a dynamical result of the global geometry. In fact, in (15) we are considering the sum over all the spinfoam boundary intertwiners. Different geometries might give different (non-regular) average boundary angles, which is for example the case of the $\Delta_3$ spinfoam graph [1].

The corresponding quantum spread $\Delta \cos(\theta)$ is shown in Figure 4. It turns out to be rapidly increasing for EPRL and slightly increasing for the BF model. As originally noticed in [12], 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 particular boundary state considered.

\[\text{FIG. 3: Expectation values (25) of the dihedral angle operator (30), averaged over all the 20 nodes of the spinfoam. The orange line shows the value of the cosine of dihedral angle of a regular tetrahedron, which is } \cos(\theta_{\text{regular}}) = -0.3.\]

6 It is important to underline that this step is justified a posteriori once it has been verified that the expectation value of the operator over all the nodes is identical.
here and its symmetry reduction, which fixes the areas of the boundary triangles at the quantum level, implying that the boundary angles are quantum totally spread.

The results suggest 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 [12], preliminary results on the 16-cell spinfoam model, which constitutes the second regular triangulation of the 3-sphere after the 4-simplex, exhibit a striking similar behavior. The Gaussian distributions (29), measuring the statistical fluctuations in the Monte Carlo sampling, are shown in Figure 5.

For the sake of clarity, we underline that the statistical fluctuations in Figure 5 were computed by averaging the expectation value (25) for the operator (30) on a single node over several runs, according to (29), while in Figure 3 we performed a further average on the 20 nodes of the spinfoam. 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 behaviour is the main reason why we used a number of \( N_{MC} \) iterations larger than one order of magnitude in the EPRL model (see Table I). We computed the correlation functions (17) between dihedral angles for all the 190 independent nodes combinations of the spinfoam. The result is shown in Figure 6. For the dihedral angle operator (30), 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 same vertex, we get the same correlations originally computed in [12]. 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 is in agreement with the results on the entanglement entropy, as discussed in Section VI. The numerical results show that the EPRL and BF models give rise to essentially indistinguishable dynamic
correlations in the case of the dihedral angle operator (30). This suggests that, at least in the approximation described in Section III, the SU(2) topological model (typically much easier to compute) provides an excellent approximation for studying dynamical correlations.

![BF dihedral angles correlations](image1)

**FIG. 6:** Expectation values (26) of the correlations between dihedral angle operators (30). The correlations between EPRL and BF are essentially indistinguishable.

### B. The volume operator

There are two slightly different prescriptions for the volume operator in LQG. Here we follow the Rovelli-Smolin prescription, see [36]. Since the general expression of the volume matrix elements in the spin-network basis is not trivial [37], here we limit ourselves to the equations in symmetric-reduced space of (4) in which all the spins have the same value \( j \) and the basis states are given by (5).

Let \( A \) be the \((2j+1) \times (2j+1)\) Hermitian matrix:

\[
A = i \times \begin{pmatrix}
0 & -a_1 & 0 & 0 & \cdots & 0 \\
a_1 & 0 & -a_2 & 0 & \cdots & 0 \\
0 & a_2 & 0 & -a_3 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 0 \\
\end{pmatrix}
\]

where the coefficients \( a_k \) are defined as:

\[
a_k = \frac{1}{4} \frac{k^2((2j+1)^2 - k^2)}{\sqrt{4k^2 - 1}}
\]

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:

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

Contrary to the dihedral angle (30), the volume operator (33) is not diagonal in the basis (5). In terms of expectation values (23) and correlations (24), 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 (14) which would be required without the Monte Carlo approximation (21)). This means that computing the expectation value (23) in the case of volume operator (33) for more than 1 node of the spinfoam takes too long. The expectation values of the volumes are shown in Figure 7, in which we neglected all the constant factors in the expression (33) 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 actually 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 \( \sqrt{j(j+1)} \approx j \). This is what we observe in both BF and EPRL models, despite the fact that the spectrum is not the same. An interesting feature of the volume operator spectrum is that there is a systematic shift between integer spins and half-integers spins. That is, these are two slightly shifted curves.
FIG. 7: Expectation values (23) of the volume operator (33). The orange line is proportional to the functional dependence between the volume of a regular tetrahedron and the area of one of its faces: \( V \propto j^{3/2} \).

The quantum spread of the volume operator is shown in Figure 8. The shift between the curves corresponding to integer spins and half-integers is manifest. With respect to the angle operator \( \Theta \) the quantum spread increases faster for BF rather than EPRL.

FIG. 8: Expectation values of the quantum spread (18) for the volume operator (33). As in Figure 3, it is evident that the spectrum of the volume operator gives rise to two distinct curves for integer and half-integers spins, which turn out to be shifted with respect to each other.

The Gaussian distributions (29) are shown in Figure 9. The average values of the volumes used as mean in the Gaussian distributions (29) are the same plotted in Figure 7.

FIG. 9: Gaussian distribution (29) of the expectation values (23) of the volume operator (33). We averaged over several runs, computing the (average) volume \( \langle V \rangle \) defined on a single node and the corresponding standard deviation \( \sigma_{\langle V \rangle} \) for each \( j \).

Volume correlations are shown in Figure 10. As for the angles, the volumes correlations between nodes belonging to the same vertex are much higher than those between different vertices. For the volumes, the latter appear 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, it 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.

\[
\rho_A = \frac{1}{Z} \text{Tr}_{\bar{A}} |\psi_0\rangle\langle \psi_0| .
\]

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

\[
S_A = -\text{Tr} (\rho_A \log \rho_A) .
\]

Using the expression in (35), after some algebraic manipulations the normalized reduced density matrix (35) can be written as:

\[
\rho_A = \frac{1}{Z} \sum_{\{i_a\}} \sum_{\{i_a'\}} M(j, \{i_a\}, \{i_a'\}) \bigotimes_a |j, \{i_a\}\rangle\langle j, \{i_a\}| ,
\]

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

\[
M(j, \{i_a\}, \{i_a'\}) = \sum_{\{i_a\}} A(j, \{i_a\}, \{i_a\}) A(j, \{i_a'\}, \{i_a\}) .
\]
We replaced the sum over the full set \{i_n\} with \{i_a\}, namely, the intertwiners involved in the partition (34). By introducing the Monte Carlo approximation \((21)\), the expression for the density matrix becomes:

\[
\rho_A \approx \frac{1}{N_{MC}} \sum_{[i_n]} \sum_{[i'_a]} \frac{A(j, \{i'_a\}, [i_n])}{A(j, [i_n])} \bigotimes_{\alpha} [j, [i_a]] \langle j, \{i'_a\} |. \tag{39}
\]

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 (13). Notice that the density matrix is symmetric and \(\text{Tr}(\rho_A) = 1\). The entropy (36) becomes:

\[
S_A \approx -\nu_i \log \nu_i , \tag{40}
\]

where \(\nu_i\) is the \(i\)-th eigenvalue of the density matrix (39). 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 (39) has dimensions \((2j + 1)^{N_A} \times (2j + 1)^{N_A}\). Furthermore, at 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 which becomes smaller and smaller. For these reasons, in the numerical computation of the density matrices we introduced a second multi-threading parallelization scheme using multiple machines. That is, 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 (23) and corresponding correlations (24).

### A. Subsystem with 1 and 2 nodes

The values of the entanglement entropy as a function of the boundary spin \(j\) are shown in Figure 11 for the subsystem \(A\) in the partition (34) consisting in 1 and 2 nodes. We used the parameters listed in Table I for the data in Figure 11. Notice that, by choosing a common value \(j\) for all the links of the star spinfoam (see Figure 15), we cannot distinguish correlations between nodes belonging to distinct vertices whether or not they are connected by the same link. For example, labelling the nodes according to the notation illustrated in Figure 15, computing the entanglement entropy for the subsystems \(A = \{C235, A235\}, A' = \{C235, D134\}\) and \(A'' = \{C235, B134\}\) we obtain the same numerical value.

![BF entanglement entropy](image1)

![EPRL entanglement entropy](image2)

**FIG. 11:** Values of the entanglement entropy (40) for different subsystems \(A\) in the partition (34) with \(N_A = 1\) and \(N_A = 2\) for the star spinfoam in Figure 15.

The value of the EPRL entropy for the subsystem consisting of a single node is similar to the value obtained in (12) for the single vertex graph. It is interesting to notice that the EPRL entropy of all considered subsystems seems to tend asymptotically to a constant value as the boundary spin \(j\) increases.
FIG. 12: Gaussian distribution (29) of the entanglement entropy (40) for the subsystems in Figure 11. We proceeded as in the case of angles (Figure 5) and volumes (Figure 9). Top panel: fluctuations for the subsystem with \( N_A = 1 \). Bottom panel: fluctuations for the subsystem with \( N_A = 2 \).

The statistical fluctuations of entropy are shown in Figure 12. Notice that, with the same number of iterations \( N_{MC} \), statistical fluctuations in Figure 12 are larger for \( N_A = 2 \) for the reasons discussed above. 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 4-simplices. In order to discuss this point, we first define the mutual information \( I(k, m) \) between two generic nodes \( k \) and \( m \) as:

\[
I(k, m) = S_k + S_m - S_{km},
\]

where \( S_{km} \) is the entropy of the subsystem \( A \) composed by the nodes \( k \) and \( m \). It turns out that the mutual information \( I(k, m) \) between \( k \) and \( m \) actually provides a bound on correlations [38, 39]:

\[
\frac{(\langle O_k, O_m \rangle - \langle O_k \rangle \langle O_m \rangle)^2}{2\|O_k\|^2\|O_m\|^2} \leq I(k, m).
\]

where \( \|O_k\| \) is the norm of the local operator \( O \) on the node \( k \). Therefore, equations (41) and (42), along with the results in Figure 11, imply that the correlation function (17) has a more stringent upper bound for the subsystem containing nodes defined on different 4-simplices.

### B. Subsystem with 4 nodes

For completeness, we show in Figure 13 the values computed for the entropy of subsystem \( A \) composed of 4 adjacent nodes, that is, 4 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 Table I, for this specific calculation we set \( N_{MC} = 10^7 \) for \( j = 0.5 \ldots 5 \), averaging over 17 independent runs both for BF and EPRL. According to the hybrid parallelization scheme discussed at the beginning of this Section, we used 17 processes, each one with 64 CPUs, for a total of 1088 CPUs. With this configuration, the total computation time for the data in Figure 13, including the sampling of the intertwiners draws, took about 3 days.
FIG. 13: Values of the entanglement entropy (40) for the subsystem $A$ in the partition $\{3,4\}$ with $N_A = 4$. All the nodes in $A$ belong to the same 4-simplex.

FIG. 14: Gaussian distribution (29) of the entanglement entropy (40) for the subsystem in Figure 13. 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 [10] with recently developed high-performance codes in LQG [11, 14] to compute the expectation value and correlation functions of operators over large spinfoam graphs in the low spins regime. We applied our methods to the computations of boundary geometrical observables, correlation functions and entanglement entropy in a spinfoam model with 20 boundary nodes, viewed as a refinement of the 4-simplex graph. We investigated both the EPRL and the BF models. Our results show that the BF and also, more importantly, the EPRL model have a well defined behavior under refinement of the boundary graph. The computed boundary geometry agrees in expectation value with the geometric interpretation of the operators. We found that correlations are present in neighboring patches but decay sharply when moving to patches that belong to different vertices, opening the way to the study of 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, while the entanglement entropy shows a significant difference.

Our work provides important hints on the well-definiteness of spinfoam refinement. Our methods can be applied to spinfoam models with Lorentzian or Euclidean signature. They can also be employed to compute bulk observables and to different choices of boundary state, although for complex coherent states one is dealing with fluctuating sampling probabilities and thus reweighting or more advanced Monte Carlo techniques are needed. The proposed method provides a needed complement to already existing numerical techniques in covariant LQG [8]. It is effective precisely in the regime of low spins quantum numbers with a large number of degrees of freedom, for which the existing methods are lacking. Currently, we are applying the algorithm presented here to study the correlations functions defined on the 4-orthoplex spinfoam model: this is the second regular dual triangulation of the 3-sphere after the 4-simplex
Appendix A: The star spinfoam amplitude

In this Appendix we provide a diagrammatic expression for the EPRL star amplitude in terms of vertex amplitudes (2), which is shown in Figure 15. The diagram for the BF is similar, with the difference that each vertex amplitude is simply given by (1). That is, with respect to the graphical amplitude in Figure 15, there are no booster functions (and therefore no Y-map).

FIG. 15: Graphical representation of the EPRL spinfoam amplitude corresponding to the star triangulation geometry described in Figure 1. 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.

The labels refer to the triangulation shown in Figure 1. 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 $\{15j\}$ symbols are labeled with the position of the node in the corresponding 4-simplex.
Appendix B: Metropolis-Hastings parameters

In this Appendix we report the parameters used in the Metropolis-Hastings algorithm, which are shown in Tables I. These are the parameters used for all calculations in this paper except for the data in Figure 13 and 2.

| $j$ | $N_{MC}$ | $\text{burnin}$ | $\sigma$ | avg. chains |
|-----|--------|--------------|--------|------------|
| 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 | $3 \times 10^6$ | $10^3$ | 0.35   | 32         |
| 3.5 | $3 \times 10^6$ | $10^3$ | 0.35   | 32         |
| 4.0 | $5 \times 10^6$ | $10^3$ | 0.35   | 32         |
| 4.5 | $5 \times 10^6$ | $10^3$ | 0.35   | 32         |
| 5.0 | $5 \times 10^6$ | $10^3$ | 0.35   | 32         |
| 5.5 | $7 \times 10^6$ | $10^3$ | 0.35   | 32         |
| 6.0 | $7 \times 10^6$ | $10^3$ | 0.35   | 32         |

| $j$ | $N_{MC}$ | $\text{burnin}$ | $\sigma$ | avg. chains |
|-----|--------|--------------|--------|------------|
| 0.5 | $10^7$ | $10^3$       | 0.40   | 31         |
| 1.0 | $3 \times 10^7$ | $10^3$ | 0.39   | 31         |
| 1.5 | $3 \times 10^7$ | $10^3$ | 0.35   | 31         |
| 2.0 | $3 \times 10^7$ | $10^3$ | 0.35   | 31         |
| 2.5 | $3 \times 10^7$ | $10^3$ | 0.35   | 31         |
| 3.0 | $5 \times 10^7$ | $10^3$ | 0.37   | 31         |
| 3.5 | $5 \times 10^7$ | $10^3$ | 0.37   | 31         |
| 4.0 | $5 \times 10^7$ | $10^3$ | 0.38   | 31         |
| 4.5 | $5 \times 10^7$ | $10^3$ | 0.39   | 31         |
| 5.0 | $5 \times 10^7$ | $10^3$ | 0.40   | 31         |
| 5.5 | $8 \times 10^7$ | $10^3$ | 0.40   | 31         |
| 6.0 | $8 \times 10^7$ | $10^3$ | 0.40   | 31         |

TABLE I: 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, “burnin” is the number of burnin iterations, $\sigma$ is the standard deviation of the Gaussian proposal density and “avg. chains” corresponds to the number of Markov chains that we averaged to improve the statistic and measure the statistical fluctuations.

A general difference that we observed between the BF and EPRL model is a greater statistical fluctuation in the expectation values of operators for EPRL as $j$ increases, despite the dimension of the intertwiners’ space being the same. In order to reduce the statistical fluctuations in EPRL, we tried both to increase the number of Markov chains to be averaged by an order of magnitude (in the code \[13\] 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 Table I.

It is interesting to notice the role that the Metropolis-Hastings parameters play in the sampling process. In particular, 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. A satisfying statistical precision is therefore achieved when these two parameters are sufficiently high and balanced. While we did not find relevant differences by modifying the number of burnin iterations, we set the optimal standard deviation of the Gaussian proposal density $\sigma$ by requiring an acceptance rate of intertwiners draws $[i_n]$ around 30% in the sampling algorithm.

Appendix C: Booster functions

The booster functions \[14\], \[40\], also known as B4 functions \[23\], are the non compact remnants of the $SL(2, C)$ integrals. These functions turn out to encode all the details of the EPRL model, such as the
We define them as\(^7\):

\[
B_4(j_f, l_f; i, k) \equiv \frac{1}{4\pi} \sum_{p_f} \left( \begin{array}{c} j_f \\ l_f \\ i, k \end{array} \right)^{(i)} \left( \int_0^{\infty} dr \sinh^2 r \prod_{f=1}^4 d^{(\gamma j_f, j_f)}_{j_f, l_f, p_f}(r) \right) \left( \begin{array}{c} l_f \\ p_f \end{array} \right)^{(k)},
\]

where \(d^{(\rho, k)}(r)\) are the boost matrix elements for \(\gamma\)-simple irreducible representations of \(SL(2, \mathbb{C})\) in the principal series and \(\gamma\) is the Immirzi parameter. In their most general formulation, the booster functions turn out to be the \(SL(2, \mathbb{C})\) analogues of the usual Clebsch-Gordan coefficients for the rotation group \(SU(2)\). The semi-classical limit of booster functions was discussed in [41]. The general explicit form of the boost matrix elements can be found in the literature [23, 42]. In the case of simple irreducible representations these turn out to be:

\[
d^{(\gamma j, j)}_{j p}(r) = (-1)^{j_i} \frac{\Gamma(j + i \gamma j + 1) \Gamma(l - i \gamma j + 1)}{\Gamma(j + i \gamma j + 1) \Gamma(l - i \gamma j + 1)} \sqrt{2j + 1} \sqrt{2l + 1} \frac{2^{2j + 1} \Gamma(2j)! \Gamma(l - j)!(l + p)!(l - p)!}{(j + l + 1)!} \frac{(2j)!(l + j)!(l - j)!(l + p)!(l - p)!}{(j + p)!}(j - p)!^{1/2} \times e^{-(j - i \gamma j + p + 1)r} \sum_s \frac{(-1)^s e^{-2sr}}{s!(l - j - s)!} 2F_1[l + 1 - i \gamma j, j + p + 1 + s, j + l + 2, 1 - e^{-2r}].
\]

where \(2F_1[a, b, c, z]\) is the hypergeometric function.

\(^7\) In this Appendix we don’t indicate the dependence on multiple variables with the curly brackets in order not to weigh down the notation.
Acknowledgments  We thank to Carlo Rovelli for many discussions on this project. We acknowledge the Shared Hierarchical Academic Research Computing Network (SHARCNET) for granting access to their high-performance computing resources. We thank in particular 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 acknowledge support also from the QISS JFT grant 61466. FV’s research is supported by the Canada Research Chairs Program. We acknowledge the Anishinaabek, Haudenosaunee, Lùnaapéewak and Attawandaron peoples, on whose traditional lands Western University is located.

***

[1] P. Donà, F. Gozzini, and G. Sarno, “Numerical analysis of spin foam dynamics and the flatness problem,” Physical Review D 102 no. 10, (Nov, 2020). https://doi.org/10.1103/physrevd.102.106008
[2] P. Donà, M. Fanizza, G. Sarno, and S. Speziale, “Numerical study of the Lorentzian Engle-Pereira-Rovelli-Livine spin foam amplitude,” Physical Review D 100 no. 10, (Mar, 2019), arXiv:1903.12624
[3] P. Frisoni, F. Gozzini, and F. Vidotto, “Numerical analysis of the self-energy in covariant loop quantum gravity,” Phys. Rev. D 105 (May, 2022) 106018. https://link.aps.org/doi/10.1103/PhysRevD.105.106018
[4] P. Donà and P. Frisoni. “How-to Compute EPRL Spin Foam Amplitudes,” Universe 8 no. 4, (Mar, 2022) 208 https://doi.org/10.3390/universe8040208
[5] S. K. Asante, B. Dittrich, and H. M. Haggard, “Effective Spin Foam Models for Four-Dimensional Quantum Gravity,” Physical Review Letters 125 no. 23, (Dec, 2020) https://doi.org/10.1103/PhysRevLett.125.231301
[6] S. K. Asante, B. Dittrich, and J. Padua-Aranguel, “Effective spin foam models for Lorentzian quantum gravity,” Classical and Quantum Gravity 38 no. 19, (Sep, 2021) 195002 https://doi.org/10.1088%2F1361-6382%2Fac1b44
[7] B. Bahr and S. Steinhaus, “Hypercuboidal renormalization in spin foam quantum gravity,” Phys. Rev. D 95 (Jun, 2017) 126006 https://link.aps.org/doi/10.1103/PhysRevD.95.126006
[8] M. Han, Z. Huang, H. Liu, D. Qu, and Y. Wan, “Spinfoam on a Lefschetz thimble: Markov chain Monte Carlo computation of a Lorentzian spinfoam propagator,” Physical Review D 103 no. 8, (Apr, 2021) http://dx.doi.org/10.1103/PhysRevD.103.084026
[9] M. Han, Z. Huang, H. Liu, and D. Qu, “Complex critical points and curved geometries in four-dimensional Lorentzian spinfoam quantum gravity,” 2021.
[10] W. K. Hastings, “Monte Carlo Sampling Methods Using Markov Chains and Their Applications,” Biometrika 57 no. 1, (1970) 97–109. http://www.jstor.org/stable/2334940
[11] F. Gozzini, “A high-performance code for EPRL spin foam amplitudes,” arXiv:2107.13952 [gr-qc]
[12] F. Gozzini and F. Vidotto, “Primordial fluctuations from quantum gravity,” Frontiers in Astronomy and Space Sciences 7 (Feb, 2021) https://doi.org/10.3389/fspas.2020.629466
[13] https://github.com/PietropaoloFrisoni/Star-spinfoam-model 2021.
[14] P. Donà and G. Sarno, “Numerical methods for EPRL spin foam transition amplitudes and Lorentzian recoupling theory,” General Relativity and Gravitation 50 no. 10, (2018)
[15] T. G. et al., “GNU Multiple Precision Arithmetic Library 4.1.2,” December, 2002. http://swox.com/gmp/
[16] L. F. et al., “MPFR: A Multiple-Precision Binary Floating-Point Library With Correct Rounding”. RR5753, INRIA. 2005, pp.15. ffinria-00070266f.
[17] A. Enge, M. Gastineau, P. Théveny, and P. Zimmermann, mpc — A library for multiprecision complex arithmetic with exact rounding. INRIA, 1.1.0 ed., Jan., 2018. http://mpc.multiprecision.org/
[18] A. P. Yutsis, I. B. Levinson, and V. V. Vanagas, Mathematical Apparatus of the Theory of Angular Momentum. Israel Program for Scientific Translation, Jerusalem, Israel, 1962.
[19] H. T. Johansson and C. Forsén, “Fast and Accurate Evaluation of Wigner 3j, 6j, and 9j Symbols Using Prime Factorization and Multidigit Integer Arithmetic,” SIAM Journal on Scientific Computing 38 no. 1, (Jan, 2016) A370-A384 http://dx.doi.org/10.1137/15M1021908
[20] J. Rasch and A. C. H. Yu, “Efficient Storage Scheme for Precalculated Wigner 3j, 6j and Gaunt Coefficients,” SIAM Journal on Scientific Computing 25 no. 4, (2004) 1416–1428 https://doi.org/10.1137/S1064827503422932
[21] J. Engle, E. Livine, R. Pereira, and C. Rovelli, “LQG vertex with finite Immirzi parameter,” Nucl. Phys. B799 (2008) 136–149, arXiv:0711.0146
[22] J. Engle, R. Pereira, and C. Rovelli, “The loop-quantum-gravity vertex-amplitude,” Phys. Rev. Lett. 99 (2007) 161301. arXiv:0705.2388

[23] S. Speziale, “Boosting Wigner’s nj-symbols,” Journal of Mathematical Physics 58 no. 3, (Sep, 2017) arXiv:1609.01632 http://arxiv.org/abs/1609.01632 http://dx.doi.org/10.1063/1.4977752

[24] P. Donà, P. Frisoni, and E. Wilson-Ewing, “Radiative corrections to the Lorentzian EPRL spin foam propagator,” 2022. https://arxiv.org/abs/2206.14755

[25] P. Frisoni, “Studying the EPRL spinfoam self-energy,” 2021. https://arxiv.org/abs/2112.08528

[26] J. Engle and R. Pereira, “Regularization and finiteness of the Lorentzian LQG vertices,” Phys. Rev. D79 (2009) 84034. arXiv:0805.4696

[27] L. Freidel and S. Speziale, “Twisted geometries: A geometric parametrisation of SU(2) phase space,” Phys.Rev. D82 (2010) 84040. arXiv:1001.2748

[28] E. Bianchi, L. Modesto, C. Rovelli, and S. Speziale, “Graviton propagator in loop quantum gravity,” Class. Quant. Grav. 23 (2006) 6989–7028. arXiv:0604044 [gr-qc]

[29] E. R. Livine and S. Speziale, “Group Integral Techniques for the Spinfoam Graviton Propagator,” JHEP 11 (2006) 92. arXiv:0608131 [gr-qc]

[30] E. Alesci, E. Bianchi, and C. Rovelli, “[LQG] propagator: {III}. {T}he new vertex,” Class. Quant. Grav. 26 (2009) 215001. arXiv:0812.5018

[31] E. Bianchi, E. Magliaro, and C. Perini, “LQG propagator from the new spin foams,” Nucl. Phys. B822 (2009) 245–269. arXiv:0905.4082

[32] S. Speziale, “Studying the EPRL spinfoam self-energy,” 2021. https://arxiv.org/abs/2112.08528

[33] E. Bianchi and Y. Ding, “Lorentzian spin foam propagator,” Phys.Rev. D86 (2012) 104040 arXiv:1109.6538

[34] T. Besard, C. Foket, and B. De Sutter, “Effective Extensible Programming: Unleashing Julia on GPUs,” IEEE Transactions on Parallel and Distributed Systems 30 no. 4, (Apr, 2019) 827–841 http://dx.doi.org/10.1109/TPDS.2018.2872064

[35] M. Fishman, S. R. White, and E. M. Stoudenmire, “The ITensor Software Library for Tensor Network Calculations,” 2020.

[36] F. Goggin, “A high-performance code for EPRL spin foam amplitudes,” Oct, 2021. http://dx.doi.org/10.1088/1361-6382/ac2b0b

[37] C. Rovelli and L. Smolin, “Discreteness of area and volume in quantum gravity,” Nucl. Phys. B442 (1995) 593–622. arXiv:9411005 [gr-qc]

[38] C. Rovelli and F. Vidotto, Covariant Loop Quantum Gravity. Cambridge University Press, Cambridge, 2015.

[39] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press, 2010.

[40] E. Bianchi, P. Donà, and I. Vilensky, “Entanglement entropy of Bell-network states in loop quantum gravity: Analytical and numerical results,” Physical Review D 99 no. 8, (Apr, 2019) http://dx.doi.org/10.1103/PhysRevD.99.086013

[41] P. Donà, M. Fanizza, G. Sarno, and S. Speziale, “Numerical study of the Lorentzian Engle-Pereira-Rovelli-Livine spin foam amplitude,” Physical Review D 100 (2019) arXiv version:1903.12624 [gr-qc].

[42] W. Ruhl, The Lorentz group and harmonic analysis. W.A. Benjamin, Inc, New York, 1970.