NAWI Graz
Natural Sciences

UNIVERSITÄT GRAZ
UNIVERSITY OF GRAZ
Institute of Physics

Bachelor's Thesis

Extraction of the metric within the framework of Causal Dynamical Triangulation

Lukas Walter Würger

12103741

in partial fulfillment of the requirements for the degree of

Bachelor - BSc

supervised by Univ.-Prof. Dipl.-Phys. Dr.rer.nat. Axel Torsten Maas

September 24, 2024

Contents

Abstract

This thesis tries to resolve the issue regarding the overdetermination of events that was previously encountered in the Bachelor's thesis of Christina Perner titled "Metric reconstruction with causal dynamical triangulation" [1]. Thus, the aim of this thesis is to properly answer the question: "Is it possible to extract the metric tensor in two-dimensional spacetime via Causal Dynamical Triangulation in the presence of singularities if only the structure and the distances between the events are known?". Regarding the feasibility, there may be a few approaches to tackle this exact problem, but in the course of this work two specific approaches will be covered. First, a new set of events called "support events" will be introduced and their distance to the closest spacelike connected regular events in the grid is going to zero in the limit, therefore overlapping with other events in the limit, with the idea of working around occurring singularities. The second approach considers a dual grid, which results in new events and a new structure together with changed distances. If the metric tensor can be extracted for every event in the dual grid, it would resolve the issue of an emerging singularity in the original grid altogether.

1 Introduction

Since the last century quantum mechanics and general relativity are two core components of our understanding of the universe and modern physics. Quantum mechanics describes the microscopic world on the scale of atoms and subatomic particles, giving rise to a theory that characterizes the nature of particles and waves based on statistic and probability. Contrary to the scale of quantum mechanics, general relativity focuses on heavy, large scale objects such as planets and stars, describing the spacetime in the presence of mass. So far no formalism has been found to truly unify the gravitational force, one of the four fundamental forces, with the concept of a quantum field theory. All of the various approaches that have been developed in trying to do so are collected under the framework of a theory referred to as quantum gravity. However, such a unification has been well established for the other three fundamental forces: the weak force, the strong force and the electromagnetic force. Many questions arise considering as to why the gravitational force is the only fundamental force that does not yet fit in the greater scheme of a quantum mechanical unification.

One promising attempt at quantum gravity is the concept of Causal Dynamical Triangulation with the idea of cutting the spacetime into piece-wise flat equilateral simplices to describe Lorentzian spacetime. For the last years this theory emerged from the underlying idea of Dynamical Triangulation and so far great progress has been achieved in both two dimensions and four dimensions. Causal Dynamical Triangulation is the underlying concept regarding the method used in this thesis. [2]

It should be noted that this bachelor's thesis only discusses two-dimensional spacetime, since it offers an interesting playground to test concepts and theories before extending the work into higher dimensions. This thesis works in a region where the considered structure is technically still classical.

The thesis starts with a theoretical background, thereby covering the basics needed to understand the thesis and additionally providing further information as some sort of motivation to deepen the knowledge regarding the topic. After the theoretical background, the utilized gauge and path integral will be covered as well as the basic equations involving the metric in a two-dimensional grid. Afterwards, a demonstration of the formalism serves to verify the functionality and show the suitability of the chosen approach. The main part of the thesis demonstrates the concrete procedure how the two candidate theories to resolve the issue are approached. The core part of the thesis finishes off with a discussion and conclusion, followed by a declaration, the literature and an appendix including the utilized codes written in "Wolfram Mathematica".

2 Theoretical background

This section serves to give a proper introduction to the topic and the underlying theory. Since quantum gravity involves both the theory of general relativity and a quantum field theory, firstly a brief insight into general relativity will be given. Afterwards the necessary quantum theory is introduced, a short insight in Regge calculus is given and in the end the approach of Causal Dynamical Triangulation is portrayed.

For the sake of completeness, further aspects of the theory shall be mentioned nonetheless. Many properties need to be considered if the manifold is raised from two dimensions to four dimensions by extension of the results of this work. It further serves a better purpose of introducing the theory and define some major terms together with the associated quantities.

2.1 Classical theory

General relativity is a well established theory in its own right and has been proven to be accurate on a large scale. Since the most common way to introduce general relativity is to start with special relativity and extend the concepts behind it, it will be done the same way in this theory section[1]. Within the theory of general relativity, the metric tensor and the curvature of spacetime are a broadly understood concept. The basics of the metric tensor [2] as well as the concept of causality can be obtained from the framework of special relativity in Minkowski space as a special case of a four-dimensional manifold. The Minkowski space describes flat spacetime in the absence of mass, but serves nonetheless to illustrate the underlying idea as well as to define major terms, which will be used regularly throughout the thesis. [3–6]

2.1.1 Special relativity and the metric tensor

In the case of four-dimensional spacetime, three dimensions represent the spatial nature and one dimension is given along a time axis. A worldline refers to the path that a particle takes through spacetime. In this case, the characterization as a curve is the most fitting. This concept can be understood as a parametrized one-dimensional set of events. At every event a lightcone can be defined, creating the collection of every possible path a particle can take, in order to still obey causality. Obeying causality means that the particle is not able to travel faster than light and therefore every possible future and past of the particle lies inside a cone that is constructed based on the worldline a photon would create. A worldline that coincides with the path of the photon is given by the edges of the cone, meaning that the particle itself would be moving at the speed of light, called lightlike or null separated from a point p. The entirety of all points that lie inside the future and past light cone of a point p are referred to as timelike separated from p. Every point that lies outside of the light cone is called spacelike separated from p. Therefore, it is impossible for an event α to be the direct cause of an event β if they are spacelike separated. The future and past lightcones of given events may intercept at some point, but they are not causally related to each other. Incidentally, it is impossible for two timelike separated points to happen exactly simultaneous. The spacetime interval between two events will be defined as $(\Delta s)^2 = -(c\Delta t)^2 + (\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2$, which can be written as $(\Delta s)^2 = \eta_{\mu\nu}\Delta x^\mu \Delta x^\nu$ with the Minkowski metric $\eta_{\mu\nu}$ given in Eq. (1). [3]

$$\eta_{\mu\nu} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

[1] This theory section will be based on the four-dimensional case since in the literature it is common to include three spatial dimensions.

[2] In special relativity the metric tensor is written as $\eta_{\mu\nu}$, denoting the Minkowski metric, whereas in the case of general relativity, $g_{\mu\nu}$ is used as a general expression for the metric tensor.

The interval between two points is per definition negative if the separation is of timelike nature. The occurence of a positive $(\Delta s)^2$ implies that two points are spacelike separated. In the case of the interval being zero, the points are lightlike separated. It is inconvenient for the interval to be negative in case of a timelike connection. To deal with this, the proper time τ is introduced that satisfies $(\Delta\tau)^2 = -(\Delta s)^2 = -\eta_{\mu\nu}\Delta x^\mu \Delta x^\nu$, where the integral is taken over the path. The proper time between two events is an important quantity since it gives a measurement for the elapsed time that would be perceived by an observer that moves between events by maintaining on a straight line. It should not be overlooked that the proper time and coordinate time are not the same in case of more general trajectories. The definition of the interval between events is summarized in Eq. (2) with Δx being representative for any vector. [3]

$$\text{if } \eta_{\mu\nu}\Delta x^\mu \Delta x^\nu \text{ is } \begin{cases} < 0, \ (\Delta s)^2 \text{ is timelike} \\ = 0, \ (\Delta s)^2 \text{ is lightlike or null} \\ > 0, \ (\Delta s)^2 \text{ is spacelike} \end{cases} \tag{2}$$

The important concept of covariant and contravariant vectors is best understood by considering vector spaces. Every properly established vector space is in possession of a dual vector space that does not differ in dimension. For example, the tangent space T_p has a well-defined dual vector space that is called the cotangent space and written as T_p^*. The concept of the dual space entails the total of every single linear map from the original vector space to the real numbers. Vectors that are referred to as contravariant vectors are elements of T_p, denoted with upper indices x^μ. Covariant vectors are contained in T_p^* and are written via the usage of lowered indices x_μ. A change of basis in the tangent space is caused by changing coordinates ($x^\mu \to x^{\mu'}$). [3]

One rather useful application of the metric tensor is to convert a contravariant vector to a covariant vector and vice versa. Thus, the metric serves as a tool in order to raise or lower vectors indices. A demonstration thereof can be seen in Eq. (3). [3]

$$\begin{aligned} \eta^{\mu\nu} x_\nu &= x^\mu \\ \eta_{\mu\nu} x^\nu &= x_\mu \end{aligned} \tag{3}$$

If the metric indices are written superscript ($\eta^{\mu\nu}$), the metric is called contravariant, in consistency with the nomenclature of vectors, and classified as a tensor of type (2, 0). Subscript indices ($\eta_{\mu\nu}$) indicate that the metric is covariant and a tensor of type (0, 2). In either case, the summation convention is used, signifying that summation occurs over all indices. [3, 6]

The so-called signature is defined as the number of positive and negative eigenvalues. The Minkowski metric $\eta_{\mu\nu}$, for example, is a metric in possession of a signature -+++. If at least one eigenvalue equals zero, the inverse metric is nonexistent due to being degenerate. In the case of general relativity only continuous, non-degenerate metrics will be encountered since the metric tensor needs to be invertible. Depending on the signs of the signature, the metric will be classified further. A Euclidean, Riemannian or just positive definite metric has only positive signs. A Lorentzian or pseudo-Riemannian metric includes a single minus sign in its signature. The signature of an indefinite metric has some +1's and some -1's. In the case of general relativity the spacetime of interest is of Lorentzian nature. [3]

2.1.2 Extension to general relativity

The expression η was used specifically for the Minkowski metric in Minkowski space. In the sense of not making a restriction to individual manifolds, from this point onward, the metric is denoted as g to indicate universal usage.

Space and time can be thought of as a continuum consisting of events called spacetime. A more sophisticated definition of spacetime would be that of a four-dimensional manifold. Each event represents a point in space at a given time and can be uniquely described by the three spatial positions and the value of time. [4]

A manifold can be understood as a space that looks like $\mathbb{R}^n$ in local regions, but may have quite different global properties. In this case "looks like" does not imply that the metric is the same, but rather that only functions and coordinates behave in a similar way. By smoothly sewing together the regions that look like open subsets of $\mathbb{R}^n$, the whole manifold is obtained. Every single subset of the manifold needs to be of the same dimension n. In doing so a manifold of dimension n is constructed, in which it is possible to locally convert functions into Euclidean space in order to analyze them. Since manifolds are a broad concept, there are many examples for manifolds like $\mathbb{R}^n$ itself, the n-sphere S^n and the n-torus T^n. However, there are still structures that are per definition not a manifold. One example of such structures includes two cones stuck together at their vertices. In this sense, a one-dimensional line running into a two-dimensional plane is also not a proper manifold. The direct product of two manifolds results in yet another manifold. [3, 4]

In general relativity, the concept of parallel transport plays an important role. It is understood as keeping a vector constant during the movement along a given path. In order for the parallel transport to be well-defined, a connection is of essence. It should be mentioned that if a vector in flat space is being manipulated, the Christoffel connection on this space is implicitly taken into account. The chosen path between two events affects the outcome of parallel transporting a vector from one event to the other in the case of curved spacetime. [3]

One way to treat tensors is like a map from vectors and dual vectors to the real numbers. The covariant derivative can be treated as a quantity to measure the rate of change of tensors. Directional derivatives along curves while keeping all of the other coordinates constant is achieved via partial derivatives. In addition, the partial derivative ∂ determines the covariant derivative (∇_μ) for each direction μ. [3]

It is necessary to be able to relate vectors in the tangent spaces of nearby points. This can be achieved by the definition of a so-called connection. The Christoffel symbol is an object that encompasses a unique connection, which is defined in terms of the metric tensor. The concrete definition in mathematical terms is given according to Eq. (4). [3]

$$\Gamma^\lambda_{\mu\nu} = \frac{1}{2} g^{\lambda\sigma} (\partial_\mu g_{\nu\sigma} + \partial_\nu g_{\sigma\mu} - \partial_\sigma g_{\mu\nu}) \tag{4}$$

The Christoffel symbol looks like it would be a tensor in terms of the notion, but this is not the case. To take a covariant derivative ∇_μ can be seen as the fundamental use of a connection. In

the case of a vector field V^ν, the covariant derivative is given according to Eq. (5). [3]

$$\nabla_\mu V^\nu = \partial_\mu V^\nu + \Gamma^\nu_{\mu\sigma} V^\sigma \tag{5}$$

In general relativity, the geodesic equation is given by Eq. (6). In order for a parametrized curve $x^\mu(\lambda)$ to be classified as a geodesic, the curve must obey this equation. For the sake of definition, the term geodesic shall be properly explained. In flat Euclidean space, the path of shortest distance between any two given points is given by a straight line. For curved spaces, the geodesic is understood as the generalization of this exact notion, therefore describing the path of shortest distance under the consideration of curved spacetime. A somewhat more sophisticated definition of a straight line encompasses the formulation of a path that applies a parallel-transport on its own tangent vector. The only case in which these both concepts coincide is if the connection at hand is a Christoffel connection. [3]

$$\frac{d^2x^\mu}{d\lambda^2} + \Gamma^\mu_{\rho\sigma}\frac{dx^\rho}{d\lambda}\frac{dx^\sigma}{d\lambda} = 0 \tag{6}$$

The deviation of one geodesic from another nearby geodesic can be seen as the way in which the curvature of spacetime manifests itself. Gravitation can be seen as a manifestation of this curvature [5]. The intrinsic geometry of a space is measurable by observers that are confined to the manifold. If it is of interest in which way a space is embedded in a larger space, the extrinsic geometry of the manifold needs to be considered. The intrinsic geometry of spacetime is not reliant on any embeddings and the Riemann curvature can be seen as one property of the intrinsic geometry. If it becomes necessary to describe submanifolds of spacetime, the extrinsic curvature is occasionally beneficial. [3]

A fundamental quantity in general relativity is given by the Riemann tensor [3]. The Riemann tensor is bound to vanish should there be a coordinate system with constant components of the metric. In other terms, in the case of a vanishing Riemann tensor, it is always possible to construct a metric in which the components are constant. The mathematical definition is shown in Eq. (7) with mention to the Christoffel symbols. [3]

$$R^\rho{}_{\sigma\mu\nu} = \partial_\mu \Gamma^\rho_{\nu\sigma} - \partial_\nu \Gamma^\rho_{\mu\sigma} + \Gamma^\rho_{\mu\lambda}\Gamma^\lambda_{\nu\sigma} - \Gamma^\rho_{\nu\lambda}\Gamma^\lambda_{\mu\sigma} \tag{7}$$

A closely related tensor of the Riemann tensor is the Ricci tensor, which inherits the information regarding traces of the Riemann tensor entirely. A new quantity called the Weyl tensor will be introduced that incorporates the trace-free parts of the Ricci tensor. Thus, the Weyl tensor may be understood as the Riemann tensor after the elimination of every single contraction. One crucial property of the mentioned Weyl tensor is that it shows invariance under conformal transformations. In addition, the Weyl tensor satisfies the symmetry properties of the Riemann tensor. The trace of the Ricci tensor is another meaningful property, denoted as the Ricci scalar R. The formulation of the Ricci tensor, the Ricci scalar and the Weyl tensor is demonstrated in Eq. (8), Eq. (9) and Eq. (10). In Eq. (10) the dimension is denoted as n. [3,4]

$$R_{\mu\nu} = R^\lambda{}_{\mu\lambda\nu} \tag{8}$$

[3] The Riemann tensor is often referred to as the curvature tensor. [3]

$$R = R^{\mu}{}_{\mu} = g^{\mu\nu} R_{\mu\nu} \tag{9}$$

$$C_{\rho\sigma\mu\nu} = R_{\rho\sigma\mu\nu} - \frac{2}{(n-2)}(g_{\rho[\mu}R_{\nu]\sigma} - g_{\sigma[\mu}R_{\nu]\rho}) + \frac{2}{(n-1)(n-2)} g_{\rho[\mu}g_{\nu]\sigma}R \tag{10}$$

One characteristic of the Riemann tensor is that it obeys the Bianchi identity as a differential symmetry, which is given in Eq. (11). [3]

$$\nabla_{[\lambda} R_{\rho\sigma]\mu\nu} = 0 \tag{11}$$

Another description of the Ricci tensor is the trace over the second and fourth (or the first and third) indices of the Riemann tensor. The trace of the Riemann tensor over its first and second (or third and fourth) indices vanishes due to its antisymmetry properties. The symmetry of the Ricci tensor in connection with the Christoffel connection ($R_{\mu\nu} = R_{\nu\mu}$) is achieved automatically. [3,4]

The Einstein tensor, which is defined in Eq. (12), is another quite important quantity which always obeys $\nabla^{\mu} G_{\mu\nu} = 0$. The Einstein tensor appears in Einstein's equation for general relativity given in Eq. (13) and ensures consistency. In four dimensions the description of a trace-reversed version of the Ricci tensor is a fitting view of the Einstein tensor. Consequently, the symmetry of the Ricci tensor and the metric causes the Einstein tensor to have a symmetric characteristic as well. [3,4]

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} \tag{12}$$

The Einstein equation in Eq. (13) describes in which way the presence of energy-momentum affects the curvature of spacetime and further resolves the conflict between the Bianchi identity and local conservation of energy, which stood in contradiction until Einstein' equation was introduced. Here G shall not be thought of as the trace of the Einstein tensor since G symbolizes Newton's constant of gravitation. The energy-momentum-tensor is denoted $T_{\mu\nu}$. [3]

$$R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} = 8\pi G T_{\mu\nu} \tag{13}$$

2.2 Quantum theory

The formulation of a quantum theory in the case of general relativity is quite non-trivial. If an approach of a perturbative quantization of general relativity is made, it is bound to cause problems since non-renormalizable divergences will arise [7]. In order to proceed, several assumptions need to be made. Henceforth, it shall be assumed that a path integral approach, such as given in Eq. (14), works. [8,9]

$$Z = \int D g_{\mu\nu} D\phi \, e^{iS(g_{\mu\nu},\phi)} \tag{14}$$

Herein ϕ is considered as a placeholder for all matter fields since the structure of Eq. (14) would remain the same even if different or additional matter fields are included. One above mentioned assumption includes that the metric is the suitable quantum degree of freedom. Furthermore, it is a necessity that the integral is taken over a full, non-compact GL(4,$\mathbb{R}$) group, thereby covering every possible Lorentzian metric. In particular, another assumption includes that diffeomorphism orbits are all of the same size. In avoidance of potential obstructions, the measure should be an applicable Haar measure [4], without the need of including additional terms to the action. In the course of this thesis the leading order will not be exceeded, in which case these assumptions shall suffice. [8, 9]

In order to be considered a physical observable, an observable must be diffeomorphism-invariant and further obey invariance under gauge transformations. In this sense, the metric is given a nonphysical property. A fitting and quite simple choice for a physical observable in this regard is offered by considering scalar operators. Two intriguing examples are given in Eq. (15), which illustrates a scalar operator describing the physical Higgs particle, and Eq. (16) as a demonstration of a scalar operator depicting local curvature. Both operators are coordinate-independent and thus depend only on the event x. [8, 11]

$$O_1(x) = \phi^\dagger(x)\phi(x) \tag{15}$$

$$O_2(x) = R(x) \tag{16}$$

The propagator characterizes the fundamental quantity that describes a particle and can be defined in accordance to Eq. (17). A dependence on two events will be required for any resemblance to physical propagators. This reliance on events, like x and y in this case, instead of coordinates causes the propagator to be diffeomorphism-invariant since the propagator itself is a function of two events instead of distance. [8]

$$D(x,y) = \langle O(y)O(x)\rangle \tag{17}$$

Spacetime in quantum gravity is estimated to be isotropic on average. In terms of characterizing the relation between both events, one quantity that still obeys diffeomorphism-invariance will be needed. This in itself is a challenging task. Justification becomes natural once it is considered that the distance between two events is dependent on the metric. This leads to the case that the distance is given a dynamical aspect and is consequently bound to become an expectation value. A unique geodesic connecting two events can be found for each configuration. Mentioned geodesic length is depicted in Eq. (18) where r is the expectation value for an invariant length connecting the two events symbolized as x and y. The path itself is denoted $z(t)$ and g corresponds to the metric. [8]

$$r(x,y) = \left\langle \min_{z(t)} \int_x^y g^{\mu\nu} \frac{\mathrm{d}z_\mu(t)}{\mathrm{d}t} \frac{\mathrm{d}z_\nu(t)}{\mathrm{d}t}\, dt \right\rangle \tag{18}$$

[4] For further information regarding the Haar measure, see [10].

To give rise to a physical object obeying invariance under both local and spacetime gauge symmetries, the propagator in Eq. (17) is regarded as a function of the expectation value of the geodesic distance given in Eq. (18), $D(r(x,y))$. [8]

2.3 Regge calculus

Regge calculus, named after Tullio Regge, dates back to 1961, where Regge introduced the idea of taking advantage of the triangulation of surfaces in order to achieve a proper description of curvature, and therefore, gravitation, due to discrete calculus[5] [13]. In other terms, Einstein's continuum theory is approached via the suggestion of a simplicial discretization of the gravitational action and the metric spacetime manifold [14]. A core concept of Regge calculus is to construct the smooth spacetime manifold without the need for the usage of coordinates and only relying on basic concepts of topology. [13]

In Regge Calculus, the fundamental building-block of a spacetime manifold (or fundamental element of geometry) is called a simplex. Thus, spacetime is composed of flat-space simplexes that are joined face to face, edge to edge as well as vertex to vertex [5, 13]. In four-dimensional spacetime the local building blocks are four-simplices, in three dimensions they are tetrahedrons and in the case of two dimensional-spacetime triangles are utilized. [5, 14].

The first step in the triangulation process is to specify the base simplex. It is advantageous to select simplexes with closest resemblance to regular simplexes of equal edges. It should be noted that some degree of freedom needs to be accommodated in order to fit the curved surface, which makes the endeavor of maintaining all the sides of equal length an impossible task. The magnitude of the surface's curvature dictates how many simplexes need to be taken to cover it. Naturally, the number of simplexes scales with the degree of the surface's curvature. The discretization process leads to a more accurate approximation if a greater number of simplexes per unity area is present. Between the edges of the adjacent triangles, no curvature occurs since it is measured at the vertexes [13]. In this sense, Regge calculus serves as a method capable of analyzing physical situations absent of spherical symmetry or lacking in symmetry altogether. [5]

What Regge calculus excels at is the depiction of a spacetime manifold absent of coordinates and their associated gauge freedom [2]. In case of a numerical treatment of curvature, Regge calculus serves as a fundamental theoretical background, therefore substantiating its value as a concept. Though Regge calculus exhibits some limitations, it is considered to be a candidate for an appropriate approach to quantize gravitation. [13]

Four-dimensional Regge calculus is compatible with a path-integral quantization. The discretization of the Einstein action is well understood in classical terms. In this sense, it serves as a solid basis for quantum Regge calculus. In quantum Regge calculus, it is possible to recover diffeomorphism invariance in a weak-field perturbation about flat space. In the sense of a quantum theory, quantum Regge calculus is viewed to possess no gauge invariance. It is argued that various edge length assignments are in correspondence to geometries that are inequivalent. This matter is taken aside from configurations possessing special symmetries.[6]. [14]

[5] For the original paper written by Tullio Regge, see [12].

[6] Regarding the gauge invariance, individual opinions differ and a certain discrepancy is given. The view that quantum Regge calculus possess no gauge invariance is supported in [15] and [16]. A dissenting view is given in [17].

2.4 Causal Dynamical Triangulation

Due to gravity being perturbatively non-renormalizable, an inconvenient obstacle arises by trying to unite gravity with a quantum theory [18]. Thus, an approach to quantize gravity non-perturbatively is considered one way of working around the issue. In this sense, Causal Dynamical Triangulation [7], which can be viewed as a modern formulation of lattice gravity, is a suitable choice. The underlying concept behind CDT is the construction of a nonperturbative quantum field theory of gravity. This theory is received from a scaling limit of the lattice-regularized theory that is expressed as a sum taken over spacetime histories. [2, 19]

In the grand scheme of things, the goal of CDT is a proper approximation of the curved spacetimes of classical general relativity via a regularized, nonperturbative path integral over geometric lattice structures. Piecewise flat spaces serve as a way to introduce causal, Lorentzian features into the gravitational path integrals [2]. Including these causal, Lorentzian properties is an important requirement to ensure that the associated quantum gravity theory even possesses a classical limit. In this case, CDT provides concrete evidence to confirm this statement. [20]

In CDT the non-perturbative gravitational path integral Z^{CDT} takes the form of Eq. (19).

$$Z^{\mathrm{CDT}}(G_{\mathrm{N}},\Lambda) = \lim_{a\to 0}\sum_{T\in\mathscr{T}}\frac{1}{C(T)}e^{iS^{\mathrm{CDT}}[T]} \tag{19}$$

In Eq. (19) G_{N} stands for Newton's constant of gravitation, Λ denotes the cosmological constant and T represents inequivalent Lorentzian triangulations that are summed over and assembled from two types of elementary four-simplices [2]. S^{CDT} is only a bare lattice action with the so-called Regge version of the Einstein-Hilbert action serving as the default choice [2, 21]. In this case the choice is neither fundamental nor exact similarly to Regge calculus possessing no fundamental physical status in four dimensions. The amount of elements in the automorphism group of T is stated as $C(T)$. If T is deprived of any symmetry, $C(T)$ equals 1. [2]

The theory of CDT predicts a positive, renormalized, cosmological constant, indicating that the spacetime is a de Sitter space [2, 3, 5]. Only the property of being positive is predicted since CDT is unable to provide a particular value. This is due to the convergence requirement of the non-perturbative path integral as well as the renormalization behaviour the cosmological constant exhibits in the continuum limit. In nonperturbative quantum gravity, it is no common occurrence to reproduce several aspects of the classical theory, which further corroborates that CDT is a promising candidate for a theory of quantum gravity. [2]

A comfortable feature of the two-dimensional model of CDT is its exact solvability. Furthermore, it is simple to couple matter to CDT quantum gravity, regardless of CDT is considered in two or higher dimensions. [19]

A variation of CDT is given in terms of Locally Causal Dynamicl Triangulation or LCDT [8] abbreviated. In CDT only triangulations using two timelike connections are considered. LCDT gives a more general vertex structure since triangles with two timelike connections as well as

[7] Causal Dynamical Triangulation shall from now on be denoted CDT in short.

[8] LCDT is also referred to as "CDT without preferred foliation". [20]

triangles with two spacelike connections are allowed and possesses therefore no preferred slicing. [20]

In summation, the lattice topology is the key difference between CDT quantum gravity and classical as well as quantum Regge calculus. In CDT variable lattice topology and fixed edge lengths are of essence whereas fixed lattices and variable edge lengths are dealt with in Regge calculus, subject to triangle inequalities. [2]

3 Required properties of the metric in a two-dimensional grid

A suitable gauge has to be chosen to receive a representative solution of the metric in every event. The choice of the gauge condition is not unique, but in the course of this thesis the Haywood gauge will be utilized. The equation shown below, Eq. (20), represents this gauge, which includes the elements of the metric $g^{\mu\nu}$ as well as the partial derivative of the inverse metric $\partial_\nu g_{\mu\rho}$. [1]

$$\sum_{\mu,\nu} g^{\mu\nu}\partial_\nu g_{\mu\rho} = 0 \tag{20}$$

The vectors that occur during this thesis are of linear nature, which is the reason the derivative of the metric can be handled according to Eq. (21) and Eq. (22). The current position in the grid where the calculation of the derivative has to be performed is denoted as $\vec{A}$. In this case Eq. (21) represents the derivative with respect to the first coordinate whereas Eq. (22) needs to be used to deal with the derivative regarding the second coordinate. [1]

$$\partial_1 g(\vec{A}) = g\begin{pmatrix} A_1+1 \\ A_2 \end{pmatrix} - g(\vec{A}) \tag{21}$$

$$\partial_2 g(\vec{A}) = g\begin{pmatrix} A_1 \\ A_2+1 \end{pmatrix} - g(\vec{A}) \tag{22}$$

The derivative may not only be defined forward. In the course of this work it is also necessary to make use of the derivative being defined backwards, like Eq. (23) and Eq. (24) state.

$$\partial_1 g(\vec{A}) = g(\vec{A}) - g\begin{pmatrix} A_1-1 \\ A_2 \end{pmatrix} \tag{23}$$

$$\partial_2 g(\vec{A}) = g(\vec{A}) - g\begin{pmatrix} A_1 \\ A_2-1 \end{pmatrix} \tag{24}$$

In the case of a two-dimensional grid the pathintegral mentioned in Eq. (18) can be expressed as Eq. (25) suggests. Here $g^{\mu\nu}$ represents the contravariant metric tensor with μ and ν being indices that are either 1 or 2 due to the framework of the two-dimensional case. The time dependent path is denoted as $z(t)$ in the form of a vector consisting of two components. The

distance between two events is referenced as $s(x,y)^2$. Whether the metric tensor is contravariant and the path is covariant or the situation is reversed does not matter, the result will stay the same. Nonetheless, it is still important that they are not both contravariant or covariant in order for the formula to work. During the course of this thesis the contravariant metric tensor is used consistently. [1]

$$s(x,y)^2 = \int_x^y g^{\mu\nu} \frac{\mathrm{d}z(t)_\mu}{\mathrm{d}t} \frac{\mathrm{d}z(t)_\nu}{\mathrm{d}t} \, dt \tag{25}$$

The metric tensor must fulfil the invertibility condition in order to possess both a contravariant and a covariant form. The mathematical expression is given in Eq. (26). Therefore the metric of each individual event needs to have a non-zero determinant. Since the product of both tensors must be equal to the unit matrix, this equation can be expressed with the Kronecker-Delta δ^μ_ρ in a more formal way. [1]

$$g^{\mu\nu} g_{\nu\rho} = \delta^\mu_\rho \tag{26}$$

4 Demonstration of the formalism and methodology

This thesis can be understood as the continuation of the Bachelor Thesis of Christina Perner titeled "Metric reconstruction with causal dynamical triangulation" [1]. In her work, it was shown that a contradiction will inevitably arise if an event is in possession of two geodesics to other events that are oriented in different directions. This issue prevents to obtain a solution for the metric tensor in every given event.

It has to be said that the approach of this work differs from the approach of the previous thesis where the solution to the problem was handled directly via the connections as can be seen in the appendix of [1]. The approach in this work focuses on the solution of a set of equations considering all of the distances, the invertibility conditions and the gauge conditions of every event.

It has already been proven in [1] that the formalism itself is working properly. Henceforth, it will be shown that the new approach is operating as intended and is able to solve the issue at hand. For the sake of demonstration, a simple grid with nine events and only one cross connection is used, which Fig. 1 illustrates. For the sake of better visualization, every timelike connection is coloured blue and every spacelike connection is coloured red [9].

[9] The same colour code is used for every figure throughout the whole thesis.

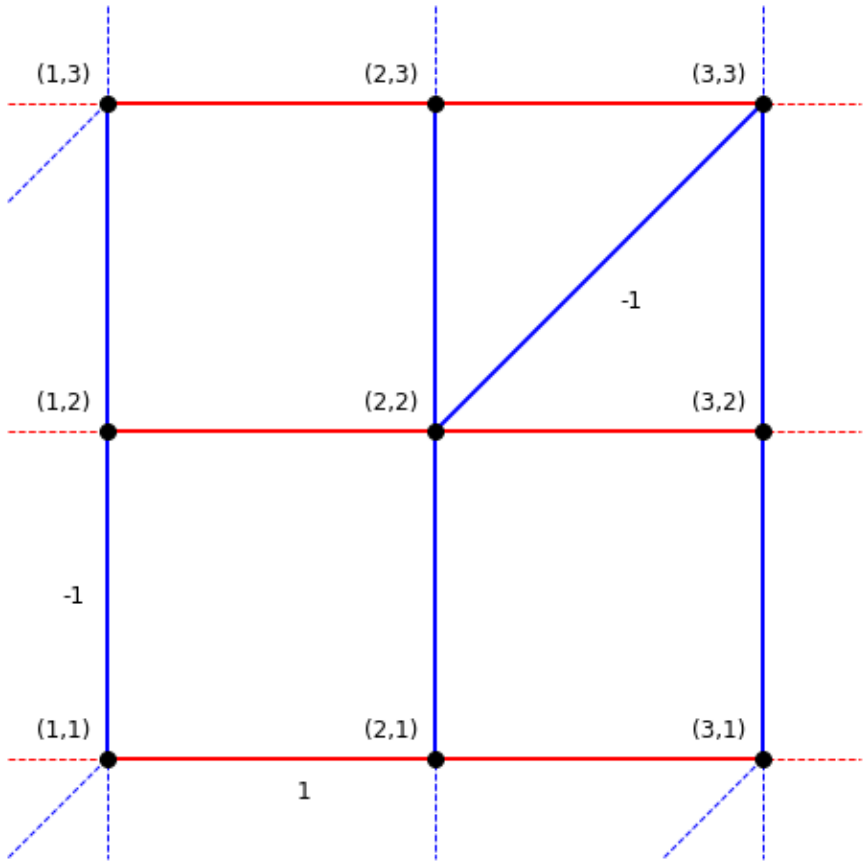

Figure 1: Grid with nine events and one cross connection. The dotted lines mark a periodical extension of the grid.

The first part of setting up the needed set of equations is to formulate the gauge condition for every event in question. The Haywood gauge in Eq. (20) needs to hold up for every single event where the derivative can be handled like in Eq. (21) and Eq. (22) as well as in Eq. (23) and Eq. (24). The indices μ and ν can be either one or two since the metric tensor is of rank two resulting in, e.g. for the contravariant metric tensor, g^{11}, g^{12}, g^{21} and g^{22}. Since the metric tensor needs to be a symmetric tensor the elements g^{12} and g^{21} will always result in the same value, which will be made use of in the course of the calculation. The full form of the Haywood gauge in this case is shown below in Eq. (27).

$$g^{11}\partial_1 g_{11} + g^{12}\partial_2 g_{11} + g^{12}\partial_2 g_{12} + g^{21}\partial_1 g_{22} + g^{21}\partial_1 g_{21} + g^{22}\partial_2 g_{22} + g^{22}\partial_2 g_{21} + g^{11}\partial_1 g_{12} = 0 \tag{27}$$

An example how this translates to the problem at hand is given in Eq. (28) where the written out form of Eq. (27) is used on the first event (1,1) from the grid in Fig. 1.

$$\begin{aligned} &g^{11}(1,1)(g_{11}(2,1) - g_{11}(1,1)) + g^{12}(1,1)(g_{11}(1,2) - g_{11}(1,1)) \\ &+g^{12}(1,1)(g_{12}(1,2) - g_{12}(1,1)) + g^{21}(1,1)(g_{22}(2,1) - g_{22}(1,1)) \\ &+g^{21}(1,1)(g_{21}(2,1) - g_{21}(1,1)) + g^{22}(1,1)(g_{22}(1,2) - g_{22}(1,1)) \\ &+g^{22}(1,1)(g_{21}(1,2) - g_{21}(1,1)) + g^{11}(1,1)(g_{12}(2,1) - g_{12}(1,1)) = 0 \end{aligned} \tag{28}$$

Incidentally this procedure is repeated for every point in the grid with the use of periodic boundary conditions. In this context, a timelike connection from point (1,3) onwards is equivalent to a connection from (1,1) onwards.

The next set of equations that need to be fulfilled entails the invertibility condition from Eq. (26), which results in Eq. (29) if written out. These equations must hold true for every event in the grid so that in combination with the Haywood gauge the possible values of the metric elements are limited.

$$\begin{aligned} g^{11}g_{11} + g^{12}g_{21} &= 1 \\ g^{11}g_{12} + g^{12}g_{22} &= 0 \\ g^{21}g_{11} + g^{22}g_{21} &= 0 \\ g^{21}g_{12} + g^{22}g_{22} &= 1 \end{aligned} \tag{29}$$

The only information that is already given is the distance between the events. Every spacelike path has a length of 1 and every timelike path takes on a length of -1 per definition. The specific values of these lengths are not set in stone and could be exchanged if the need arises. However, these values are specifically chosen since the are easy to work with and have proven to be an adequate choice so far. This information is crucial and has to be considered via the path integral from Eq. (25). Therefore every length needs to be equal to $s(x,y)^2$ for every path. As it has already been shown in the work of Christina Perner [1], the below mentioned paths $z_{horizontal}(t)$ and $z_{vertical}(t)$ can be used for the horizontal and vertical paths that only follow one direction. For a timelike path that is diagonal in the grid the path in the form of $z_{diag}(t)$ can be used, as shown below. Which variant of $z_{diag}(t)$ is used does not matter at all since the minus sign vanishes anyway in this case due to Eq. (25).

$$z_{horizontal}(t) = \begin{pmatrix} t \\ 0 \end{pmatrix}, z_{vertical}(t) = \begin{pmatrix} 0 \\ t \end{pmatrix} \text{ and } z_{diag} = \frac{1}{\sqrt{2}} \begin{pmatrix} t \\ t \end{pmatrix} \text{ or } z_{diag} = \frac{1}{\sqrt{2}} \begin{pmatrix} -t \\ -t \end{pmatrix}$$

The next equations entail all possible paths for every event and are the key part for extracting the metric since these equations offer concrete values of distances to work with. An example will be given for the timelike connection from event (1,1) to event (1,2) in Eq. (30). It should be noted that according to the grid the term $t+1$ would be more accurate for $t \in [0,1]$ for the vertical connection. Since the linear term will always vanish in the derivative and the interval does not matter as long as the difference between start and end stays the same, the result is not influenced and therefore Eq. (30) is still correct. The fact that g^{12} and g^{21} are equal has been abused to shorten the calculation. In the same way, by applying the correct vector to the correct connection, a path integral for every necessary connection of every event can be obtained.

$$\begin{aligned} -1 = \int_0^1 g^{11} \cdot \frac{\mathrm{d}z_{vertical}(t)_1}{\mathrm{d}t} \frac{\mathrm{d}z_{vertical}(t)_1}{\mathrm{d}t} + 2g^{12} \cdot \frac{\mathrm{d}z_{vertical}(t)_1}{\mathrm{d}t} \frac{\mathrm{d}z_v(t)_2}{\mathrm{d}t} \\ + g^{22} \cdot \frac{\mathrm{d}z_{vertical}(t)_2}{\mathrm{d}t} \frac{\mathrm{d}z_{vertical}(t)_2}{\mathrm{d}t} \, dt \end{aligned} \tag{30}$$

It is crucial that the path integral for the diagonal connection is not only applied to the events (2,2) and (3,3), but to (1,1), (1,3) and (3,1) as well. Due to the periodical boundary conditions,

all of these events possess a diagonal connection. This can be verified by periodically extending the grid, like it is indicated in Fig. 1.

With all of the above mentioned equations one big system of linear equations has been set up. To compute a system of this scale "Wolfram Mathematica" has been used. The polished code can be seen in the appendix. It should be mentioned that the grid from Fig. 1 does not represent a proper spacetime structure in the sense of CDT since there are only two triangles to work with but this grid serves for the purpose of demonstration and can be solved nonetheless. The results are shown below.

$$g(1,1) = g(1,2) = g(1,3) = g(2,1) = g(2,2) = g(2,3) =$$

$$g(3,1) = g(3,2) = g(3,3) = \begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix}$$

To further justify that the approach itself is working it has been tested on a real possible spacetime structure according to CDT. In this case the structure is shown in Fig. 2.

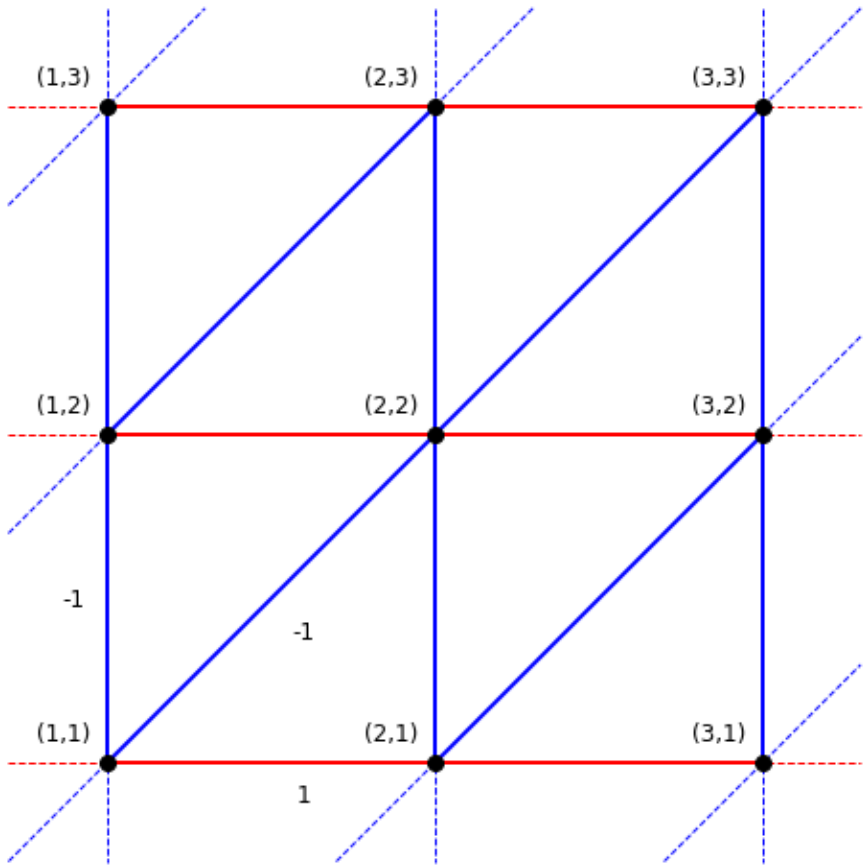

Figure 2: Grid with nine events and four identically oriented cross connections. The dotted lines mark a periodical extension of the grid.

The set of equations is essentially the same with the addition of three more timelike connections that point all in the same direction. In the end, the result does not differ and is therefore exactly the same as for the grid with only one timelike diagonal connection. This occurs due to the fact that they all point in the same direction and subsequently do not cause any singularities that arise contradictions. If for example the timelike connection between (1,2) and (2,3) would be mirrored to be a connection between (1,3) and (2,2) the gauge would collapse without further adjustments.

One crucial aspect of the structure given in Fig. 2 is the curvature of spacetime. In the grid, the triangles that emerge due to the cross connections seem to be rectangular triangles. Regardless of what the triangles seem to be, they are still equilateral which is a truth obtained by taking a look at the distances that make up these triangles. In this sense, the connection is a timelike connection because the connection between (1,1) and (2,2) is at an angle of 60° even though it looks like it would be an angle of 45° in the grid. This fact should be considered and becomes especially important if the dual grid is set up.

If another vector is used to describe the cross connection between events, the elements g^{12} and g^{21} would change. For the purpose of demonstration, an example is the below mentioned vector $z_{diag,flat}(t)$ used to describe the cross connection between two events if the grid would be given in flat space time even though the structure itself would collapse.

$$z_{diag,flat}(t) = \frac{1}{2} \cdot \begin{pmatrix} t \\ \sqrt{3} \cdot t \end{pmatrix}$$

Using $z_{diag,flat}(t)$ instead of $z_{diag}(t)$ while every other equation stays the same results in the same metric for every event, which is given below.

$$g(1,1) = g(1,2) = g(1,3) = g(2,1) = g(2,2) = g(2,3) =$$

$$g(3,1) = g(3,2) = g(3,3) = \begin{pmatrix} 1 & -\frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} & -1 \end{pmatrix}$$

Since different vectors lead to different solutions, the choice of a suitable vector is essential for obtaining the correct solution of the metric. However, since there are solutions for more than one vector, the choice is not unique if the primary goal is only to prove that the metric can be extracted.

Since the method to extract the metric in itself is operating as intended, the next sections cover both ideas to fix the issue of overdetermination and the issue of an occurring singularity.

5 Determination of the metric with the use of "support events"

The idea behind this approach is to place so-called "support events" in between events in order to shift the diagonal connection from proper events to "support events" and send the distance between the "support events" and their nearest neighbour event to zero in the limit. The events and the "support events" should not overlap initially and still have a short gap in between. This distance is chosen to be a specific value α and is shortened to see what happens if this distance becomes small enough. The "support events" are only needed if the standard approach shown in the previous section fails to produce a result for the metric tensor because of a contradiction. The main grid of interest for this thesis is given in Fig. 3. One example of a contradiction would be the point (2,2) where two timelike connections cross this point from different directions, resulting in no solution whatsoever. For further reasoning why a contradiction arises, see [1].

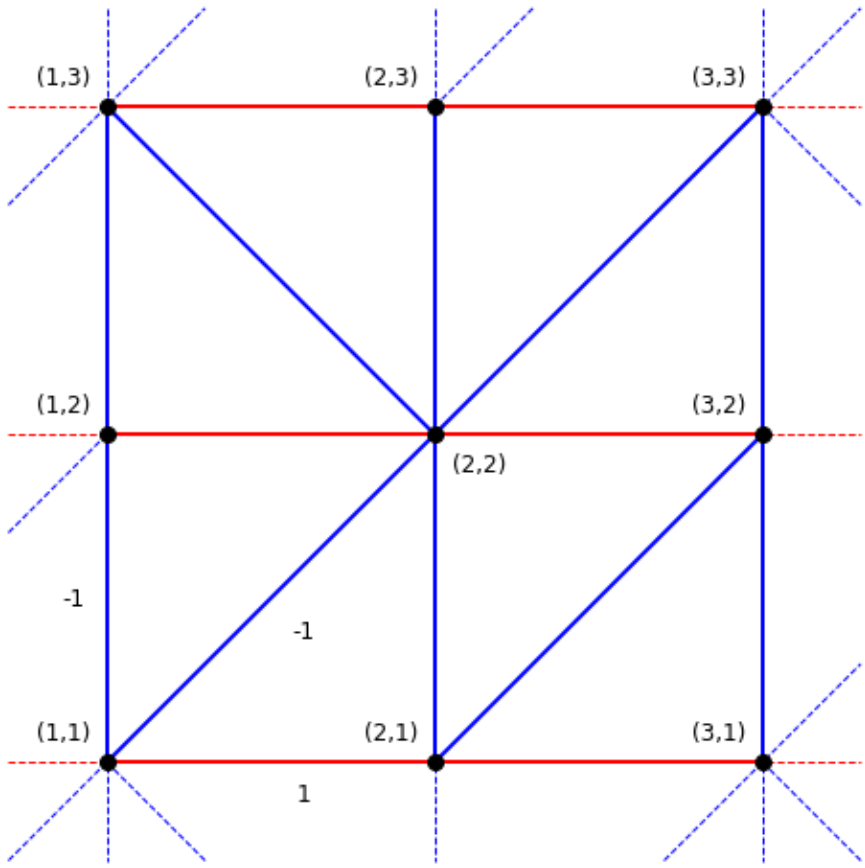

Figure 3: Grid with nine events and four cross connections. To create a contradiction, the connection from (1,2) to (2,3) in Fig. 2 is replaced by a connection from (1,3) to (2,2). The dotted lines mark a periodical extension of the grid.

The implementation of support events for the grid given in Fig. 3 results in the structure shown in Fig. 4.

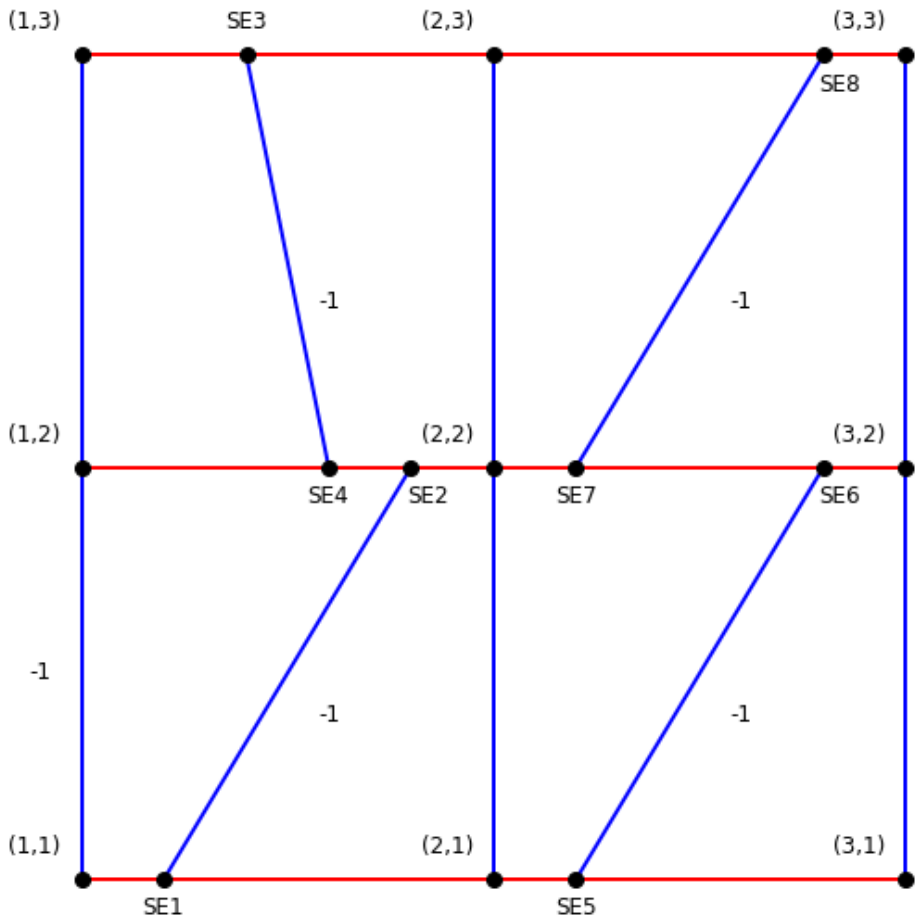

Figure 4: Grid of Fig. 3 after the implementation of "support events".

Figure 4 describes the base grid without giving too much thought regarding the periodical continuation of the grid. If this grid should periodically repeat itself, more "support events" need to be accounted for. Taking this into consideration, Fig. 5 is the grid of interest which will be studied further. The grid in Fig. 4 is not viewed as being the correct representation of the problem. However, due to technical limitations, the viability of this grid could not be disproved by a proper variation of the code used for dealing with the structure in Fig. 5.

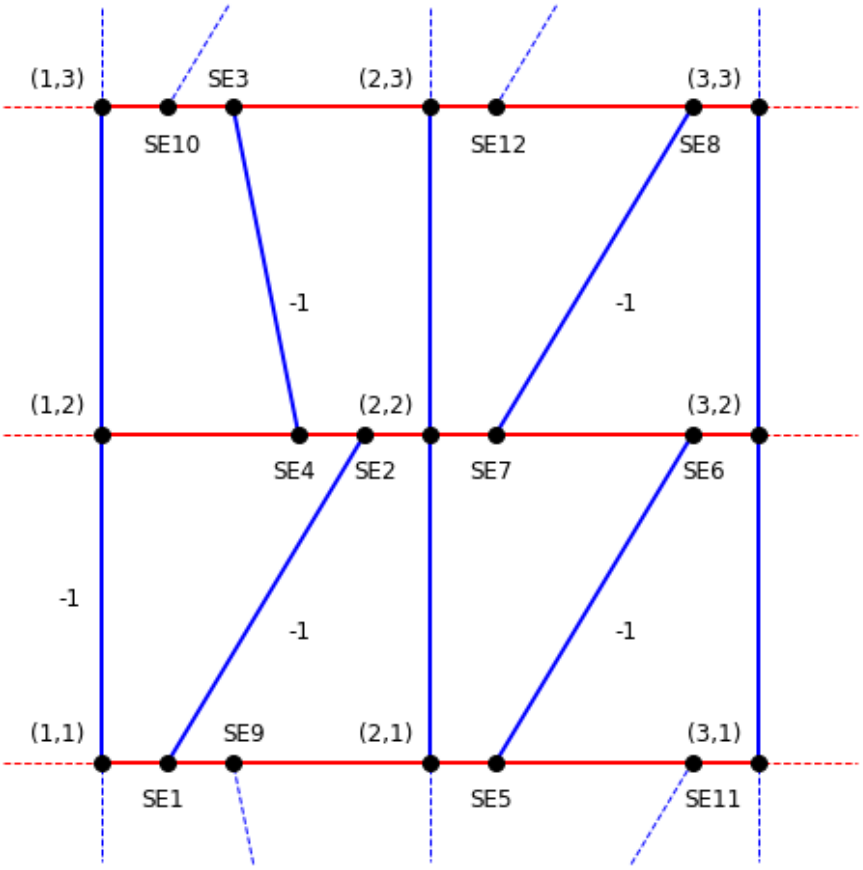

Figure 5: Grid of Fig. 4 after taking the periodic expansion of the grid into account.

The "support events" in Fig. 4 and Fig. 5 are labelled "SE" with a corresponding number attached. Since a "support event" is placed between two events, the distance to the closest proper event will be denoted α. The distance between "SE1" and "SE9" (and therefore between "SE10" and "SE3") is also set to be α in order to simplify the calculation. For the sake of simplicity this holds true for most implemented "support events". One further exception aside from the case of "SE10" and "SE3" is "SE4" where the distance to (2,2) is set to be 2α because if only α would be chosen "SE4" would overlap with "SE2", therefore reenacting the contradiction of (2,2). This is the exact same reason why "SE1" and "SE9" are set a distance of α apart. It is not necessary to set up different distances for every single "support event" since the distance α is always bound to be zero in the limit, but this would be a viable choice nonetheless.

It would also be possible to introduce only one "support event" per connection while the other event remains a regular event. In doing so it is necessary to pick the correct event for this endeavor. Both variants work out if handled correctly, though the solutions differ. If for example the event (3,3) is chosen to be connected to "SE7", the event (1,3) can not be chosen to be connected with "SE4". Because of the periodicity this choice would lead to the occurrence of a contradiction. By taking the diagonal connections only between "support events" no contradiction can arise in this sense. For this reason, the timelike geodesic for the diagonal connection is chosen to be between two support events for the majority of calculations performed in this thesis.

In order to solve the problem, the "support events" need to be treated like regular events in terms of the gauge, therefore creating a larger system of equations. Every condition that regular events have to obey must hold true for "support events" as well, like the invertibility condition and the Haywood gauge.

The overall approach of tackling the issue remains the same as explained in the section "Demonstration of the formalism and methodology". The complete set of equations is shown in the corresponding code embedded in the appendix. The only exception is the vector that connects the "support events". Without the use of "support events", before mentioned $z_{diag}(t)$ would be a viable choice for the vector of the cross connection. Now the connection has a shortened horizontal distance to cover while the vertical distance remains unchanged as well as the length of the connection overall. Therefore the diagonal connection between two "support events" is steeper than between two proper events. The angle of the connection scales with the shortened horizontal distance, in this sense the angle is dependent on α. The chosen vector for the connection between "SE3" and "SE4" is labelled $z_{diag,34}(t)$ and the used vector for every other diagonal connection between "support events" is denoted $z_{diag,SE}(t)$. Both vectors are given below.

$$z_{diag,SE}(t) = \sqrt{\frac{1}{1+(\frac{1}{1-2\alpha})^2}} \begin{pmatrix} t \\ \frac{1}{1-2\alpha} \cdot t \end{pmatrix}, \; z_{diag,34}(t) = \sqrt{\frac{1}{1+(\frac{1}{1-4\alpha})^2}} \begin{pmatrix} t \\ \frac{-1}{1-4\alpha} \cdot t \end{pmatrix}$$

The choice of a vector capable of obtaining a result for the metric tensor is not unique. For example, the vector $z_{diag}(t)$, which worked out in the grid of Fig. 2 could also be a possible choice. This is the vector that arises in the limit for $\alpha = 0$. The factor of $\sqrt{3}$ could also be considered in the second entry of the vector since the real slope is steeper as it seems in the grid. All of the above mentioned variants have been tested and lead all to the same conclusion for the grid of Fig. 5.

After creating the set of equations and running the calculation, no success in extracting the metric tensor properly is achieved. Henceforth, the result is $g = \{\}$ and the metric for every event is not known. If the metric of only one event leads to a contradiction or is unable to be solved properly, every other event is missing crucial information. Since every event is connected to sufficiently other events, the result of the metric of one event heavily influences the determination of the metric of a connected event. Therefore, the set of equations is indispensable because calculating only one event and start the further calculation from there on leads to wrong or non-existing solutions.

The way the problem was handled in the code involved SE1 and SE10 being treated as different, separate "support events". In a sense, SE10 is a relic of the periodical continuation and not different from SE1. Thus, SE10 possesses a diagonal connection to SE2. The Haywood gauge is dependent on the derivative defined by the position of one event in the grid compared to the next event given for the same coordinate. In this sense, the connection from SE1 to SE2 has been chosen for the derivative in the vertical direction for SE1 in the gauge. If more "support events" would be included in a way that every "support event" would be clearly timelike connected only in the vertical direction, like the connection between (1,1) and (1,2), the problem

is still not solvable. In this case, it does not matter if the timelike connection is established by defining the same vertical connection with the same distance or if the connection only appears in the grid without further description. Every calculation for the grid in Fig. 5 was performed multiple times for different values of α, yet none of them led to a solution for the metric tensor. It was tested if it would make a difference to replace every appearance of periodicity with the original "support event", like replacing SE10 by SE1 and SE9 by SE3. In order to do so, the code in the appendix has been modified, but no result was obtained due to technical limitations regarding the emerging system of equations.

Since the approach seems to fail so far, the general viability is tested. This is done on a simple grid with four events, two "support events" and only one cross connection given in Fig. 6. Without the introduced "support events", a connection between the events (1,1) and (2,2) would lead to a perfectly fine grid with the same result as the grid of Fig. 2. Here the spacelike distance between every event and "support event" is set to α. This leads to a set spacelike distance of $1-2\alpha$ between two "support events".

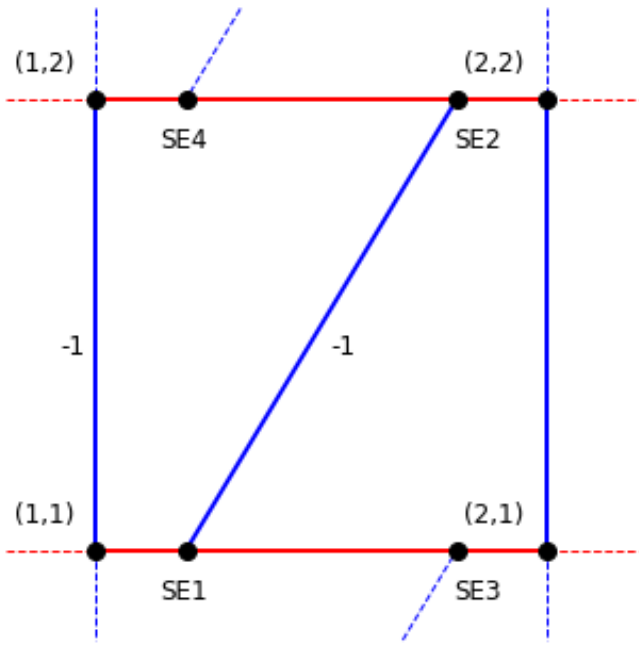

Figure 6: Grid with four events and four "support events". The dotted lines mark a periodical extension of the grid.

The grid given in Fig. 6 could indeed be solved. Interestingly, there are multiple solutions and the problem offers a concrete solution for the metric tensor in various ways. Considering the periodical boundary conditions, SE4 is essentially SE1 and SE3 equals SE2. If they are treated equally and a system of equations is set up for six events to solve in which SE2 is treated as the next event in the vertical direction from SE1, the problem could be solved. This involves the derivative defined backwards for the vertical connection of SE2, as it can be seen in the corresponding code in the appendix. The solution is dependent on the set value of the distance α between (1,1) and SE1 that is involved in every distance regarding "support events". The values 0.2, 0.01 and 0 have all been tested and it was verified that they lead to a solution. For a distance of zero, the events overlap, recreating the image of the original grid. However, the

original solution, which is the same as for the grid in Fig. 2, could not be recovered. Instead, the resulting metric looked the following way if α is set to zero:

$$g(1,1) = g(1,2) = g(2,1) = g(2,2) = \begin{pmatrix} 0 & -1 \\ -1 & -1 \end{pmatrix}$$

$$g(\text{SE1}) = \begin{pmatrix} 1 & -2 \\ -2 & 1 \end{pmatrix}$$

$$g(\text{SE2}) = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$

If the code is modified and only two "support events" would be utilized by removing SE2 and SE3, therefore establishing a connection from SE1 to (2,2), the original metric would be recovered for (1,2), (2,2) and SE1.

This is not the only option how a solution could be obtained. If the gauge would be defined in a way where SE1 would be the next vertical event for SE1 itself, another solution is achieved. This solution can only be received if no proper connection with a distance of -1 is demanded. If there would be such a connection, this connection must have a length of -1 corresponding to the other timelike connections. However, if this requirement is considered, no solution can be found. By willingly not defining a specific length, a solution can be obtained for various values of α. In the limit however, the original metric could also not be recovered for any event.

The original metric could be achieved if the "support events" are treated distinct with SE4 being an own "support event" with its own conditions. If now the gauge would be defined in a way where SE4 would be the next vertical event for SE1 and the distance α is set to zero, the original metric is at least recovered for the "support event" SE1. However, this is only the case if no specific value for the timelike connection is set and multiple solutions were received. One such solution is given below.

$$g(1,1) = g(1,2) = g(2,1) = g(2,2) = \begin{pmatrix} 0 & -1 \\ -1 & -1 \end{pmatrix}$$

$$g(\text{SE1}) = \begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix}$$

$$g(\text{SE2}) = g(\text{SE3}) = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$

$$g(\text{SE4}) = \begin{pmatrix} 1 & -2 \\ -2 & 1 \end{pmatrix}$$

Since there have been existing solutions for the grid in Fig. 6, another interesting structure regarding the inclusion of "support events" is given in Fig. 7 with the underlying idea of testing the simplest structure involving a contradiction. If no "support events" would be introduced, the metric for the event (2,1) can not be obtained because of the connection to (1,2) and the opposite directed connection to (3,2). It should be mentioned that this would not be the only singularity in this case. In fact, every single event could be considered a singularity due to periodical extension.

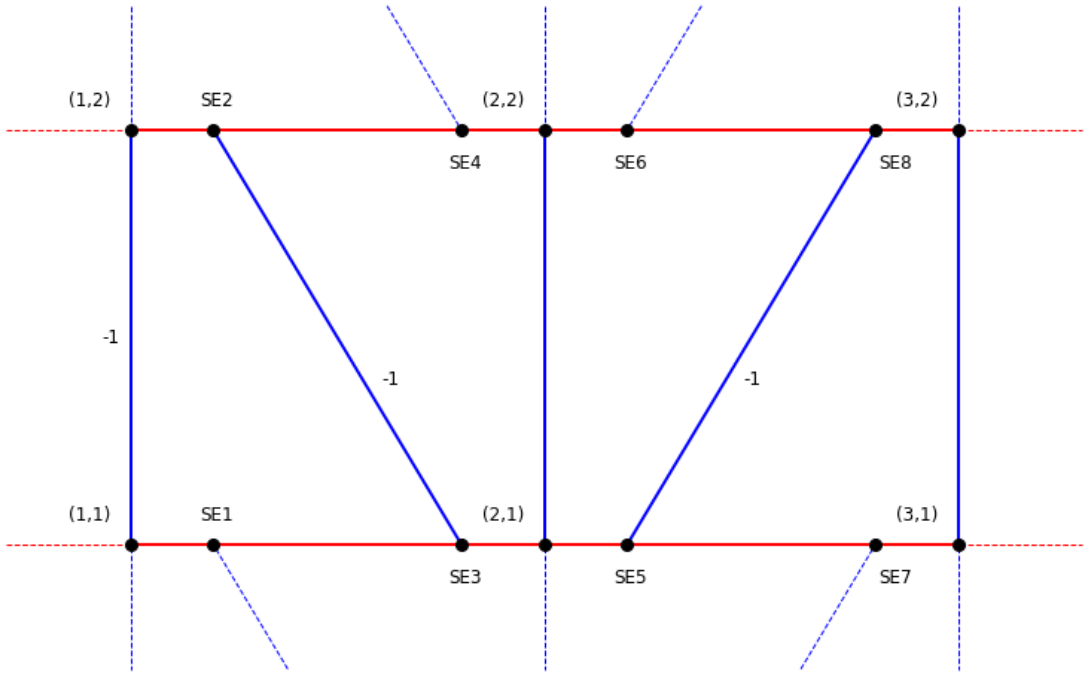

Figure 7: Grid with six events and eight "support events". The dotted lines mark a periodical extension of the grid.

As it was the case for Fig. 6, the metric tensor for the grid in Fig. 7 could be obtained. If the "support events" SE1 and SE2 are treated as one and the same and in addition the same holds true for SE3 and SE4, SE5 and SE6 as well as SE7 and SE8, a system of equations containing ten events ("support events" included) can be formulated. By solving the system giving in the corresponding code in the appendix, two results are found. One of the solutions is given below.

$$g(1,1) = g(1,2) = g(3,1) = g(3,2) = \begin{pmatrix} 0 & -3 \\ -3 & -1 \end{pmatrix}$$

$$g(2,1) = g(2,2) = \begin{pmatrix} 0 & -1.5 \\ -1.5 & -1 \end{pmatrix}$$

$$g(\mathrm{SE1}) = g(\mathrm{SE2}) = \begin{pmatrix} 1 & -1 \\ -1 & -5 \end{pmatrix}$$

$$g(\mathrm{SE3}) = g(\mathrm{SE4}) = \begin{pmatrix} 0 & -2 \\ -2 & -6 \end{pmatrix}$$

$$g(\mathrm{SE5}) = g(\mathrm{SE6}) = \begin{pmatrix} 1 & -3 \\ -3 & 3 \end{pmatrix}$$

$$g(\mathrm{SE7}) = g(\mathrm{SE8}) = \begin{pmatrix} 0 & -2 \\ -2 & 2 \end{pmatrix}$$

It is rather strange that the element $g^{12} = g^{21}$ is negative for every event in the grid. Normally, the diagonal connection between SE2 and SE3 would suggest a positive value of g^{12} due to the direction. To obtain the above mentioned solution, the gauge of SE1 involved SE3 as the next event horizontally and vertically. It has been tried to define a vertical connection between the "support events" like between SE1 and SE2 while they are still treated as the same "support event". This way no solution could be obtained. The same has been tried by making a distinction between the "support events", but to no avail. Because of technical limitations, this system of equations could not be computed.

Since a solution exists for the grids in Fig. 6 and Fig. 7, but not in the case of Fig. 5, the approach itself does not operate as intended. Everything that worked out for Fig. 6 and Fig. 7 was also tested on the grid of Fig. 5, but to no avail. The solutions that could be obtained were not unique, but heavily dependent on how the gauge was defined and how exactly the periodicity was handled. A plausible assumption is that the symmetry of the structure influences the solvability. A grid resembling Fig. 6 would consist of the same basic structure with only four proper events. If this is treated as a cell, every cell would be the smallest possible unit while every unit is equivalent. The symmetry decreases in Fig. 7, but the basic building block of the whole structure would be still small with one half being the mirrored image of the other half. The structure in Fig. 3 would have the biggest basic building block to repeat the structure uniformly and one half can not be seen as the mirrored version of the other half. The grid covered in the next section given in Fig. 9 would be more symmetric than the structure of Fig. 3. However, even in a more symmetric case, the metric could still not be obtained. By trying the approach on the more symmetric structure in the way illustrated by Fig. 8, the result was unaffected. Everything that did not work out for Fig. 5 did not work out for Fig. 8 either. The same approach used in the code for Fig. 5 led to no results, the other variations could not be calculated due to technical limitations.

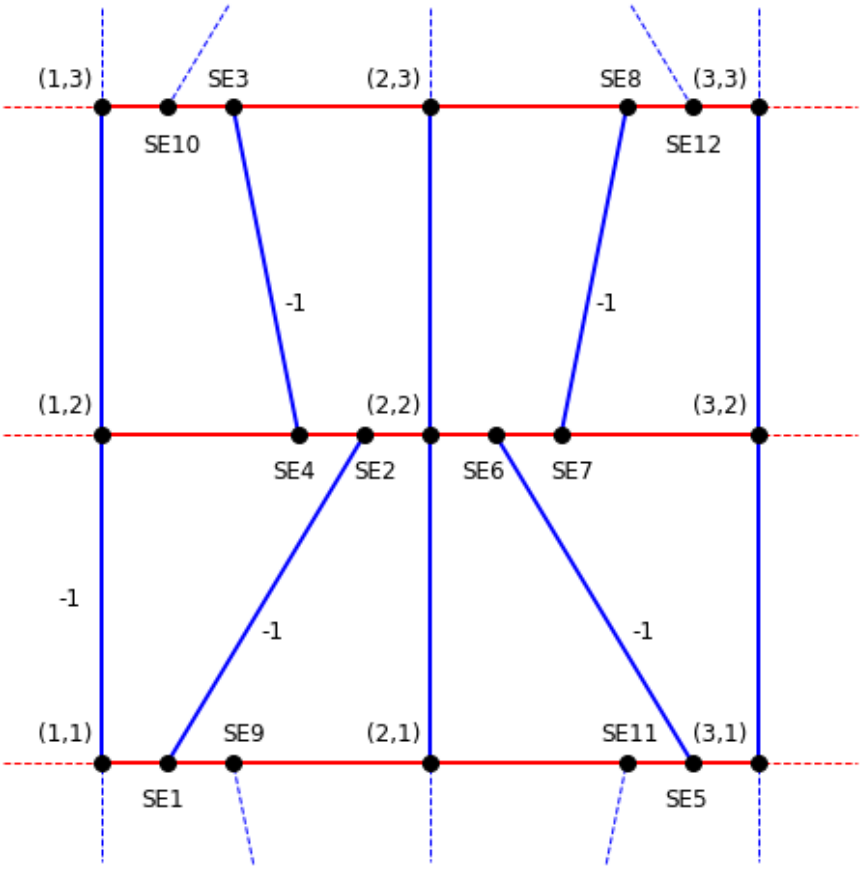

Figure 8: Grid of Fig. 9 in the case of implemented "support events". The dotted lines mark a periodical extension of the grid.

In summation, the implementation of "support events" is not able to resolve the issue of occurring singularities in the spacetime-structure. The approach is viable for simpler structures without singularities like Fig. 6 and simpler structures that would possess singularities like Fig. 7. For a more complex grid, like the grid demonstrated in Fig. 5, the metric could not be obtained.

6 Determination of the metric via the implementation of a dual grid

The underlying idea is based on the implementation of a dual grid in the spacetime-structure. This has a higher impact on the system of equations than the approach using "support events". So far only the vertices on the grid where horizontal and a vertical paths interfere and cut each other have been considered to be proper points on the grid. With the timelike paths as a diagonal, a clearly defined structure with proper triangles has been established. By using the centre of the triangles and making connections between the nearest neighbours of every single centre of a triangle in the old structure, a new grid arises. Both grids are equally suitable to describe the spacetime-structure, only resulting in different distances between the events and subsequently

resulting in another metric. Not only the positions of the events change, but also their numbers. In the previously studied grid nine events were discussed for the original case and 21 events counting "support events" as well. Within the dual grid only eight events need to be considered.

To properly show that the implication of the dual grid solves or does not solve the problem, this work is done on two different structures. The first structure is the same as in Fig. 3 and will be called "grid 1" from now on. The other structure arises if the timelike connection between (2,1) and (3,2) is mirrored to connect the events (2,2) and (3,1) instead. This grid, denoted as "grid 2", can be seen in Fig. 9.

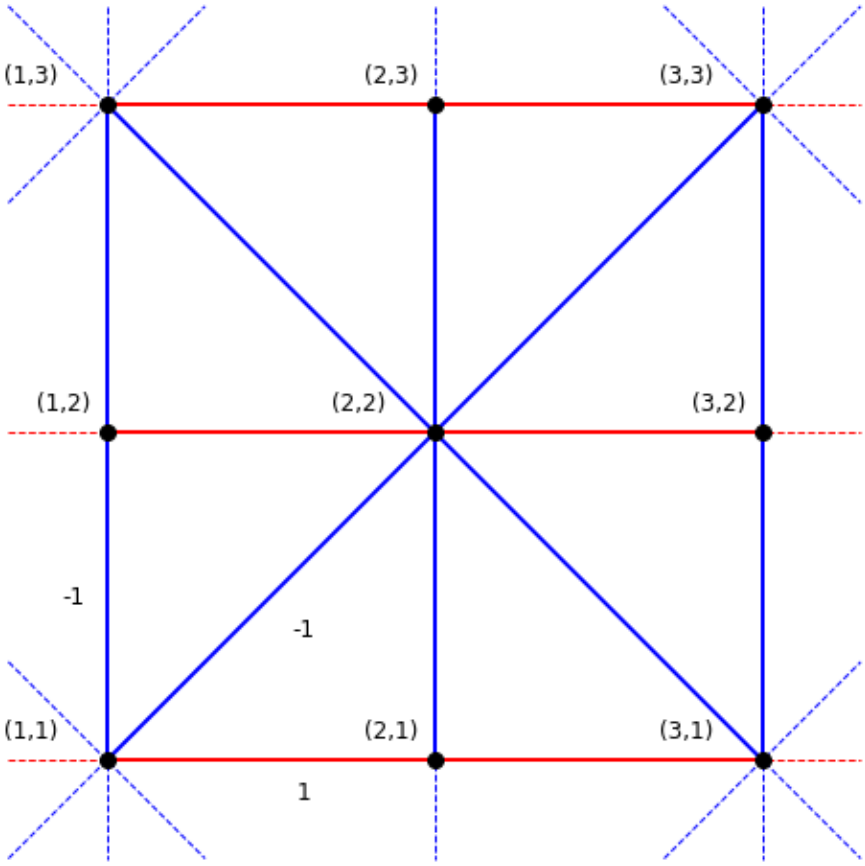

Figure 9: "Grid 2" with 9 events and four cross connections. The dotted lines mark a periodical extension of the grid.

Since "grid 2" offers the more symmetric structure this will be the structure to demonstrate the dual grid and which the first calculation will be performed on. If the centre of every triangle is taken as an event and connected to the surrounding centres of the neighbouring triangles, the dual grid in Fig. 10 emerges.

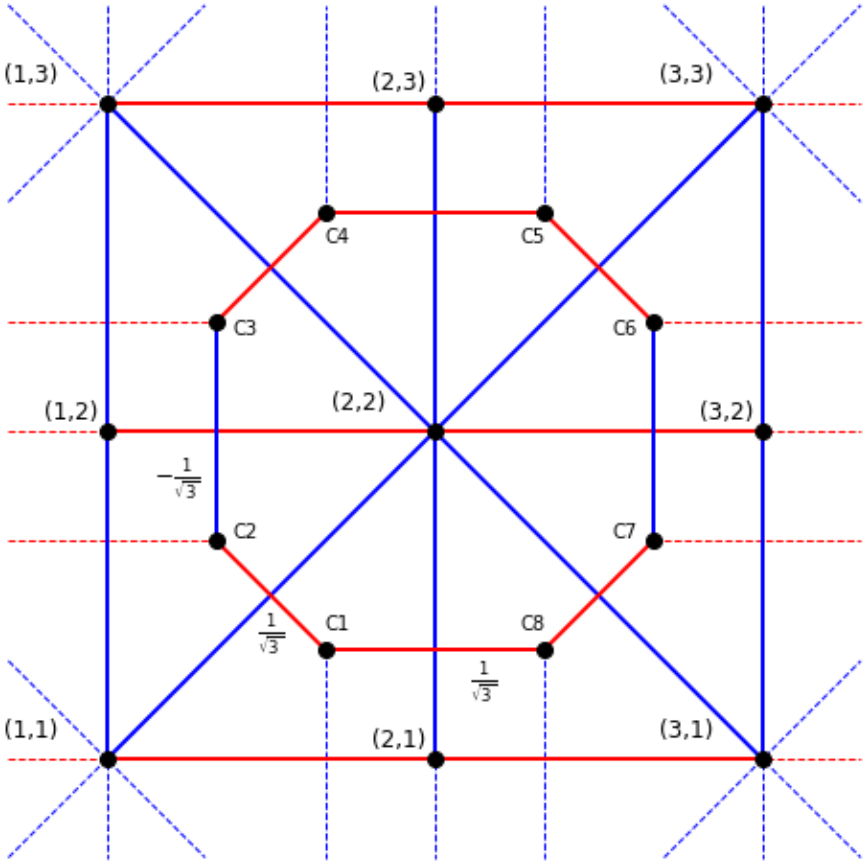

Figure 10: Corresponding dual grid to “grid 2”. The dotted lines mark a periodical extension of the grid. The centres in the figure only serve demonstration purposes for a better visualization of the structure. These centres do not show the true locations of the centres on the grid.

These new connections lead to a change of distances. The new distances between the events are calculated like in Euclidean space. In the grid the triangles may seem to be rectangular, but in reality they are still equilateral triangles. The structure has to be considered to be curved and if the structure is flattened, it inevitably collapses. This is shown in Fig. 11 which demonstrates the spacetime-structure how it would appear if it was flattened. The collapse can be clearly seen by taking a look at the void that emerged. What would normally be one single event (2,2) is now split apart.

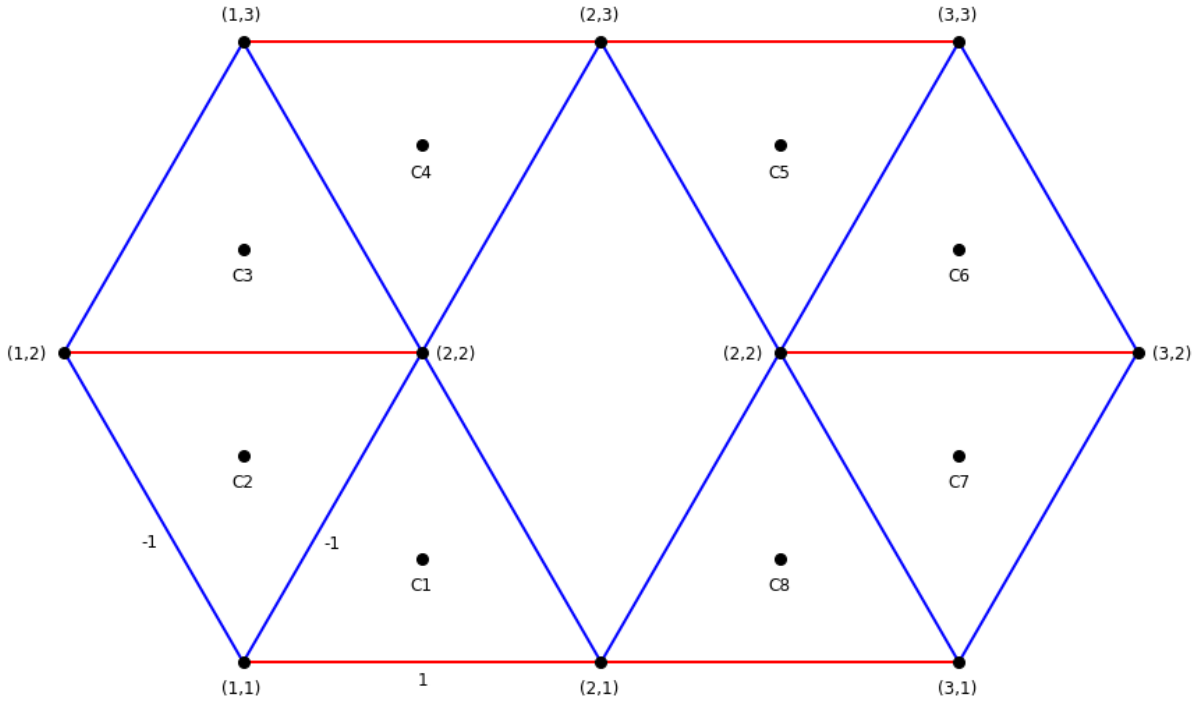

Figure 11: Flattened spacetime-structure of "grid 2". The corresponding centres of the triangles are denoted with a C.

Figure 11 serves as a tool to calculate the distances in between the centres. Even if curved, the distances still remain the same, as it was the case before. Since every triangle is equilateral with an edge length of 1 (neglecting the sign), the height of the whole structure equals $\sqrt{3}$ and the total length is 3. The total height becomes immediately clear if the altitude of an equilateral triangle is considered, which equals $\frac{\sqrt{3}}{2}$ in the case of a triangle with an edge length of 1. To make the calculation of the distances easier, an imaginary coordinate system is laid upon the structure. By putting the origin of the coordinate system underneath (1,2) at the same height as (1,1), the simplest choice has been taken. Using this, the events are at the following position:

$$
\begin{array}{ccc}
(1,1) : (\frac{1}{2},0) & (2,1) : (\frac{3}{2},0) & (3,1) : (\frac{5}{2},0) \\
(1,2) : (0,\frac{\sqrt{3}}{2}) & (2,2)_1 : (1,\frac{\sqrt{3}}{2}) & (2,2)_2 : (2,\frac{\sqrt{3}}{2}) \\
(3,2) : (3,\frac{\sqrt{3}}{2}) & (1,3) : (\frac{1}{2},\sqrt{3}) & (2,3) : (\frac{3}{2},\sqrt{3}) \\
(3,3) : (\frac{5}{2},\sqrt{3}) & C1 : (1,\frac{\sqrt{3}}{6}) & C2 : (\frac{1}{2},\frac{\sqrt{3}}{3}) \\
C3 : (\frac{1}{2},\frac{2\sqrt{3}}{3}) & C4 : (1,\frac{5\sqrt{3}}{6}) & C5 : (2,\frac{5\sqrt{3}}{6}) \\
C6 : (\frac{5}{2},\frac{2\sqrt{3}}{3}) & C7 : (\frac{5}{2},\frac{\sqrt{3}}{3}) & C8 : (2,\frac{\sqrt{3}}{6})
\end{array}
$$

From here on, the difference between the positions of the centres needs to be calculated. The calculation is straightforward by applying the Pythagorean theorem or reading the difference off directly. Therefore the distance between C1 and C8, C4 and C5, C2 and C7 as well as C3 and C6 is 1, whereas every other connection between the centres is $\frac{1}{\sqrt{3}}$. Here the length of the connection between C3 and C6 as well as C2 and C7 is determined by a periodical continuation of the structure. It is not possible to periodically extend the structure without the emergence of further voids. The length of the connection between C1 and C4 is the same as for the geodesic connecting C2 and C3, which is also obtained by a periodical extension of the structure. Since

in Fig. 10 every single triangle is joint together in curved spacetime, the distance geodesic connecting the centres is the same for every single event. Therefore, every centre is connected to another centre with a geodesic of $\frac{1}{\sqrt{3}}$. Depending on the type of the connection, the sign changes. Whether the connection between the centres is spacelike or timelike depends on the connections of the original grid. For example in the case of the connection between C1 and C2, this connection is spacelike because of the angle it cuts through the original connection between (1,1) and (2,2). By looking at the connection of the original grid and at which angle the connection of the centres are in relation, it becomes clear which connection has to be spacelike and which connection has to be timelike.

Once the connections have been determined, the impact on the gauge is discussed. The utilized derivatives from Eq. (21), Eq. (22), Eq. (23) and Eq. (24) are used for the dual grid as well. In the sense of events that are the next in the horizontal direction, the events that are spacelike connected are used. The timelike connections have been used to identify the events that are next in the vertical direction. Every other connection is included via the path integral. The used vectors are shown below.

$$z_{dual,1}(t) = \sqrt{\frac{1}{1+\frac{1}{3}}} \cdot \begin{pmatrix} t \\ \frac{1}{\sqrt{3}} \cdot t \end{pmatrix}, \; z_{dual,2}(t) = \sqrt{\frac{1}{1+\frac{1}{3}}} \cdot \begin{pmatrix} t \\ -\frac{1}{\sqrt{3}} \cdot t \end{pmatrix}$$

These vectors correspond to the ones that would be used in the flattened version. In this case, the spacelike property of the connection is clearly recognizable by looking at the vector. This vector is used since it is the only vector that could be used given the present information. Every other vector would be a wild guess since nothing can be said about the curvature to account for. To get a grasp of the curvature, the metric tensor is needed which is the property that is aimed to obtain. Therefore, $z_{dual,1}$ and $z_{dual,2}$ are the only fitting choices in this regard.

The whole set of equations that has been used to solve the dual grid can be seen in the appendix. Again, the procedure is the same as it was mentioned in the section "Demonstration of the formalism and methodology". After starting the calculation of the system, a solution for the metric is still not found and therefore the result is $g = \{\}$.

The above mentioned definition of which event is the next event in each direction is not responsible for the outcome. If the definition would be changed and only the clear horizontally connected events are used for the definition of the next horizontal event and only clear vertical connected events are used for the definition of the next vertical event, the result remains the same. This can be seen by rewriting the equations regarding the gauge in the used code.

Since "grid 2" possesses no solution, the same calculation will be performed on "grid 1". The dual grid version of "grid 1" is illustrated in Fig. 12.

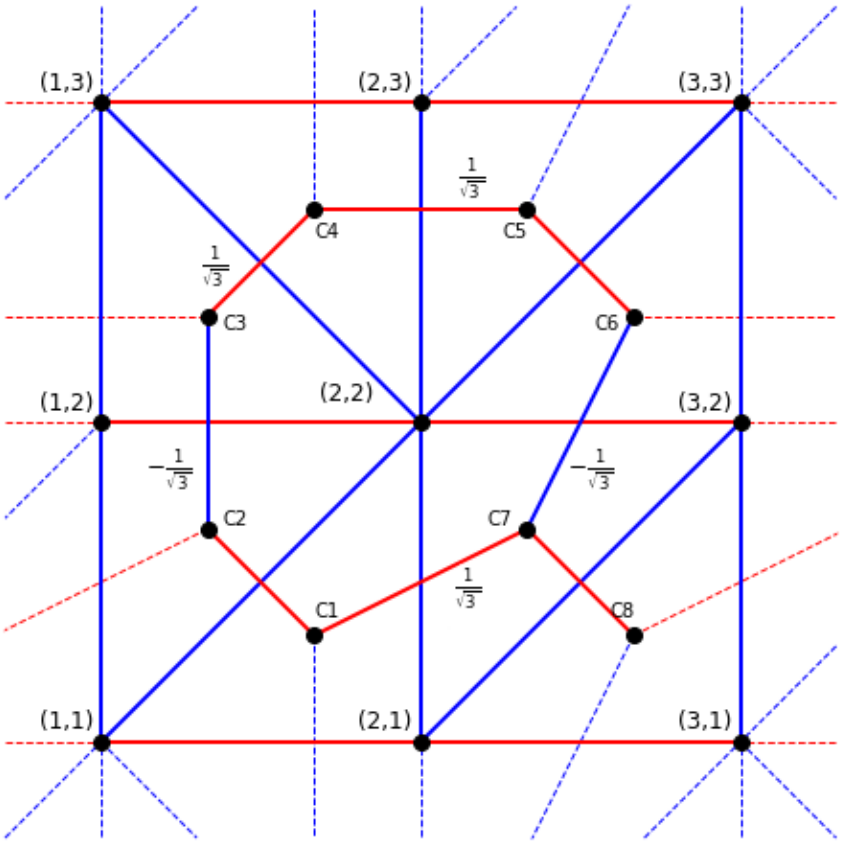

Figure 12: Corresponding dual grid to “grid 1”. The dotted lines mark a periodical extension of the grid. The centres in the figure only serve demonstration purposes for a better visualization of the structure. These centres do not show the true locations of the centres on the grid.

If the spacetime-structure of “grid 1” is flattened, it would look like the structure shown in Fig. 13. Similarly to the flattened version of “grid 2” the spacetime-structure collapses and a void emerges. In this case, both events (2,2) and (2,1) are torn apart and now constitute two events respectively.

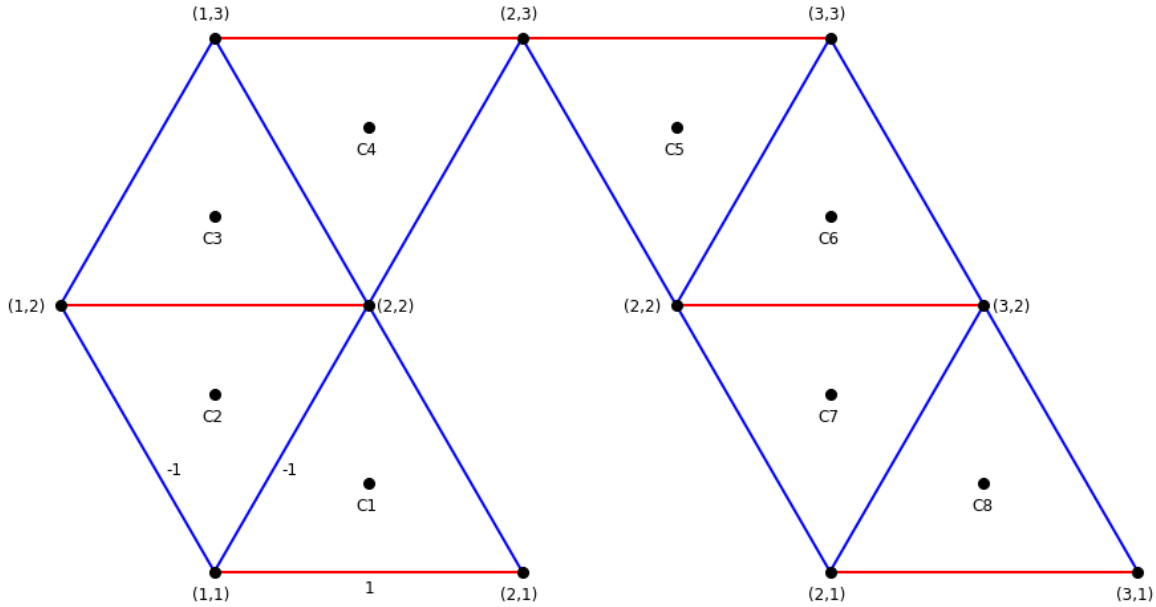

Figure 13: Flattened spacetime-structure of "grid 1". The corresponding centres of the triangles are denoted with a C.

It can be seen that the connection between C6 and C7 in Fig. 12 becomes distorted in comparison to the flattened state. This needs to be considered during the construction of a suited vector to describe the path. The connections between C6 and C7, C7 and C1, C5 and C8 as well as C2 to C8 will be treated in a different way. Since the arrangement of the triangles is different, the position of the centres C7 and C8 is different as well. In this regard it is not clear how to proceed. Therefore the same value of the other connections is applied to these connections, thereby being $\frac{1}{\sqrt{3}}$ without considering the sign. In this regard, the same argument for the dual grid of "grid 2" will be used here as well. Even if it seems counterintuitive in Fig. 12, the figure only serves as a helping tool in trying to visualize the problem at hand. The centres marked in the grid are not the true positions since the triangles are still equilateral and not rectangular. The question which vectors should be used is not trivial. For the connection between C1 and C7 the same vector $z_{dual,1}(t)$ will be used as in the case for the connection between C3 and C4. For the connection between C6 and C7 as well as C5 and C8 the vector $z_{dual,1}(t)$ will be used with flipped components therefore describing a timelike connection. This vector will be called $z_{dual,time}(t)$ and is shown below.

$$z_{dual,time}(t) = \sqrt{\frac{1}{1+\frac{1}{3}}} \cdot \begin{pmatrix} \frac{1}{\sqrt{3}} \cdot t \\ t \end{pmatrix}$$

The vectors and distances have been tested using the system of equations given in the appendix. The steps of the calculation were the same as for the dual grid of "grid 2". Aside from the vectors and distances, the connections and the gauge needed to be adjusted to suit "grid 1". After the calculation is completed, the result did not differ, which indicates that it is not possible to obtain the metric tensor. For the dual grid of "grid 1" the vectors can not be taken simply like it would be in Euclidean space like it was the case for "grid 2". In the flattened structure C6 would be directly on top of C7 and by periodical extension the nature of the connection between C8 and C5 seems to be spacelike, not timelike. Even the vectors of the Euclidean

case were tested together with changing the connection between C5 and C8 to a spacelike path. The vectors $z_{C6,C7}(t)$ and $z_{C1,C7}(t)$ are listed below, but could still not change the fact that no solution for the metric could be obtained. The vector for the geodesic connecting C5 and C8 would be $z_{dual,1}(t)$, but in the Euclidean case the length would be doubled.

$$z_{C6,C7}(t) = \begin{pmatrix} 0 \\ t \end{pmatrix}, z_{C1,C7}(t) = \frac{1}{\sqrt{1+\frac{3}{81}}} \begin{pmatrix} t \\ \frac{\sqrt{3}}{9} \cdot t \end{pmatrix}$$

The vector $z_{C1,C7}(t)$ is obtained by applying the same imaginary coordinate system to the flattened version of "grid 1". By comparing the positions that are obtained in the same way as for "grid 2", the connection by periodically extending the grid offers a clear vector for the connection between C1 and C7. This vector only needs to be modified so that the derivative of the vector is properly normalized. This way the distances between C1 and C7 of $\frac{\sqrt{21}}{3}$ and the distance between C5 and C8 of $\frac{2\sqrt{3}}{3}$ are obtained. However, the changed distances and vectors did not affect the result and $g = \{\}$ is still the only result that can be obtained. These vectors and distances are included in the code for the dual grid and even various combinations of the vectors have been tested. Every test has been performed for a spacelike and timelike version of the connection between C5 and C8.

Between "grid 1" and "grid 2" it can be argued that "grid 2" would be the only reasonable option to circumvent the occurrence of a singularity. In Fig. 10 it can be seen that every event is connected vertically and horizontally to another event as well as diagonally to one neighbouring event. There are no contradictions for a specific event. In the case of "grid 1" however, it is visible in Fig. 12 that there are some events that show the same contradictory properties as the event (2,2) in Fig. 3. The events C1, C2, C5, C6, C7 and C8 show all differently oriented geodesics to other events regardless of the nature of the connection. This is assumed to be the most likely reason why the dual grid approach failed for "grid 1". Regardless of the exact vectors that are used to describe the connection, the orientation of the vectors alone would be sufficient for a contradiction to arise.

As a result, the dual grid approach failed for both grid structures resulting in no solution for the metric tensor.

7 Discussions and Conclusions

The aim of this work was to properly test if it is possible to extract the metric out of a two dimensional grid in which only the distances are given. The problems of the contradiction and overdetermination that led to a singularity could not be resolved entirely. First, it was tested if the implementation of so-called "support events" would be a viable approach. It was shown that the introduction of "support events" is an approach able to obtain the metric tensor for spacetime-structures given in Fig. 6 and Fig. 7, though Fig. 6 is a structure without singularities. However, various approaches regarding the gauge conditions and connections led to multiple solutions for the grid in Fig. 6. The first presented solution of the metric achieved with the use of the code in the appendix is considered to be the most promising attempt in terms of the applied method. Especially the reduction of "support events" leading to a geodesic from SE1 to (2,2) worked out well for the grid of Fig. 6. For the grid in Fig. 7 only one approach was able to achieve results, which are rather surprising due to the negative sign of the metric element g^{12} for every event and "support event". The metric tensor for the spacetime-structures given in Fig. 5 and Fig. 8 could not be obtained using "support events". It is assumed that the solvability of the spacetime-structure is dependent on the symmetry of the structure to at least some degree.

The choice of the vector for the diagonal connection influences the components of the metric. However, reasonably chosen vectors still offer a solution to the problem at hand, which has been verified in the section "Demonstration of the formalism and methodology". The fixation on periodic boundary conditions might be troublesome as well regarding "support events". If a periodical continuation of the grid is considered, more "support events" need to be inserted to still obey the boundary conditions [10]. However, this is not considered to be the main issue regarding the approach. One problem could be the definition of the next vertical event in terms of the gauge. In Fig. 5 for example, the next vertical element to "SE1" would be "SE2", even if it is not as clear as for the regular events. This is also not the predominant problem at hand since the introduction of more events to give the "support events" such a clear connection in the vertical direction did not affect the result for the grid of Fig. 5. Without looking further into the problem why given definitions of the gauge conditions lead to certain conclusions and if the symmetry is truly an involved property or not, the approach of introducing "support events" is considered to be not fitting for truly resolving the issue of arising singularities.

The next approach covers the introduction of a dual grid in order to circumvent the singularity entirely. Figure 12 and Fig. 10 represent the dual grid of the two considered spacetime-structures. The change of events caused a change in the distances of the connections as well. After calculating the distances and taking the only reasonable vectors to describe the path integral, this approach failed too. For both considered grids no solution to the metric could be found. In Fig. 12 the distances create the illusion that the connection between C6 and C7 would be different than between C2 and C3, which is not the case. It can be seen in Fig. 13 that these connections are in fact of the same length. Thus, it becomes distorted due to the curvature of spacetime. The distances and the vectors can only be treated like in Euclidean space. If the Euclidean treatment is not correct, the problem is in itself contradictory. If some kind of curvature is required to properly set up the dual grid, this curvature would be needed to get a hold of the metric tensor, which is essential in describing the curvature. If the property is needed to obtain the property itself, the approach will inevitably fail. Assuming that this is not the case, the only

[10] See Fig. 5 for reference in comparison to Fig. 4.

viable choice leads to no conclusion. The question which vector is suited best to describe the geodesic between the centres C6 and C7 as well as C5 and C8 is still left unanswered. The vector $z_{dual,time}(t)$ is considered to be the most suitable vector in this case. Thus, the connection between C5 and C8 being of a timelike nature is most reasonable. The vector could also be described by a curve if information of the curvature is given. However, since the vector itself is required to obtain information of the curvature, the only reasonable choice is to try the vectors that have been tested. In addition, the solvability of a spacetime-structure by using a dual grid depends on the structure itself. The dual grid of "grid 2" in Fig. 10 would be fitting to circumvent the emergence of a singularity whereas the dual grid of "grid 1" in Fig. 12 only leads to the occurrence of another singularity therefore only shifting the problem.

For a future outlook this may be taken further and be extended into four dimensions to correctly translate the results of this work into the ongoing scientific field. It is possible that crucial aspects are missing regarding the implementation of "support events" and the construction of a dual grid that can only be perceived in higher dimensions. Considering the results of this thesis, the dual grid is not viewed to be a promising approach, whereas the introduction of "support events" could be promising if the research is deepened and the proposed questions are studied farther. However, if only one approach turns out to work in higher dimensions, reconstructing spacetime properties only out of information regarding the distances and structure may give rise to other possibilities still left uncovered up until now.

8 Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to UNIGRAZonline or TUGRAZonline is identical to the present bachelor's thesis.

Graz, September 24, 2024

Lukas Würger

References

[1] C. PERNER. Metric reconstruction with causal dynamical triangulation, 2023.

[2] R. LOLL. *Quantum gravity from causal dynamical triangulations: a review*. Classical and Quantum Gravity **37** (2020).
doi:/10.1088/1361-6382/ab57c7

[3] S. M. CARROLL. *An Introduction to to General Relativity SPACETIME and GEOMETRY*. Cambridge University Press, 2019.
doi:10.1017/9781108770385

[4] R. M. WALD. *General relativity*. The University of Chicago Press, Chicago, 1984.
doi:10.7208/chicago/9780226870373.001.0001

[5] J. A. W. CHARLES W. MISNER, KIP S. THORNE. *Gravitation*. W. H. FREEMAN AND COMPANY San Francisco, 1973.

[6] N. WOODHOUSE. *General relativity*. Springer-Verlag London Limited 2007, 2007.
doi:10.1007/s10714-007-0513-4

[7] A. ZEE. *Quantum Field Theory in a Nutshell, Second Edition*. Princeton University Press, 2010.

[8] A. MAAS. *The Fröhlich-Morchio-Strocchi mechanism and quantum gravity*. SciPost Physics **8** (2020).
doi:10.21468/SciPostPhys.8.4.051

[9] M. MARKL. Black Holes as Quantum Phenomena, 2020.

[10] A. A. MELE. *Introduction to Haar Measure Tools in Quantum Information: A Beginner's Tutorial*. Quantum **8** (2024) 1340.
doi:10.22331/q-2024-05-08-1340

[11] A. MAAS. Towards testing the Fröhlich-Morchio-Strocchi mechanism in quantum gravity. *Proceedings of 41st International Conference on High Energy physics — PoS(ICHEP2022)*, ICHEP2022. Sissa Medialab, 2022 S. 424.
doi:10.22323/1.414.0424
http://dx.doi.org/10.22323/1.414.0424

[12] T. REGGE. *GENERAL RELATIVITY WITHOUT COORDINATES*. Il Nuovo Cimento **19** (1961).
doi:10.1007/BF02733251

[13] R. R. CUZINATTO, C. A. M. DE MELO, C. N. DE SOUZA. Introduction to Regge Calculus for Gravitation, 2019.
https://arxiv.org/abs/1904.01966

[14] R. LOLL. *Discrete Approaches to Quantum Gravity in Four Dimensions*. Living Reviews in Relativity **1** (1998).
doi:10.12942/lrr-1998-13

[15] R. SORKIN. *Time-evolution problem in regge calculus*. Phys. Rev. D **12** (1975) 385. doi:10.1103/PhysRevD.12.385

[16] J. CHEEGER, W. MÜLLER, R. SCHRADER. Lattice gravity or Riemannian structure on piecewise linear spaces. *Unified Theories of Elementary Particles* (herausgegeben von P. BREITENLOHNER, H. P. DÜRR). Springer Berlin Heidelberg, Berlin, Heidelberg, 1982 S. 176 – 188.

[17] H. ROMER, M. ZAHRINGER. *Functional integration and the diffeomorphism group in Euclidean lattice quantum gravity*. Classical and Quantum Gravity **3** (1986) 897. doi:10.1088/0264-9381/3/5/019

[18] G. 'T HOOFT, M. J. G. VELTMAN. *One loop divergencies in the theory of gravitation*. Annales de l'institut Henri Poincaré. Section A, Physique Théorique **20** (1974) 69 .

[19] J. AMBJØRN, A. GÖRLICH, J. JURKIEWICZ, R. LOLL. *Nonperturbative quantum gravity*. Physics Reports **519** (2012) 127–210. doi:10.1016/j.physrep.2012.03.007

[20] R. LOLL, B. RUIJL. *Locally causal dynamical triangulations in two dimensions*. Physical Review D **92** (2015). doi:10.1103/physrevd.92.084002

[21] J. AMBJØRN, J. JURKIEWICZ, R. LOLL. *Quantum Gravity as Sum over Spacetimes*, S. 59–124. Springer Berlin Heidelberg, 2010. doi:10.1007/978-3-642-11897-5_2

Appendices

For the sake of a compact code with reasonable long names of variables, a simplification is introduced that convert the actual terms in just symbols. This simplification used for every single grid is given below:

$$g^{11} := \mathrm{x},\ g^{12} = g^{21} := \mathrm{y},\ g^{22} := \mathrm{z},$$
$$g_{11} := \mathrm{a},\ g_{12} = g_{21} := \mathrm{b},\ g_{22} := \mathrm{c}$$

The numeration of events is individual and differs depending on the grid. The events are numbered to make it easier to assign certain equations to certain events. The numeration for the grid in Fig. 1 and Fig. 2 is done in the following way:

$$(1,1) := 1,\ (1,2) := 2,\ (1,3) := 3,\ (2,1) := 4,\ (2,2) := 5,$$
$$(2,3) := 6,\ (3,1) := 7,\ (3,2) := 8,\ (3,3) := 9$$

The numeration of the events demonstrated in Fig. 4 and Fig. 5 is given below:

$$(1,1) := 1,\ (1,2) := 2,\ (1,3) := 3,\ (2,1) := 4,\ (2,2) := 5,$$
$$(2,3) := 6,\ (3,1) := 7,\ (3,2) := 8,\ (3,3) := 9,\ \mathrm{SE1} := 10,$$
$$\mathrm{SE2} := 11,\ \mathrm{SE3} := 12,\ \mathrm{SE4} := 13,\ \mathrm{SE5} := 14,\ \mathrm{SE6} := 15,$$
$$\mathrm{SE7} := 16,\ \mathrm{SE8} := 17,\ \mathrm{SE9} := 18,\ \mathrm{SE10} := 19,\ \mathrm{SE11} := 20,\ \mathrm{SE12} := 21$$

The events of the grid in Fig. 6 are numbered as follows:

$$(1,1) := 1,\ (1,2) := 2,\ (2,1) := 3,\ (2,2) := 4,$$
$$\mathrm{SE1} = \mathrm{SE4} := 5,\ \mathrm{SE2} = \mathrm{SE3} := 6$$

In Fig. 7 the numeration of the events is given by:

$$(1,1) := 1,\ (1,2) := 2,\ (2,1) := 3,\ (2,2) := 4,\ (3,1) := 5,\ (3,2) := 6,$$
$$\mathrm{SE1} = \mathrm{SE2} := 7,\ \mathrm{SE3} = \mathrm{SE4} := 8,\ \mathrm{SE5} = \mathrm{SE6} := 9,\ \mathrm{SE7} = \mathrm{SE8} := 10$$

The numeration of the events used for the dual grid of "grid 1" in Fig. 12 and the dual grid of "grid 2" in Fig. 10 is defined the following way:

$$\mathrm{C1} := 1,\ \mathrm{C2} := 2,\ \mathrm{C3} := 3,\ \mathrm{C4} := 4,$$
$$\mathrm{C5} := 5, \mathrm{C6} := 6,\ \mathrm{C7} := 7, \mathrm{C8} := 8$$

The simplification of the metric elements and the numeration of the events is used in the according code. The codes for each individual grid is listed below in the order they appear in the thesis.

```
In[1]:= (*code for a grid with 9 events and 1 cross connection*)

In[2]:= zhorizontal = {t, 0} (*vector for a horizontal connection between events*)

Out[2]= {t, 0}

In[3]:= zvertical = {0, t} (*vector for a vertical connection between events*)

Out[3]= {0, t}

In[4]:= zdiag = {t, t} * (1 / Sqrt[2]) (*vector for the connection between (2,2) and (3,3)*)
```

Out[4]= $\left\{\frac{t}{\sqrt{2}}, \frac{t}{\sqrt{2}}\right\}$

```
In[5]:= zdiagflat = {t , Sqrt[3] * t} * 1 / 2 (*alternative vector for the diagonal connection*)
```

Out[5]= $\left\{\frac{t}{2}, \frac{\sqrt{3}\, t}{2}\right\}$

```
In[6]:= (*take the derivative of all vectors*)

In[7]:= dtzhorizontal = D[zhorizontal, t]

Out[7]= {1, 0}

In[8]:= dtzvertical = D[zvertical, t]

Out[8]= {0, 1}

In[9]:= dtzdiag = D[zdiag, t]
```

Out[9]= $\left\{\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right\}$

```
In[10]:= dtzdiagflat = D[zdiagflat, t]
```

Out[10]=

$\left\{\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}$

```
In[11]:= (*set the distances for the connections*)

In[12]:= stimelike = -1

Out[12]= -1

In[13]:= sspacelike = 1

Out[13]= 1

In[14]:= (*set up the equations for the Haywood gauge*)

In[15]:= gauge1 := x1 * a4 - x1 * a1 + y1 * a2 - y1 * a1 + y1 * b2 - y1 * b1 + y1 * c4 - y1 * c1 +
            y1 * b4 - y1 * b1 + z1 * c2 - z1 * c1 + z1 * b2 - z1 * b1 + x1 * b4 - x1 * b1 == 0

In[16]:= gauge2 := x2 * a5 - x2 * a2 + y2 * a3 - y2 * a2 + y2 * b3 - y2 * b2 + y2 * c5 - y2 * c2 +
            y2 * b5 - y2 * b2 + z2 * c3 - z2 * c2 + z2 * b3 - z2 * b2 + x2 * b5 - x2 * b2 == 0
```

```
In[17]:= gauge3 := x3 * a6 - x3 * a3 + y3 * a2 - y3 * a3 + y3 * b2 - y3 * b3 + y3 * c6 - y3 * c3 +
    y3 * b6 - y3 * b3 + z3 * c2 - z3 * c3 + z3 * b2 - z3 * b3 + x3 * b6 - x3 * b3 == 0

In[18]:= gauge4 := x4 * a7 - x4 * a4 + y4 * a5 - y4 * a4 + y4 * b5 - y4 * b4 + y4 * c7 - y4 * c4 +
    y4 * b7 - y4 * b4 + z4 * c5 - z4 * c4 + z4 * b5 - z4 * b4 + x4 * b7 - x4 * b4 == 0

In[19]:= gauge5 := x5 * a8 - x5 * a5 + y5 * a6 - y5 * a5 + y5 * b6 - y5 * b5 + y5 * c8 - y5 * c5 +
    y5 * b8 - y5 * b5 + z5 * c6 - z5 * c5 + z5 * b6 - z5 * b5 + x5 * b8 - x5 * b5 == 0

In[20]:= gauge6 := x6 * a9 - x6 * a6 + y6 * a5 - y6 * a6 + y6 * b5 - y6 * b6 + y6 * c9 - y6 * c6 +
    y6 * b9 - y6 * b6 + z6 * c5 - z6 * c6 + z6 * b5 - z6 * b6 + x6 * b9 - x6 * b6 == 0

In[21]:= gauge7 := x7 * a4 - x7 * a7 + y7 * a8 - y7 * a7 + y7 * b8 - y7 * b7 + y7 * c4 - y7 * c7 +
    y7 * b4 - y7 * b7 + z7 * c8 - z7 * c7 + z7 * b8 - z7 * b7 + x7 * b4 - x7 * b7 == 0

In[22]:= gauge8 := x8 * a5 - x8 * a8 + y8 * a9 - y8 * a8 + y8 * b9 - y8 * b8 + y8 * c5 - y8 * c8 +
    y8 * b5 - y8 * b8 + z8 * c9 - z8 * c8 + z8 * b9 - z8 * b8 + x8 * b5 - x8 * b8 == 0

In[23]:= gauge9 := x9 * a6 - x9 * a9 + y9 * a8 - y9 * a9 + y9 * b8 - y9 * b9 + y9 * c6 - y9 * c9 +
    y9 * b6 - y9 * b9 + z9 * c8 - z9 * c9 + z9 * b8 - z9 * b9 + x9 * b6 - x9 * b9 == 0

In[24]:= (*define the timelike and spacelike connections of each event*)

In[25]:= pathvertical1 := stimelike == x1 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y1 * dtzvertical[[1]] * dtzvertical[[2]] + z1 * dtzvertical[[2]] * dtzvertical[[2]]

In[26]:= pathvertical2 := stimelike == x2 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y2 * dtzvertical[[1]] * dtzvertical[[2]] + z2 * dtzvertical[[2]] * dtzvertical[[2]]

In[27]:= pathvertical3 := stimelike == x3 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y3 * dtzvertical[[1]] * dtzvertical[[2]] + z3 * dtzvertical[[2]] * dtzvertical[[2]]

In[28]:= pathvertical4 := stimelike == x4 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y4 * dtzvertical[[1]] * dtzvertical[[2]] + z4 * dtzvertical[[2]] * dtzvertical[[2]]

In[29]:= pathvertical5 := stimelike == x5 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y5 * dtzvertical[[1]] * dtzvertical[[2]] + z5 * dtzvertical[[2]] * dtzvertical[[2]]

In[30]:= pathvertical6 := stimelike == x6 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y6 * dtzvertical[[1]] * dtzvertical[[2]] + z6 * dtzvertical[[2]] * dtzvertical[[2]]

In[31]:= pathvertical7 := stimelike == x7 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y7 * dtzvertical[[1]] * dtzvertical[[2]] + z7 * dtzvertical[[2]] * dtzvertical[[2]]

In[32]:= pathvertical8 := stimelike == x8 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y8 * dtzvertical[[1]] * dtzvertical[[2]] + z8 * dtzvertical[[2]] * dtzvertical[[2]]

In[33]:= pathvertical9 := stimelike == x9 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y9 * dtzvertical[[1]] * dtzvertical[[2]] + z9 * dtzvertical[[2]] * dtzvertical[[2]]

In[34]:= pathhorizontal1 := sspacelike == x1 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y1 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z1 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[35]:= pathhorizontal2 := sspacelike == x2 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y2 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z2 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[36]:= pathhorizontal3 := sspacelike == x3 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y3 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z3 * dtzhorizontal[[2]] * dtzhorizontal[[2]]
```

```
In[37]:= pathhorizontal4 := sspacelike == x4 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y4 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z4 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[38]:= pathhorizontal5 := sspacelike == x5 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y5 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z5 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[39]:= pathhorizontal6 := sspacelike == x6 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y6 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z6 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[40]:= pathhorizontal7 := sspacelike == x7 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y7 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z7 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[41]:= pathhorizontal8 := sspacelike == x8 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y8 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z8 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[42]:= pathhorizontal9 := sspacelike == x9 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y9 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z9 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[43]:= pathdiag5 := stimelike == x5 * dtzdiag[[1]] * dtzdiag[[1]] +
    2 * y5 * dtzdiag[[1]] * dtzdiag[[2]] + z5 * dtzdiag[[2]] * dtzdiag[[2]]

In[44]:= pathdiag9 := stimelike == x9 * dtzdiag[[1]] * dtzdiag[[1]] +
    2 * y9 * dtzdiag[[1]] * dtzdiag[[2]] + z9 * dtzdiag[[2]] * dtzdiag[[2]]

In[45]:= pathdiag1 := stimelike == x1 * dtzdiag[[1]] * dtzdiag[[1]] +
    2 * y1 * dtzdiag[[1]] * dtzdiag[[2]] + z1 * dtzdiag[[2]] * dtzdiag[[2]]

In[46]:= pathdiag3 := stimelike == x3 * dtzdiag[[1]] * dtzdiag[[1]] +
    2 * y3 * dtzdiag[[1]] * dtzdiag[[2]] + z3 * dtzdiag[[2]] * dtzdiag[[2]]

In[47]:= pathdiag7 := stimelike == x7 * dtzdiag[[1]] * dtzdiag[[1]] +
    2 * y7 * dtzdiag[[1]] * dtzdiag[[2]] + z7 * dtzdiag[[2]] * dtzdiag[[2]]

In[48]:= (*solve the system of equations by including all of the equations*)

In[49]:= (*the equations regarding the invertibility condition are included
    directly into the system because of the sheer number of equations*)

In[50]:= (*the solutions are restricted to only
    real solutions since the metric is ought to be real*)
```

In[51]:=
```
NSolve[{gauge1, gauge2, gauge3, gauge4, gauge5, gauge6, gauge7,
  gauge8, gauge9, x1 * a1 + y1 * b1 == 1, x1 * b1 + y1 * c1 == 0, y1 * a1 + z1 * b1 == 0,
  y1 * b1 + z1 * c1 == 1, x2 * a2 + y2 * b2 == 1, x2 * b2 + y2 * c2 == 0, y2 * a2 + z2 * b2 == 0,
  y2 * b2 + z2 * c2 == 1, x3 * a3 + y3 * b3 == 1, x3 * b3 + y3 * c3 == 0, y3 * a3 + z3 * b3 == 0,
  y3 * b3 + z3 * c3 == 1, x4 * a4 + y4 * b4 == 1, x4 * b4 + y4 * c4 == 0, y4 * a4 + z4 * b4 == 0,
  y4 * b4 + z4 * c4 == 1, x5 * a5 + y5 * b5 == 1, x5 * b5 + y5 * c5 == 0, y5 * a5 + z5 * b5 == 0,
  y5 * b5 + z5 * c5 == 1, x6 * a6 + y6 * b6 == 1, x6 * b6 + y6 * c6 == 0, y6 * a6 + z6 * b6 == 0,
  y6 * b6 + z6 * c6 == 1, x7 * a7 + y7 * b7 == 1, x7 * b7 + y7 * c7 == 0, y7 * a7 + z7 * b7 == 0,
  y7 * b7 + z7 * c7 == 1, x8 * a8 + y8 * b8 == 1, x8 * b8 + y8 * c8 == 0, y8 * a8 + z8 * b8 == 0,
  y8 * b8 + z8 * c8 == 1, x9 * a9 + y9 * b9 == 1, x9 * b9 + y9 * c9 == 0, y9 * a9 + z9 * b9 == 0,
  y9 * b9 + z9 * c9 == 1, pathvertical1, pathvertical2, pathvertical3, pathvertical4,
  pathvertical5, pathvertical6, pathvertical7, pathvertical8, pathvertical9,
  pathhorizontal1, pathhorizontal2, pathhorizontal3, pathhorizontal4,
  pathhorizontal5, pathhorizontal6, pathhorizontal7, pathhorizontal8,
  pathhorizontal9, pathdiag1, pathdiag3, pathdiag5, pathdiag7, pathdiag9},
 {x1, x2, x3, x4, x5, x6, x7, x8, x9, y1, y2, y3, y4, y5, y6, y7, y8, y9, z1,
  z2, z3, z4, z5, z6, z7, z8, z9, a1, a2, a3, a4, a5, a6, a7, a8, a9, b1, b2,
  b3, b4, b5, b6, b7, b8, b9, c1, c2, c3, c4, c5, c6, c7, c8, c9}, Reals]
```

Out[51]=
```
{{x1 → 1., x2 → 1., x3 → 1., x4 → 1., x5 → 1., x6 → 1., x7 → 1., x8 → 1., x9 → 1.,
  y1 → -1., y2 → -1., y3 → -1., y4 → -1., y5 → -1., y6 → -1., y7 → -1., y8 → -1.,
  y9 → -1., z1 → -1., z2 → -1., z3 → -1., z4 → -1., z5 → -1., z6 → -1., z7 → -1.,
  z8 → -1., z9 → -1., a1 → 0.5, a2 → 0.5, a3 → 0.5, a4 → 0.5, a5 → 0.5, a6 → 0.5,
  a7 → 0.5, a8 → 0.5, a9 → 0.5, b1 → -0.5, b2 → -0.5, b3 → -0.5, b4 → -0.5,
  b5 → -0.5, b6 → -0.5, b7 → -0.5, b8 → -0.5, b9 → -0.5, c1 → -0.5, c2 → -0.5,
  c3 → -0.5, c4 → -0.5, c5 → -0.5, c6 → -0.5, c7 → -0.5, c8 → -0.5, c9 → -0.5}}
```

In[52]:=
```
(*for the same grid with 4 cross connections instead of 1 in the same direction,
the following equations need to be added:*)
```

In[53]:=
```
(*pathdiag2 := stimelike == x2*dtzdiag[[1]]*dtzdiag[[1]] +
   2*y2*dtzdiag[[1]]*dtzdiag[[2]] + z2*dtzdiag[[2]]*dtzdiag[[2]]*)
```

In[54]:=
```
(*pathdiag4 := stimelike == x4*dtzdiag[[1]]*dtzdiag[[1]] +
   2*y4*dtzdiag[[1]]*dtzdiag[[2]] + z4*dtzdiag[[2]]*dtzdiag[[2]]*)
```

In[55]:=
```
(*pathdiag6 := stimelike == x6*dtzdiag[[1]]*dtzdiag[[1]] +
   2*y6*dtzdiag[[1]]*dtzdiag[[2]] + z6*dtzdiag[[2]]*dtzdiag[[2]]*)
```

In[56]:=
```
(*pathdiag8 := stimelike == x8*dtzdiag[[1]]*dtzdiag[[1]] +
   2*y8*dtzdiag[[1]]*dtzdiag[[2]] + z8*dtzdiag[[2]]*dtzdiag[[2]]*)
```

```
(*code for a grid with 9 events, 4 cross connections and 12 support events; grid 1*)

In[ ]:= alpha = 0.01
         (*set the value for the distance between event (1,1) and support event SE1*)
Out[ ]=
0.01

beta = 1 - 2 * alpha
   (*set the value for the distance between event (1,2) and support event SE4*)
Out[ ]=
0.98

In[ ]:= zhorizontal = {t, 0} (*vector for a horizontal connection between events*)
Out[ ]=
{t, 0}

In[ ]:= zvertical = {0, t} (*vector for a vertical connection between events*)
Out[ ]=
{0, t}

tan = 1 / (1 - 2 * alpha) (*slope of vector zdiagSE depending on alpha*)
Out[ ]=
1.02041

tanSE34 = 1 / (1 - 4 * alpha) (*slope of vector zdiag34 depending on alpha*)
Out[ ]=
1.04167

zdiagSE = {t, tan * t} * Sqrt[1 / (1 + tan^2)]
  (*vector for a diagonal connection between support events*)
Out[ ]=
{0.699929 t, 0.714213 t}

zdiag34 = {t, - tanSE34 * t} * Sqrt[1 / (1 + tanSE34^2)]
  (*vector for the diagonal connection between SE3 and SE4*)
Out[ ]=
{0.692532 t, -0.721387 t}

In[ ]:= (*take the derivative of all vectors*)
dtzhorizontal = D[zhorizontal, t]
Out[ ]=
{1, 0}

In[ ]:= dtzvertical = D[zvertical, t]
Out[ ]=
{0, 1}

In[ ]:= dtzdiagSE = D[zdiagSE, t]
Out[ ]=
{0.699929, 0.714213}

In[ ]:= dtzdiag34 = D[zdiag34, t]
Out[ ]=
{0.692532, -0.721387}
```

```
In[ ]:= (*set the distances for the connections*)

In[ ]:= stimelike = -1

Out[ ]=
        -1

In[ ]:= sspacelike = 1

Out[ ]=
        1

In[ ]:= (*set up the equations for the Haywood gauge*)

In[ ]:= gauge1 := x1 * a10 - x1 * a1 + y1 * a2 - y1 * a1 + y1 * b2 - y1 * b1 + y1 * c10 - y1 * c1 +
           y1 * b10 - y1 * b1 + z1 * c2 - z1 * c1 + z1 * b2 - z1 * b1 + x1 * b10 - x1 * b1 == 0

In[ ]:= gauge2 := x2 * a13 - x2 * a2 + y2 * a3 - y2 * a2 + y2 * b3 - y2 * b2 + y2 * c13 - y2 * c2 +
           y2 * b13 - y2 * b2 + z2 * c3 - z2 * c2 + z2 * b3 - z2 * b2 + x2 * b13 - x2 * b2 == 0

In[ ]:= gauge3 := x3 * a19 - x3 * a3 + y3 * a2 - y3 * a3 + y3 * b2 - y3 * b3 + y3 * c19 - y3 * c3 +
           y3 * b19 - y3 * b3 + z3 * c2 - z3 * c3 + z3 * b2 - z3 * b3 + x3 * b19 - x3 * b3 == 0

In[ ]:= gauge4 := x4 * a14 - x4 * a4 + y4 * a5 - y4 * a4 + y4 * b5 - y4 * b4 + y4 * c14 - y4 * c4 +
           y4 * b14 - y4 * b4 + z4 * c5 - z4 * c4 + z4 * b5 - z4 * b4 + x4 * b14 - x4 * b4 == 0

In[ ]:= gauge5 := x5 * a16 - x5 * a5 + y5 * a6 - y5 * a5 + y5 * b6 - y5 * b5 + y5 * c16 - y5 * c5 +
           y5 * b16 - y5 * b5 + z5 * c6 - z5 * c5 + z5 * b6 - z5 * b5 + x5 * b16 - x5 * b5 == 0

In[ ]:= gauge6 := x6 * a21 - x6 * a6 + y6 * a5 - y6 * a6 + y6 * b5 - y6 * b6 + y6 * c21 - y6 * c6 +
           y6 * b21 - y6 * b6 + z6 * c5 - z6 * c6 + z6 * b5 - z6 * b6 + x6 * b21 - x6 * b6 == 0

In[ ]:= gauge7 := x7 * a10 - x7 * a7 + y7 * a8 - y7 * a7 + y7 * b8 - y7 * b7 + y7 * c10 - y7 * c7 +
           y7 * b10 - y7 * b7 + z7 * c8 - z7 * c7 + z7 * b8 - z7 * b7 + x7 * b10 - x7 * b7 == 0

In[ ]:= gauge8 := x8 * a13 - x8 * a8 + y8 * a9 - y8 * a8 + y8 * b9 - y8 * b8 + y8 * c13 - y8 * c8 +
           y8 * b13 - y8 * b8 + z8 * c9 - z8 * c8 + z8 * b9 - z8 * b8 + x8 * b13 - x8 * b8 == 0

In[ ]:= gauge9 := x9 * a19 - x9 * a9 + y9 * a8 - y9 * a9 + y9 * b8 - y9 * b9 + y9 * c19 - y9 * c9 +
           y9 * b19 - y9 * b9 + z9 * c8 - z9 * c9 + z9 * b8 - z9 * b9 + x9 * b19 - x9 * b9 == 0

In[ ]:= gauge10 := x10 * a18 - x10 * a10 + y10 * a11 - y10 * a10 +
           y10 * b11 - y10 * b10 + y10 * c18 - y10 * c10 + y10 * b18 - y10 * b10 +
           z10 * c11 - z10 * c10 + z10 * b11 - z10 * b10 + x10 * b18 - x10 * b10 == 0

In[ ]:= gauge11 := x11 * a5 - x11 * a11 + y11 * a11 - y11 * a10 +
           y11 * b11 - y11 * b10 + y11 * c5 - y11 * c11 + y11 * b5 - y11 * b11 +
           z11 * c11 - z11 * c10 + z11 * b11 - z11 * b10 + x11 * b5 - x11 * b11 == 0

In[ ]:= gauge12 := x12 * a6 - x12 * a12 + y12 * a12 - y12 * a13 +
           y12 * b12 - y12 * b13 + y12 * c6 - y12 * c12 + y12 * b6 - y12 * b12 +
           z12 * c12 - z12 * c13 + z12 * b12 - z12 * b13 + x12 * b6 - x12 * b12 == 0

In[ ]:= gauge13 := x13 * a11 - x13 * a13 + y13 * a12 - y13 * a13 +
           y13 * b12 - y13 * b13 + y13 * c11 - y13 * c13 + y13 * b11 - y13 * b13 +
           z13 * c12 - z13 * c13 + z13 * b12 - z13 * b13 + x13 * b11 - x13 * b13 == 0
```

```
In[ ]:= gauge14 := x14 * a20 - x14 * a14 + y14 * a15 - y14 * a14 +
          y14 * b15 - y14 * b14 + y14 * c20 - y14 * c14 + y14 * b20 - y14 * b14 +
          z14 * c15 - z14 * c14 + z14 * b15 - z14 * b14 + x14 * b20 - x14 * b14 == 0

In[ ]:= gauge15 := x15 * a8 - x15 * a15 + y15 * a15 - y15 * a14 +
          y15 * b15 - y15 * b14 + y15 * c8 - y15 * c15 + y15 * b8 - y15 * b15 +
          z15 * c15 - z15 * c14 + z15 * b15 - z15 * b14 + x15 * b8 - x15 * b15 == 0

In[ ]:= gauge16 := x16 * a15 - x16 * a16 + y16 * a17 - y16 * a16 +
          y16 * b17 - y16 * b16 + y16 * c15 - y16 * c16 + y16 * b15 - y16 * b16 +
          z16 * c17 - z16 * c16 + z16 * b17 - z16 * b16 + x16 * b15 - x16 * b16 == 0

In[ ]:= gauge17 := x17 * a9 - x17 * a17 + y17 * a17 - y17 * a16 +
          y17 * b17 - y17 * b16 + y17 * c9 - y17 * c17 + y17 * b9 - y17 * b17 +
          z17 * c17 - z17 * c16 + z17 * b17 - z17 * b16 + x17 * b9 - x17 * b17 == 0

In[ ]:= gauge18 := x18 * a4 - x18 * a18 + y18 * a18 - y18 * a13 +
          y18 * b18 - y18 * b13 + y18 * c4 - y18 * c18 + y18 * b4 - y18 * b18 +
          z18 * c18 - z18 * c13 + z18 * b18 - z18 * b13 + x18 * b4 - x18 * b18 == 0

In[ ]:= gauge19 := x19 * a12 - x19 * a19 + y19 * a11 - y19 * a19 +
          y19 * b11 - y19 * b19 + y19 * c12 - y19 * c19 + y19 * b12 - y19 * b19 +
          z19 * c11 - z19 * c19 + z19 * b11 - z19 * b19 + x19 * b12 - x19 * b19 == 0

In[ ]:= gauge20 := x20 * a7 - x20 * a20 + y20 * a20 - y20 * a16 +
          y20 * b20 - y20 * b16 + y20 * c7 - y20 * c20 + y20 * b7 - y20 * b20 +
          z20 * c20 - z20 * c16 + z20 * b20 - z20 * b16 + x20 * b7 - x20 * b20 == 0

In[ ]:= gauge21 := x21 * a17 - x21 * a21 + y21 * a15 - y21 * a21 +
          y21 * b15 - y21 * b21 + y21 * c17 - y21 * c21 + y21 * b17 - y21 * b21 +
          z21 * c15 - z21 * c21 + z21 * b15 - z21 * b21 + x21 * b17 - x21 * b21 == 0

In[ ]:= (*define the timelike and spacelike connections of each event*)

In[ ]:= pathhorizontal1 := alpha == x1 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
          2 * y1 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z1 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal2 := beta == x2 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
          2 * y2 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z2 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal3 := alpha == x3 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
          2 * y3 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z3 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal4 := alpha == x4 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
          2 * y4 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z4 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal5 := alpha == x5 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
          2 * y5 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z5 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal6 := alpha == x6 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
          2 * y6 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z6 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal7 := alpha == x7 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
          2 * y7 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z7 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal8 := beta == x8 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
          2 * y8 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z8 * dtzhorizontal[[2]] * dtzhorizontal[[2]]
```

```
In[ ]:= pathhorizontal9 := alpha == x9 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
          2 * y9 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z9 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal10 :=
         alpha == x10 * dtzhorizontal[[1]] * dtzhorizontal[[1]] + 2 * y10 * dtzhorizontal[[1]] *
            dtzhorizontal[[2]] + z10 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal11 :=
         alpha == x11 * dtzhorizontal[[1]] * dtzhorizontal[[1]] + 2 * y11 * dtzhorizontal[[1]] *
            dtzhorizontal[[2]] + z11 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal12 :=
         beta == x12 * dtzhorizontal[[1]] * dtzhorizontal[[1]] + 2 * y12 * dtzhorizontal[[1]] *
            dtzhorizontal[[2]] + z12 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal13 :=
         alpha == x13 * dtzhorizontal[[1]] * dtzhorizontal[[1]] + 2 * y13 * dtzhorizontal[[1]] *
            dtzhorizontal[[2]] + z13 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal14 :=
         beta == x14 * dtzhorizontal[[1]] * dtzhorizontal[[1]] + 2 * y14 * dtzhorizontal[[1]] *
            dtzhorizontal[[2]] + z14 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal15 :=
         alpha == x15 * dtzhorizontal[[1]] * dtzhorizontal[[1]] + 2 * y15 * dtzhorizontal[[1]] *
            dtzhorizontal[[2]] + z15 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal16 :=
         beta == x16 * dtzhorizontal[[1]] * dtzhorizontal[[1]] + 2 * y16 * dtzhorizontal[[1]] *
            dtzhorizontal[[2]] + z16 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal17 :=
         alpha == x17 * dtzhorizontal[[1]] * dtzhorizontal[[1]] + 2 * y17 * dtzhorizontal[[1]] *
            dtzhorizontal[[2]] + z17 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal18 :=
         beta == x18 * dtzhorizontal[[1]] * dtzhorizontal[[1]] + 2 * y18 * dtzhorizontal[[1]] *
            dtzhorizontal[[2]] + z18 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal19 :=
         alpha == x19 * dtzhorizontal[[1]] * dtzhorizontal[[1]] + 2 * y19 * dtzhorizontal[[1]] *
            dtzhorizontal[[2]] + z19 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal20 :=
         alpha == x20 * dtzhorizontal[[1]] * dtzhorizontal[[1]] + 2 * y20 * dtzhorizontal[[1]] *
            dtzhorizontal[[2]] + z20 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathhorizontal21 :=
         beta == x21 * dtzhorizontal[[1]] * dtzhorizontal[[1]] + 2 * y21 * dtzhorizontal[[1]] *
            dtzhorizontal[[2]] + z21 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[ ]:= pathvertical1 := stimelike == x1 * dtzvertical[[1]] * dtzvertical[[1]] +
          2 * y1 * dtzvertical[[1]] * dtzvertical[[2]] + z1 * dtzvertical[[2]] * dtzvertical[[2]]

In[ ]:= pathvertical2 := stimelike == x2 * dtzvertical[[1]] * dtzvertical[[1]] +
          2 * y2 * dtzvertical[[1]] * dtzvertical[[2]] + z2 * dtzvertical[[2]] * dtzvertical[[2]]
```

```
In[ ]:= pathvertical3 := stimelike == x3 * dtzvertical[[1]] * dtzvertical[[1]] +
          2 * y3 * dtzvertical[[1]] * dtzvertical[[2]] + z3 * dtzvertical[[2]] * dtzvertical[[2]]

In[ ]:= pathvertical4 := stimelike == x4 * dtzvertical[[1]] * dtzvertical[[1]] +
          2 * y4 * dtzvertical[[1]] * dtzvertical[[2]] + z4 * dtzvertical[[2]] * dtzvertical[[2]]

In[ ]:= pathvertical5 := stimelike == x5 * dtzvertical[[1]] * dtzvertical[[1]] +
          2 * y5 * dtzvertical[[1]] * dtzvertical[[2]] + z5 * dtzvertical[[2]] * dtzvertical[[2]]

In[ ]:= pathvertical6 := stimelike == x6 * dtzvertical[[1]] * dtzvertical[[1]] +
          2 * y6 * dtzvertical[[1]] * dtzvertical[[2]] + z6 * dtzvertical[[2]] * dtzvertical[[2]]

In[ ]:= pathvertical7 := stimelike == x7 * dtzvertical[[1]] * dtzvertical[[1]] +
          2 * y7 * dtzvertical[[1]] * dtzvertical[[2]] + z7 * dtzvertical[[2]] * dtzvertical[[2]]

In[ ]:= pathvertical8 := stimelike == x8 * dtzvertical[[1]] * dtzvertical[[1]] +
          2 * y8 * dtzvertical[[1]] * dtzvertical[[2]] + z8 * dtzvertical[[2]] * dtzvertical[[2]]

In[ ]:= pathvertical9 := stimelike == x9 * dtzvertical[[1]] * dtzvertical[[1]] +
          2 * y9 * dtzvertical[[1]] * dtzvertical[[2]] + z9 * dtzvertical[[2]] * dtzvertical[[2]]

In[ ]:= pathdiag10 := stimelike == x10 * dtzdiagSE[[1]] * dtzdiagSE[[1]] +
          2 * y10 * dtzdiagSE[[1]] * dtzdiagSE[[2]] + z10 * dtzdiagSE[[2]] * dtzdiagSE[[2]]

In[ ]:= pathdiag11 := stimelike == x11 * dtzdiagSE[[1]] * dtzdiagSE[[1]] +
          2 * y11 * dtzdiagSE[[1]] * dtzdiagSE[[2]] + z11 * dtzdiagSE[[2]] * dtzdiagSE[[2]]

In[ ]:= pathdiag14 := stimelike == x14 * dtzdiagSE[[1]] * dtzdiagSE[[1]] +
          2 * y14 * dtzdiagSE[[1]] * dtzdiagSE[[2]] + z14 * dtzdiagSE[[2]] * dtzdiagSE[[2]]

In[ ]:= pathdiag15 := stimelike == x15 * dtzdiagSE[[1]] * dtzdiagSE[[1]] +
          2 * y15 * dtzdiagSE[[1]] * dtzdiagSE[[2]] + z15 * dtzdiagSE[[2]] * dtzdiagSE[[2]]

In[ ]:= pathdiag16 := stimelike == x16 * dtzdiagSE[[1]] * dtzdiagSE[[1]] +
          2 * y16 * dtzdiagSE[[1]] * dtzdiagSE[[2]] + z16 * dtzdiagSE[[2]] * dtzdiagSE[[2]]

In[ ]:= pathdiag17 := stimelike == x17 * dtzdiagSE[[1]] * dtzdiagSE[[1]] +
          2 * y17 * dtzdiagSE[[1]] * dtzdiagSE[[2]] + z17 * dtzdiagSE[[2]] * dtzdiagSE[[2]]

In[ ]:= pathdiag19 := stimelike == x19 * dtzdiagSE[[1]] * dtzdiagSE[[1]] +
          2 * y19 * dtzdiagSE[[1]] * dtzdiagSE[[2]] + z19 * dtzdiagSE[[2]] * dtzdiagSE[[2]]

In[ ]:= pathdiag20 := stimelike == x20 * dtzdiagSE[[1]] * dtzdiagSE[[1]] +
          2 * y20 * dtzdiagSE[[1]] * dtzdiagSE[[2]] + z20 * dtzdiagSE[[2]] * dtzdiagSE[[2]]

        pathdiag21 := stimelike == x21 * dtzdiagSE[[1]] * dtzdiagSE[[1]] +
          2 * y21 * dtzdiagSE[[1]] * dtzdiagSE[[2]] + z21 * dtzdiagSE[[2]] * dtzdiagSE[[2]]


In[ ]:= pathdiag12 := stimelike == x12 * dtzdiag34[[1]] * dtzdiag34[[1]] +
          2 * y12 * dtzdiag34[[1]] * dtzdiag34[[2]] + z12 * dtzdiag34[[2]] * dtzdiag34[[2]]

In[ ]:= pathdiag13 := stimelike == x13 * dtzdiag34[[1]] * dtzdiag34[[1]] +
          2 * y13 * dtzdiag34[[1]] * dtzdiag34[[2]] + z13 * dtzdiag34[[2]] * dtzdiag34[[2]]

In[ ]:= pathdiag18 := stimelike == x18 * dtzdiag34[[1]] * dtzdiag34[[1]] +
          2 * y18 * dtzdiag34[[1]] * dtzdiag34[[2]] + z18 * dtzdiag34[[2]] * dtzdiag34[[2]]
```

In[ ]:=
```
(*solve the system of equations by including all of the equations*)
(*the equations regarding the invertibility condition are included
 directly into the system because of the sheer number of equations*)
```

In[ ]:=
```
(*the solutions are restricted to only
 real solutions since the metric is ought to be real*)
```

In[ ]:=
```
NSolve[{gauge1, gauge2, gauge3, gauge4, gauge5, gauge6, gauge7, gauge8,
  gauge9, gauge10, gauge11, gauge12, gauge13, gauge14, gauge15, gauge16,
  gauge17, gauge18, gauge19, gauge20, gauge21, x1 * a1 + y1 * b1 == 1,
  x1 * b1 + y1 * c1 == 0, y1 * a1 + z1 * b1 == 0, y1 * b1 + z1 * c1 == 1,
  x2 * a2 + y2 * b2 == 1, x2 * b2 + y2 * c2 == 0, y2 * a2 + z2 * b2 == 0,
  y2 * b2 + z2 * c2 == 1, x3 * a3 + y3 * b3 == 1, x3 * b3 + y3 * c3 == 0,
  y3 * a3 + z3 * b3 == 0, y3 * b3 + z3 * c3 == 1, x4 * a4 + y4 * b4 == 1,
  x4 * b4 + y4 * c4 == 0, y4 * a4 + z4 * b4 == 0, y4 * b4 + z4 * c4 == 1,
  x5 * a5 + y5 * b5 == 1, x5 * b5 + y5 * c5 == 0, y5 * a5 + z5 * b5 == 0,
  y5 * b5 + z5 * c5 == 1, x6 * a6 + y6 * b6 == 1, x6 * b6 + y6 * c6 == 0,
  y6 * a6 + z6 * b6 == 0, y6 * b6 + z6 * c6 == 1, x7 * a7 + y7 * b7 == 1,
  x7 * b7 + y7 * c7 == 0, y7 * a7 + z7 * b7 == 0, y7 * b7 + z7 * c7 == 1,
  x8 * a8 + y8 * b8 == 1, x8 * b8 + y8 * c8 == 0, y8 * a8 + z8 * b8 == 0,
  y8 * b8 + z8 * c8 == 1, x9 * a9 + y9 * b9 == 1, x9 * b9 + y9 * c9 == 0,
  y9 * a9 + z9 * b9 == 0, y9 * b9 + z9 * c9 == 1, x10 * a10 + y10 * b10 == 1,
  x10 * b10 + y10 * c10 == 0, y10 * a10 + z10 * b10 == 0, y10 * b10 + z10 * c10 == 1,
  x11 * a11 + y11 * b11 == 1, x11 * b11 + y11 * c11 == 0, y11 * a11 + z11 * b11 == 0,
  y11 * b11 + z11 * c11 == 1, x12 * a12 + y12 * b12 == 1, x12 * b12 + y12 * c12 == 0,
  y12 * a12 + z12 * b12 == 0, y12 * b12 + z12 * c12 == 1, x13 * a13 + y13 * b13 == 1,
  x13 * b13 + y13 * c13 == 0, y13 * a13 + z13 * b13 == 0, y13 * b13 + z13 * c13 == 1,
  x14 * a14 + y14 * b14 == 1, x14 * b14 + y14 * c14 == 0, y14 * a14 + z14 * b14 == 0,
  y14 * b14 + z14 * c14 == 1, x15 * a15 + y15 * b15 == 1, x15 * b15 + y15 * c15 == 0,
  y15 * a15 + z15 * b15 == 0, y15 * b15 + z15 * c15 == 1, x16 * a16 + y16 * b16 == 1,
  x16 * b16 + y16 * c16 == 0, y16 * a16 + z16 * b16 == 0, y16 * b16 + z16 * c16 == 1,
  x17 * a17 + y17 * b17 == 1, x17 * b17 + y17 * c17 == 0, y17 * a17 + z17 * b17 == 0,
  y17 * b17 + z17 * c17 == 1, x18 * a18 + y18 * b18 == 1, x18 * b18 + y18 * c18 == 0,
  y18 * a18 + z18 * b18 == 0, y18 * b18 + z18 * c18 == 1, x19 * a19 + y19 * b19 == 1,
  x19 * b19 + y19 * c19 == 0, y19 * a19 + z19 * b19 == 0, y19 * b19 + z19 * c19 == 1,
  x20 * a20 + y20 * b20 == 1, x20 * b20 + y20 * c20 == 0, y20 * a20 + z20 * b20 == 0,
  y20 * b20 + z20 * c20 == 1, x21 * a21 + y21 * b21 == 1, x21 * b21 + y21 * c21 == 0,
  y21 * a21 + z21 * b21 == 0, y21 * b21 + z21 * c21 == 1, pathvertical1,
  pathvertical2, pathvertical3, pathvertical4, pathvertical5, pathvertical6,
  pathvertical7, pathvertical8, pathvertical9, pathhorizontal1,
  pathhorizontal2, pathhorizontal3, pathhorizontal4, pathhorizontal5,
  pathhorizontal6, pathhorizontal7, pathhorizontal8, pathhorizontal9,
  pathhorizontal10, pathhorizontal11, pathhorizontal12, pathhorizontal13,
  pathhorizontal14, pathhorizontal15, pathhorizontal16, pathhorizontal17,
  pathhorizontal18, pathhorizontal19, pathhorizontal20, pathhorizontal21,
  pathdiag10, pathdiag11, pathdiag12, pathdiag13, pathdiag14, pathdiag15,
  pathdiag16, pathdiag17, pathdiag18, pathdiag19, pathdiag20, pathdiag21},
 {x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12, x13, x14, x15, x16, x17, x18,
  x19, x20, x21, y1, y2, y3, y4, y5, y6, y7, y8, y9, y10, y11, y12, y13, y14, y15, y16,
  y17, y18, y19, y20, y21, z1, z2, z3, z4, z5, z6, z7, z8, z9, z10, z11, z12, z13, z14,
  z15, z16, z17, z18, z19, z20, z21, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11,
  a12, a13, a14, a15, a16, a17, a18, a19, a20, a21, b1, b2, b3, b4, b5, b6, b7, b8,
  b9, b10, b11, b12, b13, b14, b15, b16, b17, b18, b19, b20, b21, c1, c2, c3, c4, c5,
  c6, c7, c8, c9, c10, c11, c12, c13, c14, c15, c16, c17, c18, c19, c20, c21}, Reals]
```

Out[ ]=
```
{}
```

```
In[1]:= (*code for a grid with 4 events, 1 cross connection and 2 support events*)

In[2]:= alpha = 0 (*set the value for the distance between event (1,1) and support event SE1*)

Out[2]= 0

In[3]:= beta = 1 -
          2 * alpha (*set the value for the horizontal distance between the support events*)

Out[3]= 1

In[4]:= zhorizontal = {t, 0} (*vector for a horizontal connection between events*)

Out[4]= {t, 0}

In[5]:= zvertical = {0, t} (*vector for a vertical connection between events*)

Out[5]= {0, t}

In[6]:= tan = 1 / (1 - 2 * alpha) (*slope of vector zdiag depending on alpha*)

Out[6]= 1

In[7]:= zdiag = {t, tan * t} * Sqrt[1 / (1 + tan^2)] (*vector for a diagonal connection*)
```

Out[7]= $\left\{\frac{t}{\sqrt{2}}, \frac{t}{\sqrt{2}}\right\}$

```
In[8]:= (*take the derivative of all vectors*)

In[9]:= dtzhorizontal = D[zhorizontal, t]

Out[9]= {1, 0}

In[10]:= dtzvertical = D[zvertical, t]

Out[10]=
        {0, 1}

In[11]:= dtzdiag = D[zdiag, t]

Out[11]=
```

$\left\{\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right\}$

```
In[12]:= (*set the distances for the connections*)

In[13]:= stimelike = -1

Out[13]=
        -1

In[14]:= sspacelike = 1

Out[14]=
        1

In[15]:= (*set up the equations for the Haywood gauge*)

In[16]:= gauge1 := x1 * a5 - x1 * a1 + y1 * a2 - y1 * a1 + y1 * b2 - y1 * b1 + y1 * c5 - y1 * c1 +
           y1 * b5 - y1 * b1 + z1 * c2 - z1 * c1 + z1 * b2 - z1 * b1 + x1 * b5 - x1 * b1 == 0

In[17]:= gauge2 := x2 * a5 - x2 * a2 + y2 * a2 - y2 * a2 + y2 * b2 - y2 * b2 + y2 * c5 - y2 * c2 +
           y2 * b5 - y2 * b2 + z2 * c2 - z2 * c2 + z2 * b2 - z2 * b2 + x2 * b5 - x2 * b2 == 0
```

```
In[18]:= gauge3 := x3 * a5 - x3 * a3 + y3 * a4 - y3 * a3 + y3 * b4 - y3 * b3 + y3 * c5 - y3 * c3 +
    y3 * b5 - y3 * b3 + z3 * c4 - z3 * c3 + z3 * b4 - z3 * b3 + x3 * b5 - x3 * b3 == 0

In[19]:= gauge4 := x4 * a5 - x4 * a4 + y4 * a4 - y4 * a4 + y4 * b4 - y4 * b4 + y4 * c5 - y4 * c4 +
    y4 * b5 - y4 * b4 + z4 * c4 - z4 * c4 + z4 * b4 - z4 * b4 + x4 * b5 - x4 * b4 == 0

In[20]:= gauge5 := x5 * a6 - x5 * a5 + y5 * a6 - y5 * a5 + y5 * b6 - y5 * b5 + y5 * c6 - y5 * c5 +
    y5 * b6 - y5 * b5 + z5 * c6 - z5 * c5 + z5 * b6 - z5 * b5 + x5 * b6 - x5 * b5 == 0

In[21]:= gauge6 := x6 * a4 - x6 * a6 + y6 * a6 - y6 * a5 + y6 * b6 - y6 * b5 + y6 * c4 - y6 * c6 +
    y6 * b4 - y6 * b6 + z6 * c6 - z6 * c5 + z6 * b6 - z6 * b5 + x6 * b4 - x6 * b6 == 0

In[22]:= (*define the timelike and spacelike connections of each event*)

In[23]:= pathhorizontal1 := alpha == x1 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y1 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z1 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[24]:= pathhorizontal2 := alpha == x2 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y2 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z2 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[25]:= pathhorizontal3 := alpha == x3 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y3 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z3 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[26]:= pathhorizontal4 := alpha == x4 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y4 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z4 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[27]:= pathhorizontal5 := beta == x5 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y5 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z5 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[28]:= pathhorizontal6 := alpha == x6 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y6 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z6 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[29]:= pathvertical1 := stimelike == x1 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y1 * dtzvertical[[1]] * dtzvertical[[2]] + z1 * dtzvertical[[2]] * dtzvertical[[2]]

In[30]:= pathvertical2 := stimelike == x2 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y2 * dtzvertical[[1]] * dtzvertical[[2]] + z2 * dtzvertical[[2]] * dtzvertical[[2]]

In[31]:= pathvertical3 := stimelike == x3 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y3 * dtzvertical[[1]] * dtzvertical[[2]] + z3 * dtzvertical[[2]] * dtzvertical[[2]]

In[32]:= pathvertical4 := stimelike == x4 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y4 * dtzvertical[[1]] * dtzvertical[[2]] + z4 * dtzvertical[[2]] * dtzvertical[[2]]

In[33]:= pathdiag5 := stimelike == x5 * dtzdiag[[1]] * dtzdiag[[1]] +
    2 * y5 * dtzdiag[[1]] * dtzdiag[[2]] + z5 * dtzdiag[[2]] * dtzdiag[[2]]

In[34]:= pathdiag6 := stimelike == x6 * dtzdiag[[1]] * dtzdiag[[1]] +
    2 * y6 * dtzdiag[[1]] * dtzdiag[[2]] + z6 * dtzdiag[[2]] * dtzdiag[[2]]

In[35]:= (*solve the system of equations by including all of the equations*)

In[36]:= (*the equations regarding the invertibility condition are included
    directly into the system because of the sheer number of equations*)

In[37]:= (*the solutions are restricted to only
    real solutions since the metric is ought to be real*)
```

In[38]:=
```
NSolve[{gauge1, gauge2, gauge3, gauge4, gauge5, gauge6, x1 * a1 + y1 * b1 == 1,
  x1 * b1 + y1 * c1 == 0, y1 * a1 + z1 * b1 == 0, y1 * b1 + z1 * c1 == 1,
  x2 * a2 + y2 * b2 == 1, x2 * b2 + y2 * c2 == 0, y2 * a2 + z2 * b2 == 0,
  y2 * b2 + z2 * c2 == 1, x3 * a3 + y3 * b3 == 1, x3 * b3 + y3 * c3 == 0,
  y3 * a3 + z3 * b3 == 0, y3 * b3 + z3 * c3 == 1, x4 * a4 + y4 * b4 == 1,
  x4 * b4 + y4 * c4 == 0, y4 * a4 + z4 * b4 == 0, y4 * b4 + z4 * c4 == 1,
  x5 * a5 + y5 * b5 == 1, x5 * b5 + y5 * c5 == 0, y5 * a5 + z5 * b5 == 0,
  y5 * b5 + z5 * c5 == 1, x6 * a6 + y6 * b6 == 1, x6 * b6 + y6 * c6 == 0,
  y6 * a6 + z6 * b6 == 0, y6 * b6 + z6 * c6 == 1, pathvertical1, pathvertical2,
  pathvertical3, pathvertical4, pathhorizontal1, pathhorizontal2, pathhorizontal3,
  pathhorizontal4, pathhorizontal5, pathhorizontal6, pathdiag5, pathdiag6},
 {x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5, y6, z1, z2, z3, z4, z5, z6, a1,
  a2, a3, a4, a5, a6, b1, b2, b3, b4, b5, b6, c1, c2, c3, c4, c5, c6}, Reals]
```

Out[38]=
```
{{x1 → 0., x2 → 0., x3 → 0., x4 → 0., x5 → 1., x6 → 0., y1 → -1.,
  y2 → -1., y3 → -1., y4 → -1., y5 → -2., y6 → -1., z1 → -1., z2 → -1.,
  z3 → -1., z4 → -1., z5 → 1., z6 → 0., a1 → 1., a2 → 1., a3 → 1., a4 → 1.,
  a5 → -0.333333, a6 → 0., b1 → -1., b2 → -1., b3 → -1., b4 → -1., b5 → -0.666667,
  b6 → -1., c1 → 0., c2 → 0., c3 → 0., c4 → 0., c5 → -0.333333, c6 → 0.},
 {x1 → 0., x2 → 0., x3 → 0., x4 → 0., x5 → 1., x6 → 0., y1 → -1., y2 → -1.,
  y3 → -1., y4 → -1., y5 → -2., y6 → -1., z1 → -1., z2 → -1., z3 → -1.,
  z4 → -1., z5 → 1., z6 → 0., a1 → 1., a2 → 1., a3 → 1., a4 → 1., a5 → -0.333333,
  a6 → 0., b1 → -1., b2 → -1., b3 → -1., b4 → -1., b5 → -0.666667,
  b6 → -1., c1 → 0., c2 → 0., c3 → 0., c4 → 0., c5 → -0.333333, c6 → 0.}}
```

```
In[1]:= (*code for a grid with 6 events,
        2 cross connections (different direction) and 4 support events*)

In[2]:= alpha = 0 (*set the value for the distance between event (1,1) and support event SE1*)

Out[2]= 0

In[3]:= beta = 1 - 2 * alpha (*set the value for the distance between SE1 and SE3*)

Out[3]= 1

In[4]:= zhorizontal = {t, 0} (*vector for a horizontal connection between events*)

Out[4]= {t, 0}

In[5]:= zvertical = {0, t} (*vector for a vertical connection between events*)

Out[5]= {0, t}

In[6]:= tan = 1 / (1 - 2 * alpha) (*slope of vector zdiag depending on alpha*)

Out[6]= 1

In[7]:= zdiag58 =
         {t, tan * t} * Sqrt[1 / (1 + tan^2)] (*vector for a diagonal for SE5,SE6,SE7,SE8*)
```

Out[7]= $\left\{\frac{t}{\sqrt{2}}, \frac{t}{\sqrt{2}}\right\}$

```
In[8]:= zdiag23 = {t, -tan * t} * Sqrt[1 / (1 + tan^2)]
          (*vector for the diagonal connection for SE1,SE2,SE3 and SE4*)
```

Out[8]= $\left\{\frac{t}{\sqrt{2}}, -\frac{t}{\sqrt{2}}\right\}$

```
In[9]:= (*take the derivative of all vectors*)

In[10]:= dtzhorizontal = D[zhorizontal, t]

Out[10]=
        {1, 0}

In[11]:= dtzvertical = D[zvertical, t]

Out[11]=
        {0, 1}

In[12]:= dtzdiag58 = D[zdiag58, t]

Out[12]=
```

$\left\{\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right\}$

```
In[13]:= dtzdiag23 = D[zdiag23, t]

Out[13]=
```

$\left\{\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right\}$

```
In[14]:= (*set the distances for the connections*)
```

```
In[15]:= stimelike = -1
Out[15]=
-1

In[16]:= sspacelike = 1
Out[16]=
1

In[17]:= (*set up the equations for the Haywood gauge*)

In[18]:= gauge1 := x1 * a7 - x1 * a1 + y1 * a2 - y1 * a1 + y1 * b2 - y1 * b1 + y1 * c7 - y1 * c1 +
    y1 * b7 - y1 * b1 + z1 * c2 - z1 * c1 + z1 * b2 - z1 * b1 + x1 * b7 - x1 * b1 == 0

In[19]:= gauge2 := x2 * a7 - x2 * a2 + y2 * a2 - y2 * a2 + y2 * b2 - y2 * b2 + y2 * c7 - y2 * c2 +
    y2 * b7 - y2 * b2 + z2 * c2 - z2 * c2 + z2 * b2 - z2 * b2 + x2 * b7 - x2 * b2 == 0

In[20]:= gauge3 := x3 * a9 - x3 * a3 + y3 * a4 - y3 * a3 + y3 * b4 - y3 * b3 + y3 * c9 - y3 * c3 +
    y3 * b9 - y3 * b3 + z3 * c4 - z3 * c3 + z3 * b4 - z3 * b3 + x3 * b9 - x3 * b3 == 0

In[21]:= gauge4 := x4 * a9 - x4 * a4 + y4 * a4 - y4 * a4 + y4 * b4 - y4 * b4 + y4 * c9 - y4 * c4 +
    y4 * b9 - y4 * b4 + z4 * c4 - z4 * c4 + z4 * b4 - z4 * b4 + x4 * b9 - x4 * b4 == 0

In[22]:= gauge5 := x5 * a7 - x5 * a5 + y5 * a6 - y5 * a5 + y5 * b6 - y5 * b5 + y5 * c7 - y5 * c5 +
    y5 * b7 - y5 * b5 + z5 * c6 - z5 * c5 + z5 * b6 - z5 * b5 + x5 * b7 - x5 * b5 == 0

In[23]:= gauge6 := x6 * a7 - x6 * a6 + y6 * a6 - y6 * a6 + y6 * b6 - y6 * b6 + y6 * c7 - y6 * c6 +
    y6 * b7 - y6 * b6 + z6 * c6 - z6 * c6 + z6 * b6 - z6 * b6 + x6 * b7 - x6 * b6 == 0

In[24]:= gauge7 := x7 * a8 - x7 * a7 + y7 * a7 - y7 * a8 + y7 * b7 - y7 * b8 + y7 * c8 - y7 * c7 +
    y7 * b8 - y7 * b7 + z7 * c7 - z7 * c8 + z7 * b7 - z7 * b8 + x7 * b8 - x7 * b7 == 0

In[25]:= gauge8 := x8 * a3 - x8 * a8 + y8 * a7 - y8 * a8 + y8 * b7 - y8 * b8 + y8 * c3 - y8 * c8 +
    y8 * b3 - y8 * b8 + z8 * c7 - z8 * c8 + z8 * b7 - z8 * b8 + x8 * b3 - x8 * b8 == 0

In[26]:= gauge9 := x9 * a10 - x9 * a9 + y9 * a10 - y9 * a9 + y9 * b10 - y9 * b9 + y9 * c10 - y9 * c9 +
    y9 * b10 - y9 * b9 + z9 * c10 - z9 * c9 + z9 * b10 - z9 * b9 + x9 * b10 - x9 * b9 == 0

In[27]:= gauge10 := x10 * a6 - x10 * a10 + y10 * a10 - y10 * a9 +
    y10 * b10 - y10 * b9 + y10 * c6 - y10 * c10 + y10 * b6 - y10 * b10 +
    z10 * c10 - z10 * c9 + z10 * b10 - z10 * b9 + x10 * b6 - x10 * b10 == 0

In[28]:= (*define the timelike and spacelike connections of each event*)

In[29]:= pathhorizontal1 := alpha == x1 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y1 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z1 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[30]:= pathhorizontal2 := alpha == x2 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y2 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z2 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[31]:= pathhorizontal3 := alpha == x3 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y3 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z3 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[32]:= pathhorizontal4 := alpha == x4 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y4 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z4 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[33]:= pathhorizontal5 := alpha == x5 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y5 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z5 * dtzhorizontal[[2]] * dtzhorizontal[[2]]
```

```
In[34]:= pathhorizontal6 := alpha == x6 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y6 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z6 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[35]:= pathhorizontal7 := beta == x7 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y7 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z7 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[36]:= pathhorizontal8 := alpha == x8 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y8 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z8 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[37]:= pathhorizontal9 := beta == x9 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y9 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z9 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[38]:= pathhorizontal10 :=
  alpha == x10 * dtzhorizontal[[1]] * dtzhorizontal[[1]] + 2 * y10 * dtzhorizontal[[1]] *
     dtzhorizontal[[2]] + z10 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[39]:= pathvertical1 := stimelike == x1 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y1 * dtzvertical[[1]] * dtzvertical[[2]] + z1 * dtzvertical[[2]] * dtzvertical[[2]]

In[40]:= pathvertical2 := stimelike == x2 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y2 * dtzvertical[[1]] * dtzvertical[[2]] + z2 * dtzvertical[[2]] * dtzvertical[[2]]

In[41]:= pathvertical3 := stimelike == x3 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y3 * dtzvertical[[1]] * dtzvertical[[2]] + z3 * dtzvertical[[2]] * dtzvertical[[2]]

In[42]:= pathvertical4 := stimelike == x4 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y4 * dtzvertical[[1]] * dtzvertical[[2]] + z4 * dtzvertical[[2]] * dtzvertical[[2]]

In[43]:= pathvertical5 := stimelike == x5 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y5 * dtzvertical[[1]] * dtzvertical[[2]] + z5 * dtzvertical[[2]] * dtzvertical[[2]]

In[44]:= pathvertical6 := stimelike == x6 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y6 * dtzvertical[[1]] * dtzvertical[[2]] + z6 * dtzvertical[[2]] * dtzvertical[[2]]

In[45]:= pathdiag7 := stimelike == x7 * dtzdiag23[[1]] * dtzdiag23[[1]] +
    2 * y7 * dtzdiag23[[1]] * dtzdiag23[[2]] + z7 * dtzdiag23[[2]] * dtzdiag23[[2]]

In[46]:= pathdiag8 := stimelike == x8 * dtzdiag23[[1]] * dtzdiag23[[1]] +
    2 * y8 * dtzdiag23[[1]] * dtzdiag23[[2]] + z8 * dtzdiag23[[2]] * dtzdiag23[[2]]

In[47]:= pathdiag9 := stimelike == x9 * dtzdiag58[[1]] * dtzdiag58[[1]] +
    2 * y9 * dtzdiag58[[1]] * dtzdiag58[[2]] + z9 * dtzdiag58[[2]] * dtzdiag58[[2]]

In[48]:= pathdiag10 := stimelike == x10 * dtzdiag58[[1]] * dtzdiag58[[1]] +
    2 * y10 * dtzdiag58[[1]] * dtzdiag58[[2]] + z10 * dtzdiag58[[2]] * dtzdiag58[[2]]

In[49]:= (*solve the system of equations by including all of the equations*)

In[50]:= (*the equations regarding the invertibility condition are included
  directly into the system because of the sheer number of equations*)

In[51]:= (*the solutions are restricted to only
  real solutions since the metric is ought to be real*)
```

In[52]:=
```
NSolve[{gauge1, gauge2, gauge3, gauge4, gauge5, gauge6, gauge7,
  gauge8, gauge9, gauge10, x1 * a1 + y1 * b1 == 1, x1 * b1 + y1 * c1 == 0,
  y1 * a1 + z1 * b1 == 0, y1 * b1 + z1 * c1 == 1, x2 * a2 + y2 * b2 == 1,
  x2 * b2 + y2 * c2 == 0, y2 * a2 + z2 * b2 == 0, y2 * b2 + z2 * c2 == 1,
  x3 * a3 + y3 * b3 == 1, x3 * b3 + y3 * c3 == 0, y3 * a3 + z3 * b3 == 0,
  y3 * b3 + z3 * c3 == 1, x4 * a4 + y4 * b4 == 1, x4 * b4 + y4 * c4 == 0,
  y4 * a4 + z4 * b4 == 0, y4 * b4 + z4 * c4 == 1, x5 * a5 + y5 * b5 == 1,
  x5 * b5 + y5 * c5 == 0, y5 * a5 + z5 * b5 == 0, y5 * b5 + z5 * c5 == 1,
  x6 * a6 + y6 * b6 == 1, x6 * b6 + y6 * c6 == 0, y6 * a6 + z6 * b6 == 0,
  y6 * b6 + z6 * c6 == 1, x7 * a7 + y7 * b7 == 1, x7 * b7 + y7 * c7 == 0,
  y7 * a7 + z7 * b7 == 0, y7 * b7 + z7 * c7 == 1, x8 * a8 + y8 * b8 == 1,
  x8 * b8 + y8 * c8 == 0, y8 * a8 + z8 * b8 == 0, y8 * b8 + z8 * c8 == 1,
  x9 * a9 + y9 * b9 == 1, x9 * b9 + y9 * c9 == 0, y9 * a9 + z9 * b9 == 0,
  y9 * b9 + z9 * c9 == 1, x10 * a10 + y10 * b10 == 1, x10 * b10 + y10 * c10 == 0,
  y10 * a10 + z10 * b10 == 0, y10 * b10 + z10 * c10 == 1, pathvertical1,
  pathvertical2, pathvertical3, pathvertical4, pathvertical5, pathvertical6,
  pathhorizontal1, pathhorizontal2, pathhorizontal3, pathhorizontal4,
  pathhorizontal5, pathhorizontal6, pathhorizontal7, pathhorizontal8,
  pathhorizontal9, pathhorizontal10, pathdiag7, pathdiag8, pathdiag9, pathdiag10},
 {x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, y1, y2, y3, y4, y5, y6, y7, y8, y9, y10,
  z1, z2, z3, z4, z5, z6, z7, z8, z9, z10, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, b1,
  b2, b3, b4, b5, b6, b7, b8, b9, b10, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10}, Reals]
```

Out[52]=
```
{{x1 → 0., x2 → 0., x3 → 0., x4 → 0., x5 → 0., x6 → 0., x7 → 1., x8 → 0.,
  x9 → 1., x10 → 0., y1 → -3., y2 → -3., y3 → -1.5, y4 → -1.5, y5 → -3.,
  y6 → -3., y7 → -1., y8 → -2., y9 → -3., y10 → -2., z1 → -1., z2 → -1.,
  z3 → -1., z4 → -1., z5 → -1., z6 → -1., z7 → -5., z8 → -6., z9 → 3., z10 → 2.,
  a1 → 0.111111, a2 → 0.111111, a3 → 0.444444, a4 → 0.444444, a5 → 0.111111,
  a6 → 0.111111, a7 → 0.833333, a8 → 1.5, a9 → -0.5, a10 → -0.5, b1 → -0.333333,
  b2 → -0.333333, b3 → -0.666667, b4 → -0.666667, b5 → -0.333333, b6 → -0.333333,
  b7 → -0.166667, b8 → -0.5, b9 → -0.5, b10 → -0.5, c1 → 0., c2 → 0., c3 → 0.,
  c4 → 0., c5 → 0., c6 → 0., c7 → -0.166667, c8 → 0., c9 → -0.166667, c10 → 0.},
 {x1 → 0., x2 → 0., x3 → 0., x4 → 0., x5 → 0., x6 → 0., x7 → 1., x8 → 0., x9 → 1.,
  x10 → 0., y1 → -2.82899, y2 → -2.82899, y3 → -0.918068, y4 → -0.918068,
  y5 → -2.82899, y6 → -2.82899, y7 → -0.386264, y8 → -1.38626, y9 → -1.24259,
  y10 → -1., z1 → -1., z2 → -1., z3 → -1., z4 → -1., z5 → -1., z6 → -1.,
  z7 → -3.77253, z8 → -4.77253, z9 → -0.514821, z10 → 0., a1 → 0.12495, a2 → 0.12495,
  a3 → 1.18645, a4 → 1.18645, a5 → 0.12495, a6 → 0.12495, a7 → 0.961956,
  a8 → 2.48346, a9 → 0.250053, a10 → 0., b1 → -0.353483, b2 → -0.353483,
  b3 → -1.08924, b4 → -1.08924, b5 → -0.353483, b6 → -0.353483, b7 → -0.0984933,
  b8 → -0.721363, b9 → -0.603536, b10 → -1., c1 → 0., c2 → 0., c3 → 0.,
  c4 → 0., c5 → 0., c6 → 0., c7 → -0.25499, c8 → 0., c9 → -0.485708, c10 → 0.}}
```

```
In[1]:= (*code for the dual grid of grid 2*)

In[2]:= zhorizontal = {t, 0} (*vector for a horizontal connection between events*)
```

Out[2]= $\{t, 0\}$

```
In[3]:= zvertical = {0, t} (*vector for a vertical connection between events*)
```

Out[3]= $\{0, t\}$

```
In[4]:= zdiag = {t, t / Sqrt[3]} * (1 / Sqrt[1 + 1 / 3])
```

Out[4]= $\left\{\frac{\sqrt{3}\,t}{2}, \frac{t}{2}\right\}$

```
In[5]:= zdiag2 = {t, -t / Sqrt[3]} * (1 / Sqrt[1 + 1 / 3])
```

Out[5]= $\left\{\frac{\sqrt{3}\,t}{2}, -\frac{t}{2}\right\}$

```
In[6]:= (*take the derivative of all vectors*)

In[7]:= dtzhorizontal = D[zhorizontal, t]
```

Out[7]= $\{1, 0\}$

```
In[8]:= dtzvertical = D[zvertical, t]
```

Out[8]= $\{0, 1\}$

```
In[9]:= dtzdiag = D[zdiag, t]
```

Out[9]= $\left\{\frac{\sqrt{3}}{2}, \frac{1}{2}\right\}$

```
In[10]:= dtzdiag2 = D[zdiag2, t]
```

Out[10]=

$$\left\{\frac{\sqrt{3}}{2}, -\frac{1}{2}\right\}$$

```
In[11]:= (*set the distances for the connections*)

In[12]:= sspacelike = Sqrt[3] / 3
```

Out[12]=

$$\frac{1}{\sqrt{3}}$$

```
In[13]:= stimelike = -Sqrt[3] / 3
```

Out[13]=

$$-\frac{1}{\sqrt{3}}$$

```
In[14]:= (*set up the equations for the Haywood gauge*)

In[15]:= gauge1 := x1 * a1 - x1 * a2 + y1 * a1 - y1 * a4 + y1 * b1 - y1 * b4 + y1 * c1 - y1 * c2 +
           y1 * b1 - y1 * b2 + z1 * c1 - z1 * c4 + z1 * b1 - z1 * b4 + x1 * b1 - x1 * b2 == 0

In[16]:= gauge2 := x2 * a2 - x2 * a7 + y2 * a3 - y2 * a2 + y2 * b3 - y2 * b2 + y2 * c2 - y2 * c7 +
           y2 * b2 - y2 * b7 + z2 * c3 - z2 * c2 + z2 * b3 - z2 * b2 + x2 * b2 - x2 * b7 == 0
```

```
In[17]:= gauge3 := x3 * a4 - x3 * a3 + y3 * a3 - y3 * a2 + y3 * b3 - y3 * b2 + y3 * c4 - y3 * c3 +
    y3 * b4 - y3 * b3 + z3 * c3 - z3 * c2 + z3 * b3 - z3 * b2 + x3 * b4 - x3 * b3 == 0

In[18]:= gauge4 := x4 * a5 - x4 * a4 + y4 * a1 - y4 * a4 + y4 * b1 - y4 * b4 + y4 * c5 - y4 * c4 +
    y4 * b5 - y4 * b4 + z4 * c1 - z4 * c4 + z4 * b1 - z4 * b4 + x4 * b5 - x4 * b4 == 0

In[19]:= gauge5 := x5 * a6 - x5 * a5 + y5 * a8 - y5 * a5 + y5 * b8 - y5 * b5 + y5 * c6 - y5 * c5 +
    y5 * b6 - y5 * b5 + z5 * c8 - z5 * c5 + z5 * b8 - z5 * b5 + x5 * b6 - x5 * b5 == 0

In[20]:= gauge6 := x6 * a3 - x6 * a6 + y6 * a6 - y6 * a7 + y6 * b6 - y6 * b7 + y6 * c3 - y6 * c6 +
    y6 * b3 - y6 * b6 + z6 * c6 - z6 * c7 + z6 * b6 - z6 * b7 + x6 * b3 - x6 * b6 == 0

In[21]:= gauge7 := x7 * a7 - x7 * a8 + y7 * a6 - y7 * a7 + y7 * b6 - y7 * b7 + y7 * c7 - y7 * c8 +
    y7 * b7 - y7 * b8 + z7 * c6 - z7 * c7 + z7 * b6 - z7 * b7 + x7 * b7 - x7 * b8 == 0

In[22]:= gauge8 := x8 * a8 - x8 * a1 + y8 * a8 - y8 * a5 + y8 * b8 - y8 * b5 + y8 * c8 - y8 * c1 +
    y8 * b8 - y8 * b1 + z8 * c8 - z8 * c5 + z8 * b8 - z8 * b5 + x8 * b8 - x8 * b1 == 0

In[23]:= (*define the timelike and spacelike connections of each event*)

In[24]:= pathvertical1 := stimelike == x1 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y1 * dtzvertical[[1]] * dtzvertical[[2]] + z1 * dtzvertical[[2]] * dtzvertical[[2]]

In[25]:= pathvertical2 := stimelike == x2 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y2 * dtzvertical[[1]] * dtzvertical[[2]] + z2 * dtzvertical[[2]] * dtzvertical[[2]]

In[26]:= pathvertical3 := stimelike == x3 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y3 * dtzvertical[[1]] * dtzvertical[[2]] + z3 * dtzvertical[[2]] * dtzvertical[[2]]

In[27]:= pathvertical4 := stimelike == x4 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y4 * dtzvertical[[1]] * dtzvertical[[2]] + z4 * dtzvertical[[2]] * dtzvertical[[2]]

In[28]:= pathvertical5 := stimelike == x5 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y5 * dtzvertical[[1]] * dtzvertical[[2]] + z5 * dtzvertical[[2]] * dtzvertical[[2]]

In[29]:= pathvertical6 := stimelike == x6 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y6 * dtzvertical[[1]] * dtzvertical[[2]] + z6 * dtzvertical[[2]] * dtzvertical[[2]]

In[30]:= pathvertical7 := stimelike == x7 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y7 * dtzvertical[[1]] * dtzvertical[[2]] + z7 * dtzvertical[[2]] * dtzvertical[[2]]

In[31]:= pathvertical8 := stimelike == x8 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y8 * dtzvertical[[1]] * dtzvertical[[2]] + z8 * dtzvertical[[2]] * dtzvertical[[2]]

In[32]:= pathhorizontal1 := sspacelike == x1 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y1 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z1 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[33]:= pathhorizontal2 := sspacelike == x2 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y2 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z2 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[34]:= pathhorizontal3 := sspacelike == x3 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y3 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z3 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[35]:= pathhorizontal4 := sspacelike == x4 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y4 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z4 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[36]:= pathhorizontal5 := sspacelike == x5 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y5 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z5 * dtzhorizontal[[2]] * dtzhorizontal[[2]]
```

```
In[37]:= pathhorizontal6 := sspacelike == x6 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
           2 * y6 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z6 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[38]:= pathhorizontal7 := sspacelike == x7 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
           2 * y7 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z7 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[39]:= pathhorizontal8 := sspacelike == x8 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
           2 * y8 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z8 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[40]:= pathdiag3 := sspacelike == x3 * dtzdiag[[1]] * dtzdiag[[1]] +
           2 * y3 * dtzdiag[[1]] * dtzdiag[[2]] + z3 * dtzdiag[[2]] * dtzdiag[[2]]

In[41]:= pathdiag4 := sspacelike == x4 * dtzdiag[[1]] * dtzdiag[[1]] +
           2 * y4 * dtzdiag[[1]] * dtzdiag[[2]] + z4 * dtzdiag[[2]] * dtzdiag[[2]]

In[42]:= pathdiag7 := sspacelike == x7 * dtzdiag[[1]] * dtzdiag[[1]] +
           2 * y7 * dtzdiag[[1]] * dtzdiag[[2]] + z7 * dtzdiag[[2]] * dtzdiag[[2]]

In[43]:= pathdiag8 := sspacelike == x8 * dtzdiag[[1]] * dtzdiag[[1]] +
           2 * y8 * dtzdiag[[1]] * dtzdiag[[2]] + z8 * dtzdiag[[2]] * dtzdiag[[2]]

In[44]:= pathdiag5 := sspacelike == x5 * dtzdiag2[[1]] * dtzdiag2[[1]] +
           2 * y5 * dtzdiag2[[1]] * dtzdiag2[[2]] + z5 * dtzdiag2[[2]] * dtzdiag2[[2]]

In[45]:= pathdiag6 := sspacelike == x6 * dtzdiag2[[1]] * dtzdiag2[[1]] +
           2 * y6 * dtzdiag2[[1]] * dtzdiag2[[2]] + z6 * dtzdiag2[[2]] * dtzdiag2[[2]]

In[46]:= pathdiag1 := sspacelike == x1 * dtzdiag2[[1]] * dtzdiag2[[1]] +
           2 * y1 * dtzdiag2[[1]] * dtzdiag2[[2]] + z1 * dtzdiag2[[2]] * dtzdiag2[[2]]

In[47]:= pathdiag2 := sspacelike == x2 * dtzdiag2[[1]] * dtzdiag2[[1]] +
           2 * y2 * dtzdiag2[[1]] * dtzdiag2[[2]] + z2 * dtzdiag2[[2]] * dtzdiag2[[2]]

In[48]:= (*solve the system of equations by including all of the equations*)
         (*the equations regarding the invertibility condition are included
          directly into the system because of the sheer number of equations*)

In[49]:= (*the solutions are restricted to only
          real solutions since the metric is ought to be real*)
```

In[50]:=
```
NSolve[{gauge1, gauge2, gauge3, gauge4, gauge5, gauge6, gauge7, gauge8,
  x1 * a1 + y1 * b1 == 1, x1 * b1 + y1 * c1 == 0, y1 * a1 + z1 * b1 == 0, y1 * b1 + z1 * c1 == 1,
  x2 * a2 + y2 * b2 == 1, x2 * b2 + y2 * c2 == 0, y2 * a2 + z2 * b2 == 0, y2 * b2 + z2 * c2 == 1,
  x3 * a3 + y3 * b3 == 1, x3 * b3 + y3 * c3 == 0, y3 * a3 + z3 * b3 == 0, y3 * b3 + z3 * c3 == 1,
  x4 * a4 + y4 * b4 == 1, x4 * b4 + y4 * c4 == 0, y4 * a4 + z4 * b4 == 0, y4 * b4 + z4 * c4 == 1,
  x5 * a5 + y5 * b5 == 1, x5 * b5 + y5 * c5 == 0, y5 * a5 + z5 * b5 == 0, y5 * b5 + z5 * c5 == 1,
  x6 * a6 + y6 * b6 == 1, x6 * b6 + y6 * c6 == 0, y6 * a6 + z6 * b6 == 0, y6 * b6 + z6 * c6 == 1,
  x7 * a7 + y7 * b7 == 1, x7 * b7 + y7 * c7 == 0, y7 * a7 + z7 * b7 == 0, y7 * b7 + z7 * c7 == 1,
  x8 * a8 + y8 * b8 == 1, x8 * b8 + y8 * c8 == 0, y8 * a8 + z8 * b8 == 0, y8 * b8 + z8 * c8 == 1,
  pathvertical1, pathvertical2, pathvertical3, pathvertical4, pathvertical5,
  pathvertical6, pathvertical7, pathvertical8, pathhorizontal1,
  pathhorizontal2, pathhorizontal3, pathhorizontal4, pathhorizontal5,
  pathhorizontal6, pathhorizontal7, pathhorizontal8, pathdiag1, pathdiag2,
  pathdiag3, pathdiag4, pathdiag5, pathdiag6, pathdiag7, pathdiag8},
 {x1, x2, x3, x4, x5, x6, x7, x8, y1, y2, y3, y4, y5, y6, y7, y8, z1, z2,
  z3, z4, z5, z6, z7, z8, a1, a2, a3, a4, a5, a6, a7, a8, b1, b2,
  b3, b4, b5, b6, b7, b8, c1, c2, c3, c4, c5, c6, c7, c8}, Reals]
```

Out[50]=

```
{}
```

```
In[1]:= (*code for the dual grid of grid 1*)

In[2]:= zhorizontal = {t, 0} (*vector for a horizontal connection between events*)
```

Out[2]= $\{t, 0\}$

```
In[3]:= zvertical = {0, t} (*vector for a vertical connection between events*)
```

Out[3]= $\{0, t\}$

```
In[4]:= zdiag = {t, t / Sqrt[3]} * (1 / Sqrt[1 + 1 / 3])
```

Out[4]= $\left\{\frac{\sqrt{3}\,t}{2}, \frac{t}{2}\right\}$

```
In[5]:= zdiag2 = {t, -t / Sqrt[3]} * (1 / Sqrt[1 + 1 / 3])
```

Out[5]= $\left\{\frac{\sqrt{3}\,t}{2}, -\frac{t}{2}\right\}$

```
In[6]:= (*other possible vectors that have been tested*)

In[7]:= zdualtime = {t / Sqrt[3], t} * Sqrt[1 / (1 + 1 / 3)]
```

Out[7]= $\left\{\frac{t}{2}, \frac{\sqrt{3}\,t}{2}\right\}$

```
In[8]:= zC6C7 = {0, t}
```

Out[8]= $\{0, t\}$

```
In[9]:= zC1C7 = {t, Sqrt[3] / 9 * t} * 1 / Sqrt[1 + 3 / 81]
```

Out[9]= $\left\{\frac{3}{2}\sqrt{\frac{3}{7}}\,t, \frac{t}{2\sqrt{7}}\right\}$

```
In[10]:= (*take the derivative of all vectors*)

In[11]:= dtzhorizontal = D[zhorizontal, t]
```

Out[11]=

$\{1, 0\}$

```
In[12]:= dtzvertical = D[zvertical, t]
```

Out[12]=

$\{0, 1\}$

```
In[13]:= dtzdiag = D[zdiag, t]
```

Out[13]=

$\left\{\frac{\sqrt{3}}{2}, \frac{1}{2}\right\}$

```
In[14]:= dtzdiag2 = D[zdiag2, t]
```

Out[14]=

$\left\{\frac{\sqrt{3}}{2}, -\frac{1}{2}\right\}$

In[15]:=
```
dtzdualtime = D[zdualtime, t]
```
Out[15]=

$$\left\{\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}$$

In[16]:=
```
dtzC6C7 = D[zC6C7, t]
```
Out[16]=

$$\{0, 1\}$$

In[17]:=
```
dtzC1C7 = D[zC1C7, t]
```
Out[17]=

$$\left\{\frac{3\sqrt{\frac{3}{7}}}{2}, \frac{1}{2\sqrt{7}}\right\}$$

In[18]:=
```
(*set the distances for the connections*)
```

In[19]:=
```
sspacelike = Sqrt[3] / 3
```
Out[19]=

$$\frac{1}{\sqrt{3}}$$

In[20]:=
```
stimelike = -Sqrt[3] / 3
```
Out[20]=

$$-\frac{1}{\sqrt{3}}$$

In[21]:=
```
(*set the other distances that have been tested*)
```

In[22]:=
```
sC1C7 = Sqrt[21] / 3
```
Out[22]=

$$\sqrt{\frac{7}{3}}$$

In[23]:=
```
sC5C8 = 2 * sspacelike
```
Out[23]=

$$\frac{2}{\sqrt{3}}$$

In[24]:=
```
(*set up the equations for the Haywood gauge*)
```

In[25]:=
```
gauge1 := x1 * a1 - x1 * a2 + y1 * a1 - y1 * a4 + y1 * b1 - y1 * b4 +
   y1 * c1 - y1 * c2 + y1 * b1 - y1 * b2 + z1 * c1 - z1 * c4 + z1 * b1 -
   z1 * b4 + x1 * b1 - x1 * b2 == 0
```

In[26]:=
```
gauge2 := x2 * a2 - x2 * a8 + y2 * a3 - y2 * a2 + y2 * b3 - y2 * b2 + y2 * c2 - y2 * c8 +
   y2 * b2 - y2 * b8 + z2 * c3 - z2 * c2 + z2 * b3 - z2 * b2 + x2 * b2 - x2 * b8 == 0
```

In[27]:=
```
gauge3 := x3 * a4 - x3 * a3 + y3 * a3 - y3 * a2 + y3 * b3 - y3 * b2 + y3 * c4 - y3 * c3 +
   y3 * b4 - y3 * b3 + z3 * c3 - z3 * c2 + z3 * b3 - z3 * b2 + x3 * b4 - x3 * b3 == 0
```

In[28]:=
```
gauge4 := x4 * a5 - x4 * a4 + y4 * a1 - y4 * a4 + y4 * b1 - y4 * b4 + y4 * c5 - y4 * c4 +
   y4 * b5 - y4 * b4 + z4 * c1 - z4 * c4 + z4 * b1 - z4 * b4 + x4 * b5 - x4 * b4 == 0
```

```
In[29]:= gauge5 := x5 * a6 - x5 * a5 + y5 * a8 - y5 * a5 + y5 * b8 - y5 * b5 + y5 * c6 - y5 * c5 +
    y5 * b6 - y5 * b5 + z5 * c8 - z5 * c5 + z5 * b8 - z5 * b5 + x5 * b6 - x5 * b5 == 0

In[30]:= gauge6 := x6 * a3 - x6 * a6 + y6 * a6 - y6 * a7 + y6 * b6 - y6 * b7 + y6 * c3 - y6 * c6 +
    y6 * b3 - y6 * b6 + z6 * c6 - z6 * c7 + z6 * b6 - z6 * b7 + x6 * b3 - x6 * b6 == 0

In[31]:= gauge7 := x7 * a7 - x7 * a1 + y7 * a6 - y7 * a7 + y7 * b6 - y7 * b7 + y7 * c7 - y7 * c1 +
    y7 * b7 - y7 * b1 + z7 * c6 - z7 * c7 + z7 * b6 - z7 * b7 + x7 * b7 - x7 * b1 == 0

In[32]:= gauge8 := x8 * a8 - x8 * a7 + y8 * a8 - y8 * a5 + y8 * b8 - y8 * b5 + y8 * c8 - y8 * c7 +
    y8 * b8 - y8 * b7 + z8 * c8 - z8 * c5 + z8 * b8 - z8 * b5 + x8 * b8 - x8 * b7 == 0

In[33]:= (*define the timelike and spacelike connections of each event*)

In[34]:= pathvertical1 := stimelike == x1 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y1 * dtzvertical[[1]] * dtzvertical[[2]] + z1 * dtzvertical[[2]] * dtzvertical[[2]]

In[35]:= pathvertical2 := stimelike == x2 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y2 * dtzvertical[[1]] * dtzvertical[[2]] + z2 * dtzvertical[[2]] * dtzvertical[[2]]

In[36]:= pathvertical3 := stimelike == x3 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y3 * dtzvertical[[1]] * dtzvertical[[2]] + z3 * dtzvertical[[2]] * dtzvertical[[2]]

In[37]:= pathvertical4 := stimelike == x4 * dtzvertical[[1]] * dtzvertical[[1]] +
    2 * y4 * dtzvertical[[1]] * dtzvertical[[2]] + z4 * dtzvertical[[2]] * dtzvertical[[2]]

In[38]:= pathhorizontal3 := sspacelike == x3 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y3 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z3 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[39]:= pathhorizontal4 := sspacelike == x4 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y4 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z4 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[40]:= pathhorizontal5 := sspacelike == x5 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y5 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z5 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[41]:= pathhorizontal6 := sspacelike == x6 * dtzhorizontal[[1]] * dtzhorizontal[[1]] +
    2 * y6 * dtzhorizontal[[1]] * dtzhorizontal[[2]] + z6 * dtzhorizontal[[2]] * dtzhorizontal[[2]]

In[42]:= pathdiag3 := sspacelike == x3 * dtzdiag[[1]] * dtzdiag[[1]] +
    2 * y3 * dtzdiag[[1]] * dtzdiag[[2]] + z3 * dtzdiag[[2]] * dtzdiag[[2]]

In[43]:= pathdiag4 := sspacelike == x4 * dtzdiag[[1]] * dtzdiag[[1]] +
    2 * y4 * dtzdiag[[1]] * dtzdiag[[2]] + z4 * dtzdiag[[2]] * dtzdiag[[2]]

In[44]:= pathdiag1r := sspacelike == x1 * dtzdiag[[1]] * dtzdiag[[1]] +
    2 * y1 * dtzdiag[[1]] * dtzdiag[[2]] + z1 * dtzdiag[[2]] * dtzdiag[[2]]

In[45]:= pathdiag7l := sspacelike == x7 * dtzdiag[[1]] * dtzdiag[[1]] +
    2 * y7 * dtzdiag[[1]] * dtzdiag[[2]] + z7 * dtzdiag[[2]] * dtzdiag[[2]]

In[46]:= pathdiag2l := sspacelike == x2 * dtzdiag[[1]] * dtzdiag[[1]] +
    2 * y2 * dtzdiag[[1]] * dtzdiag[[2]] + z2 * dtzdiag[[2]] * dtzdiag[[2]]

In[47]:= pathdiag8r := sspacelike == x8 * dtzdiag[[1]] * dtzdiag[[1]] +
    2 * y8 * dtzdiag[[1]] * dtzdiag[[2]] + z8 * dtzdiag[[2]] * dtzdiag[[2]]

In[48]:= pathspacediag5 := sspacelike == x5 * dtzdiag2[[1]] * dtzdiag2[[1]] +
    2 * y5 * dtzdiag2[[1]] * dtzdiag2[[2]] + z5 * dtzdiag2[[2]] * dtzdiag2[[2]]
```

```
In[49]:= pathspacediag6 := sspacelike == x6 * dtzdiag2[[1]] * dtzdiag2[[1]] +
           2 * y6 * dtzdiag2[[1]] * dtzdiag2[[2]] + z6 * dtzdiag2[[2]] * dtzdiag2[[2]]

In[50]:= pathdiag7r := sspacelike == x7 * dtzdiag2[[1]] * dtzdiag2[[1]] +
           2 * y7 * dtzdiag2[[1]] * dtzdiag2[[2]] + z7 * dtzdiag2[[2]] * dtzdiag2[[2]]

In[51]:= pathdiag8l := sspacelike == x8 * dtzdiag2[[1]] * dtzdiag2[[1]] +
           2 * y8 * dtzdiag2[[1]] * dtzdiag2[[2]] + z8 * dtzdiag2[[2]] * dtzdiag2[[2]]

In[52]:= pathdiag1l := sspacelike == x1 * dtzdiag2[[1]] * dtzdiag2[[1]] +
           2 * y1 * dtzdiag2[[1]] * dtzdiag2[[2]] + z1 * dtzdiag2[[2]] * dtzdiag2[[2]]

In[53]:= pathdiag2r := sspacelike == x2 * dtzdiag2[[1]] * dtzdiag2[[1]] +
           2 * y2 * dtzdiag2[[1]] * dtzdiag2[[2]] + z2 * dtzdiag2[[2]] * dtzdiag2[[2]]

In[54]:= pathtimediag5 := stimelike == x5 * dtzdualtime[[1]] * dtzdualtime[[1]] +
           2 * y5 * dtzdualtime[[1]] * dtzdualtime[[2]] + z5 * dtzdualtime[[2]] * dtzdualtime[[2]]

In[55]:= pathtimediag8 := stimelike == x8 * dtzdualtime[[1]] * dtzdualtime[[1]] +
           2 * y8 * dtzdualtime[[1]] * dtzdualtime[[2]] + z8 * dtzdualtime[[2]] * dtzdualtime[[2]]

In[56]:= pathtimediag6 := stimelike == x6 * dtzdualtime[[1]] * dtzdualtime[[1]] +
           2 * y6 * dtzdualtime[[1]] * dtzdualtime[[2]] + z6 * dtzdualtime[[2]] * dtzdualtime[[2]]

In[57]:= pathtimediag7 := stimelike == x7 * dtzdualtime[[1]] * dtzdualtime[[1]] +
           2 * y7 * dtzdualtime[[1]] * dtzdualtime[[2]] + z7 * dtzdualtime[[2]] * dtzdualtime[[2]]

In[58]:= (*solve the system of equations by including all of the equations*)
         (*the equations regarding the invertibility condition are included
          directly into the system because of the sheer number of equations*)

In[59]:= (*the solutions are restricted to only
          real solutions since the metric is ought to be real*)

In[60]:= NSolve[{gauge1, gauge2, gauge3, gauge4, gauge5, gauge6, gauge7,
           gauge8, x1 * a1 + y1 * b1 == 1, x1 * b1 + y1 * c1 == 0, y1 * a1 + z1 * b1 == 0,
           y1 * b1 + z1 * c1 == 1, x2 * a2 + y2 * b2 == 1, x2 * b2 + y2 * c2 == 0, y2 * a2 + z2 * b2 == 0,
           y2 * b2 + z2 * c2 == 1, x3 * a3 + y3 * b3 == 1, x3 * b3 + y3 * c3 == 0, y3 * a3 + z3 * b3 == 0,
           y3 * b3 + z3 * c3 == 1, x4 * a4 + y4 * b4 == 1, x4 * b4 + y4 * c4 == 0, y4 * a4 + z4 * b4 == 0,
           y4 * b4 + z4 * c4 == 1, x5 * a5 + y5 * b5 == 1, x5 * b5 + y5 * c5 == 0, y5 * a5 + z5 * b5 == 0,
           y5 * b5 + z5 * c5 == 1, x6 * a6 + y6 * b6 == 1, x6 * b6 + y6 * c6 == 0, y6 * a6 + z6 * b6 == 0,
           y6 * b6 + z6 * c6 == 1, x7 * a7 + y7 * b7 == 1, x7 * b7 + y7 * c7 == 0, y7 * a7 + z7 * b7 == 0,
           y7 * b7 + z7 * c7 == 1, x8 * a8 + y8 * b8 == 1, x8 * b8 + y8 * c8 == 0, y8 * a8 + z8 * b8 == 0,
           y8 * b8 + z8 * c8 == 1, pathvertical1, pathvertical2, pathvertical3, pathvertical4,
           pathhorizontal3, pathhorizontal4, pathhorizontal5, pathhorizontal6,
           pathdiag3, pathdiag4, pathdiag1r, pathdiag7l, pathdiag2l, pathdiag8r,
           pathspacediag5, pathspacediag6, pathdiag7r, pathdiag8l, pathdiag1l,
           pathdiag2r, pathtimediag5, pathtimediag8, pathtimediag6, pathtimediag7},
          {x1, x2, x3, x4, x5, x6, x7, x8, y1, y2, y3, y4, y5, y6, y7, y8, z1, z2,
           z3, z4, z5, z6, z7, z8, a1, a2, a3, a4, a5, a6, a7, a8, b1, b2,
           b3, b4, b5, b6, b7, b8, c1, c2, c3, c4, c5, c6, c7, c8}, Reals]

Out[60]=
         { }
```